Multimedia Tools and Applications (2021) 80:4533–4551
https://doi.org/10.1007/s11042-020-09899-y

On solving leaf classification using linear regression

Neha Goyal[1] · **Nitin Kumar**[1] · **Kapil**[1]

Received: 18 March 2020 / Revised: 28 July 2020 / Accepted: 16 September 2020 /
Published online: 30 September 2020

Abstract
Plant's conservation is getting close attention nowadays. It requires awareness about ecology among masses. Plant species identification has been proved as a primary step in literature for biodiversity conservation. It is a sequential process from leaf images as input followed by image enhancement algorithms, and feature extraction phase to classification. The complete process of identifying a leaf image requires substantial time. The article focuses on introducing a simpler and computationally inexpensive framework with a performance at par or better as compared to the existing framework. The article covers several findings and results while transforming the proposed framework for plant identification to a parameter specific optimized framework. The findings include optimizing the leaf image dimension, the impact of RGB to grayscale conversion method, and comparative analysis of the proposed framework for classification from images with other frameworks that first extract specific features and then classify. It also represents the whole framework as a regression problem. Further, improvement is incorporated by integrating the benefits of kernel trick in linear regression. Our finding confirms that the framework not only recognizing the leaf images with comparable accuracy but also reduces the computational time significantly to identify leaf images as compared to other frameworks.

Keywords Linear regression · Kernel function · Color to gray-scale conversion · Image down-sampling · Image projection

1 Introduction

Plants are globally identified as an essential factor in biological diversity and an essential resource for the planet and also underpins the functioning of the entire ecosystem. There are an estimated 500,000 species of plants [3]. One-third of plant species are threatened

✉ Neha Goyal
neha.goyal2309@gmail.com

Nitin Kumar
nitin@nituk.ac.in

Kapil
kapil@nitkkr.ac.in

1 NIT, Kurukshetra, Uttarakhand, India

and several plant families that are still un-subscribed or un-documented. Only a few plant species are specific to human use. Several plants play crucial roles in natural ecosystems due to the rare services provided by such plant species. Some plant species are more likely to have important characteristics that may prove to be of utmost importance in future. The significant threats to plant diversity include habitat loss, fragmentation, forest degradation, over exploitation of resources, invasive species, increasing pollution, and climate change [3, 11, 26] etc. Conservation of plant diversity is a challenging task worldwide. Plant species recognition is a necessary step towards diversity conservation due to several reasons e.g. it helps to aware people of plant traits and resources we are getting from the ecosystem and also helps the community to identify various species that are still undocumented and not known by a large community. Plant identification using computer vision methods is a fascinating task among researchers, botanists, medical science personnel, ecosystem conservation community, computer professionals and others. Traditional methods are useful in terms of identification accuracy but inefficient approaches in terms of time taken, expertise knowledge required, species related to a specific region and plant family. Numerous studies [16, 25, 36] have been discussed in the literature about plant identification after getting satisfactory results in the area of object recognition. In the last decade, the number of researchers presented their work for plant species identification using image processing methods and techniques from artificial intelligence to the deep learning models [35, 37].

In literature, many researchers have utilized general framework of plant species identification as a sequential process passing through various phases (Fig. 1). Initially, leaf images are given as input and some preprocessing functions are applied on the input image e.g. Binarizing leaf images, thresholding, boundary extraction, noise removal [17, 18, 23, 34]. In next phase, features are extracted from the image and these features can be represented in many ways such as a vector of intensity values per pixel, or in a more abstract way as a set of edges, regions of a particular shape, their morphological features, color, vein, texture, and various other traits [39]-[24]. The last phase deals with classification of leaf images based on the feature representation extracted in previous phase. There may be some representations which are better than others at simplifying the learning task.

In the last two decades, several feature extraction methods have been reported based on general features and leaf specific, i.e., morphological, color, vein, texture, and various other traits. Wu et al. [39] represented morphological features based on the geometrical attribute of a leaf at low computational complexity. Du et al. [8] introduced move median centers hyper-sphere (MMCH) classifier for plant species identification. The features used in this technique were based on morphological characteristics, i.e., aspect ratio, rectangularity, perimeter ratio of convexity, the area ratio of convexity, sphericity, circularity, eccentricity, form factor, and Hu moments. Turkoglu et. al. [34] exploited color features, textures features based on Gray Level Co-occurrence Matrix (GLCM), vein features and Fourier Descriptor. The authors divided the leaf images into two and four parts. Afterwards, various features were extracted from each piece and combined as a single feature vector corresponding to one leaf [41]. Color features and texture features using GLCM [13] are considered as statistical features in some of studies [27]. Vein features are considered as commonly used features to represent the venation network of leaf images and better describe the leaf images

Fig. 1 General Framework for plant recognition framework

comparative to HoG and LBP features [12]. Bruno et al. [2] uses fractal dimension computed by the box-counting method [33] and Minkowski's multi-scale methods to analyze the complexity of a leaf shape for plant identification using Linear Discriminant Analysis [10, 40] as classification technique. The effectiveness of the method was demonstrated on a small dataset with approximate 600 images from 14 different species. Park et al., [24] introduced a method named as a content-based image retrieval method for leaf classification using the vein network. The method analysed the venation pattern for leaf classification. Sulc et al., [32] proposed a technique to analyze the leaf texture. In this method, leaf was represented by a pair of multi-scale histograms, one computed by analyzing local features of the boundary and the other by analyzing the interior part of a leaf using scale and rotation invariant local binary patterns operator. The method used the Support Vector Machine [5] for leaf classification.

Based on the literature discussed, the plant recognition framework is critically dependent on the representation of input images as a feature vector. There are several issues in the existing framework. Any attempt of an identification system in high dimensional space may cause issue like "curse of dimensionality" [22]. Feature extraction is a pivotal step for down-sampling the feature space presented by images in low dimensional space. The feature extraction phase requires domain knowledge and 90% of the time required from input to classification. Waldchen et al. [37], it covers a range of image enhancement algorithms for improving the content and representation of an object as a feature vector. Along with proficiency in domain knowledge and time consumed, features selected for one group of plant species may not be applicable for other groups because of the diverse characteristics of leaves. It reflects a significant challenge for the feature extraction phase. The article focuses on introducing a straightforward and computationally inexpensive framework with a performance at par or better as compared to the existing framework. It overcomes all the issues with the existing framework. Along with the framework, the article covers the following facts:

- finding the impacts of various RGB to gray conversion function
- introducing the linear regression with the kernel trick
- Rigorous experimental analysis on each of the five publicly available datasets to develop the confidence in the presented framework and reduces the chance of any coincidence.

The organization for the rest of the paper is as follows: Section 2 introduced the related work. Section 3 discusses the proposed framework, the leaf classification model as a regression problem, and the impact of introducing the kernel trick with linear regression. Section 4 representing the experimental setup, including a detailed description of the dataset used. Results and the analysis are presented in Section 5, followed by Conclusion and future work.

2 Related work

2.1 Preliminaries

Let us assume that there are P different plant species in a given dataset. Each class contain N_i $(i = 1, 2, 3, ..., P)$ number of samples such that the total number of images in the datasets is $N = \sum_{i=1}^{P} N_i$. An input image $\mathbf{U}_i^m \in \mathbb{R}^d$ $m = 1, 2, 3, ..., N_i$ represents the $m - th$ image in $i - th$ class and d (=$a \times b$ represents the number of pixels in an image).

To avoid the complexity and various issues that will arise due to high dimensional space, images are down sampled to an order of $a' \times b' (= q)$ and transformed to one dimensional vector using column concatenation approach. The resultant images are represented as

$$\mathbf{U}_i^m \in \mathbb{R}^d \rightarrow \mathbf{V}_i^m \in \mathbb{R}^q$$

where $q < d$. Similar to down-sampling of images to low dimension, the framework can easily be extendable to lower dimensional input images by resizing them to d dimensional while maintaining the aspect ratio. Each image can also be represented as feature vector describing various attributes characterizing various traits and then finally, can be mathematically represented as

$$V_i = [f_i^1 \; ; \; f_i^2 \; ; \; f_i^3; \; ... \; ; \; f_i^s] \in \mathbb{R}^s$$

2.2 RGB to gray conversion

Converting a color image into a gray-scale image is a critical process as it may produce images with loss of useful information or features such as contrast, sharpness, shadow, and structure. Various methods [15, 20, 31] have been suggested in literature for the same and preserve aforementioned features in gray scale images. Here, we have chosen following three different mapping for RGB image sampling to gray scale version and Fig. 2 is representing the RGB images from different dataset and their corresponding Gray scale images using given Conversion function.

1. Luminance: It is a standard and mostly used RGB to gray mapping function. It is linear combination of R, G and B color channels with a fixed weight and resultant image will be luminance channel image.

$$G_l = 0.2989 * R + 0.5870 * G + 0.1140 * B$$

2. Using green channel as gray scale intensity.
3. Principal Component Analysis (PCA) based RGB to gray conversion [7, 29]

3 Proposed method

3.1 The proposed framework of leaf classification

The Leaf classification framework is modeled in Fig. 3.Unlike popular framework depicted in Fig. 1., here the proposed framework does not have any module for feature extraction and feature engineering, thereby getting ahead in the requirement of computational time. The framework introduces several characteristics, i.e., simple, cost-effective with significant performance, and comparable to the alternate framework. It does not require any domain knowledge and feature extraction algorithm to downsample the feature representation of images. Initially, RGB images are given as input. To work on single-channel, RGB images are converted to grayscale images and resized to predefined dimensions. Image down sampling deals with issues like the "curse of dimensionality" and ensure the uniform sampling of all the images to fit in a system of the linear equation. Images are transformed into vector representation as a column vector. Each element in the vector is pixel intensity at a specified location. The classification algorithm is trained using the feature vector representation of images in low dimensional space.

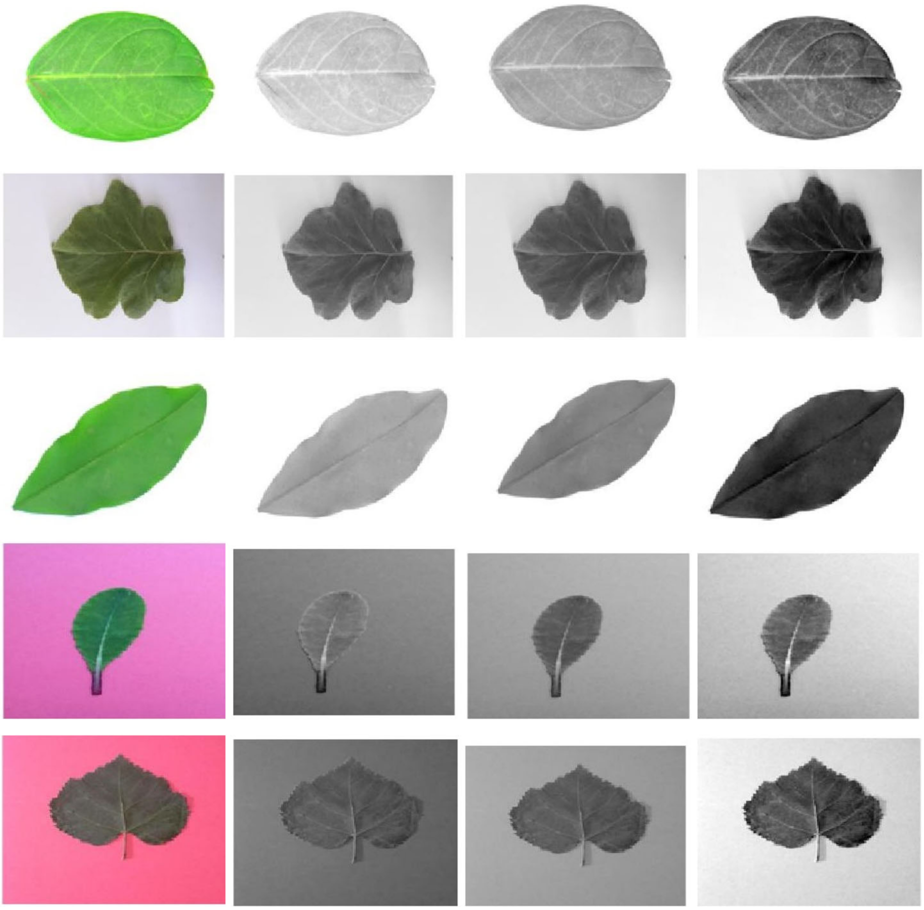


Fig. 2 Sample leaf images with corresponding gray scale images using Green channel, Luminance method and PCA based RGB to gray conversion(left to Right)

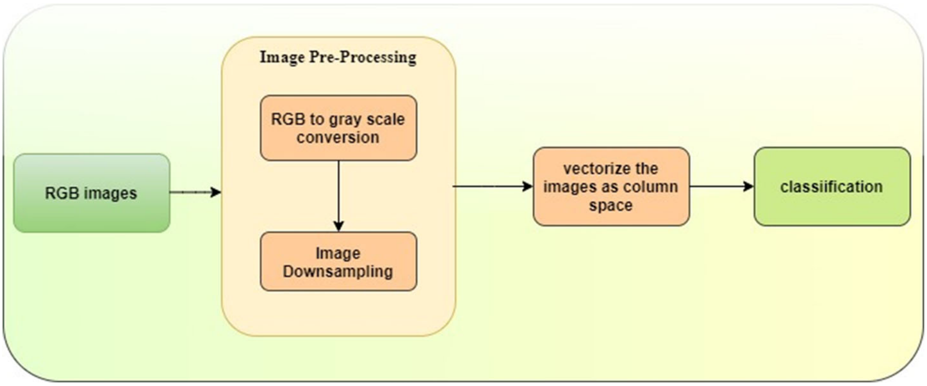


Fig. 3 Proposed framework for plant identification using leaf images

3.2 Modeling leaf classification as a regression problem

The proposed framework is based on the idea that any arbitrary sample from one class lies in subspace generated by other images of the same class. Each feature vector in class i can be represented in the column space of matrix X_i that keeps all the training images as its column i.e.-

$$\boldsymbol{X} = [V_i^1 \;\; V_i^2 \;\; V_i^3 \;\; ... \;\; V_i^{N_i}] \in \mathbb{R}^{q \times N_i} \tag{1}$$

Each column of matrix $\boldsymbol{X}$, is an observation for classification model. The main objective of the approach is to represent a leaf as linear combination of other leaves . To classify the unlabeled test sample $\mathbf{z}$ in one of the class from training data, $\mathbf{z}$ is transformed into a grayscale image with pixel intensity in the interval $[0, 1]$ and then resized to $a' \times b'$ and represented as a one-dimensional vector similar to training images.

$$z \in \mathbb{R}^{a \times b} \rightarrow \mathbb{R}^{a' \times b'} \rightarrow \tilde{z} \in \mathbb{R}^{q \times 1}$$

Here, it is assumed that if image vector $\mathbf{z}$ belongs to $i-th$ class, it can be represented as a linear combination of training images of $i-th$ class. In Fig. 4 approximation of leaf images is represented through linear combination of sample images from its own class. Mathematically,

$$\tilde{z} = \boldsymbol{X_i} \alpha_i$$

However, in reality, such a simplistic notion may not exist. Any test image vector may not entirely lie in the vector space generated by training images of the class it lies in. Therefore, we say that any image of class P has the smallest deviation from its projection on the subspace generated by the training images of the same class.

$$\tilde{z} \approx \boldsymbol{X_i \alpha_i}$$

Therefore, the decision function select the class corresponding to the subspace in which $\tilde{z}'s$ projection has least deviation from its self. That is

$$\arg\min_i ||\tilde{z} - \boldsymbol{X_i \alpha_i}|| \tag{2}$$

The problem is equivalent to find the Least square estimate of $\boldsymbol{\alpha}$ for each image in any class P. $\boldsymbol{\alpha}$ is a vector that keeps the right linear combination to represent the test image in terms of training images with minimum error.

$$\boldsymbol{\alpha_i} = (\boldsymbol{X_i}^t \boldsymbol{X_i})^{-1} \boldsymbol{X_i}^t \tilde{\boldsymbol{z}}.$$

Fig. 4 leaf image as linear combination of training images of its own class

Now for each class, α_i is evaluated for every normalized test image and used to find a mapping vector of $i-th$ class along with X_i. For any test sample $\tilde{z}$, $\hat{z_i}$ is its projection onto $i-th$ subspace generated by $i-th$ class training examples.

$$\begin{aligned} \hat{z_i} &= X_i \alpha_i \\ \hat{z_i} &= X_i (X_i{}^t X_i)^{-1} X_i{}^t \tilde{z} \\ \hat{z_i} &= \hat{Y}_i \tilde{z} \end{aligned} \tag{3}$$

$\hat{Y}_i$ is matrix that map any test sample $\tilde{z}$ to $\hat{z_i}$. $\hat{z_i}$ is the projection of image $\tilde{z}$ in the subspace generated by the training samples of the $i-th$ class. To find the class of test sample z

$$\begin{aligned} dist_i(z) &= ||\tilde{z} - \hat{z_i}|| \\ \min_i dist_i(z),\ \forall i &= 1, 2, ...P \end{aligned} \tag{4}$$

In the Fig 5, a sample leaf image is represented along with its optimal approximation in different classes as a linear combination of sample images from a particular class. After finding mapping in all the possible classes, optimal mapping is selected by minimizing the distance between the sample image and the mapped image.

3.3 Linear regression with kernel trick

Since real problems are not linear generally, here, we extend the above method to its non-linear version by utilizing the kernel trick. Let $\phi : \mathbb{R}^d \rightarrow \mathbb{R}^D$ be a mapping that maps the linear input space to high dimensional kernel space. Therefore, in kernel space decision function must be defined as:

$$\arg\min_i ||\phi(\tilde{\mathbf{z}}) - \hat{\phi}(\tilde{\mathbf{z}}_i)||^2 \tag{5}$$

In the given space $\hat{\phi}(\tilde{\mathbf{z}}_i)$ is projection of $\phi(\tilde{\mathbf{z}})$ on to subspace of training images in kernel space i.e. $\phi(\mathbf{X}_i)$, can be formulated from equation just above (3).

$$\begin{aligned} \hat{\phi}(\tilde{\mathbf{z}}_i) &= \phi(\mathbf{X}_i)(\phi(\mathbf{X}_i)^t \phi(\mathbf{X}_i))^{-1} \phi(\mathbf{X}_i)^t \phi(\tilde{\mathbf{z}}) \\ &= \phi(\mathbf{X}_i) K(\mathbf{X}_i, \mathbf{X}_i)^{-1} K(\mathbf{X}_i, \tilde{\mathbf{z}}) \end{aligned} \tag{6}$$

Here, $\phi(X)^t \phi(Y)$ is replaced by kernel function $K(X, Y)$. Now, using (6) in decision function equation, (5)

$$\begin{aligned} &\min_i ||\phi(\tilde{z}) - \hat{\phi(\tilde{z}_i)}||^2 \\ &= (\phi(\tilde{z}) - \hat{\phi(\tilde{z})})^t (\phi(\tilde{z}) - \hat{\phi}(\tilde{z})) \\ &= (\phi(\tilde{z}) - \phi(\mathbf{X}_i) K(\mathbf{X}_i, \mathbf{X}_i)^{-1} K(\mathbf{X}_i, \tilde{z}))^t (\phi(\tilde{z}) - \phi(\mathbf{X}_i) K(\mathbf{X}_i, \mathbf{X}_i)^{-1} K(\mathbf{X}_i, \tilde{z})) \\ &= K(\tilde{z}, \tilde{z}) - K(\tilde{z}, \mathbf{X}_i) K(\mathbf{X}_i, \mathbf{X}_i)^{-1} K(\mathbf{X}_i, \tilde{z}) \end{aligned}$$

$K(z, z)$ is constant for a particular test image $\mathbf{z}$; therefore, the problem changes to maximizing the 2^{nd} term, therefore:

$$\max_i (K(\tilde{z}, \mathbf{X}_i) K(\mathbf{X}_i, \mathbf{X}_i)^{-1} K(\mathbf{X}_i, \tilde{z})) \tag{7}$$

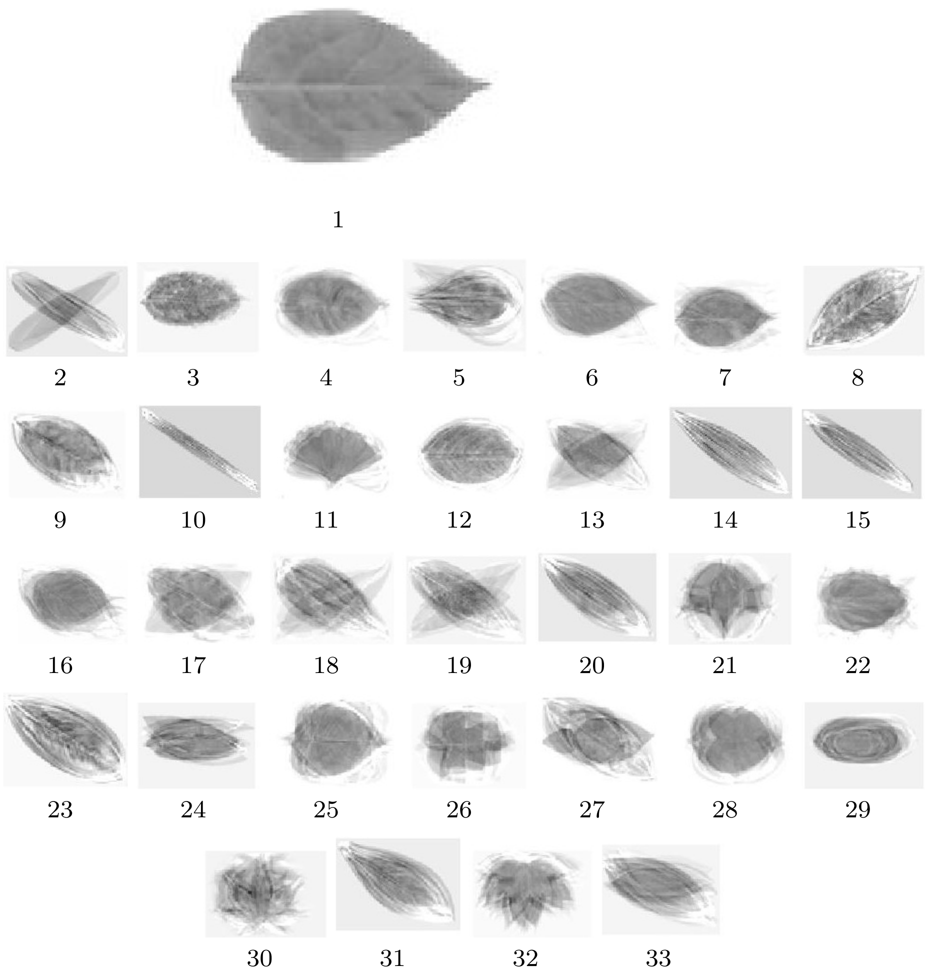


Fig. 5 leaf image(1) and its corresponded mapping in different class as a linear combination of training images from corresponding class (2-33)

4 Experimental setup

We have assumed certain settings which are consistent with the datasets considered such as -

- The background of the image is single colored.
- Generally images are of much higher resolution than required by the proposed or alternative frameworks.
- No image enhancement algorithm is applied to make the feature space better. Leaf images are converted to gray scale image without any noise removal algorithm.

To find the efficiency and significant analysis of the presented approach for plant species recognition system, we have performed extensive experiments on five publicly available leaf datasets, including Flavia, Folio, Leaf, Swedish. Detailed description including the number of plant species and way these images are collected are listed in Table 1.

Table 1 Summary of datasets used in description

Dataset	Number of species	Total leaf images	Categories	Background
Flavia	32	1907	Scanned	White
Folio	32	637	Pseudo-scanned	White
Swedish	13	975	Scanned	White
Leaf	40	443	Psuedo Scanned	Color
UCI	30	340	Psuedo Scanned	Color

Folio leaf dataset [21] contain 637 leaf images for 32 different plant species. These leaf images are photographed on a white background in enough daylight to ensure the optimal pixel intensities. Each species is represented through approximately 20 sample images. The data set represents transformation and rotational invariant leaf images.

Flavia dataset [39] is a collection of 1907 leaves from 32 different plant species, 50–77 images per species. These images are high-resolution images with a size of 1200 by 1600 pixel intensity and acquired by scanners or digital cameras on a plain background. Those leaves were collected from the campus of Nanjing University and the Sun Yat-Sen arboretum, Nanking, China. Most of them are common plants of the Yangtze Delta, China.

Leaf dataset [30] is a collection of 340 leaf samples represented as digital images originating from a total of 40 different plant species. Each leaf image was acquired by taking a photograph over a colored background using an Apple iPad 2 device with a resolution of 720 by 920 pixels.

The Swedish leaf dataset [28]has been collected jointly by Linkoping University and the Swedish Museum of Natural History as a joint project on leaf classification. The dataset contains images of 15 Swedish tree species, with 75 leaves per species (1125 images in total). This dataset contains plant species with high inter-species similarity and rotational and transformations invariant. We are considering only 13 classes from this dataset. Two classes with combined leaves are rejected while performing experiments.

UCI dataset [30] is a subset of Leaf Data set with 30 classes from Leaf dataset. Species numbered from 1 to 15 and from 22 to 36 from Leaf dataset exhibit simple leaves, are considered in the UCI dataset.

Sample images of one identity from all databases are presented in Fig. 6. From the figure, it is noted that Flavia leaf images are taken with different scales and varying rotation. Even the Swedish dataset image is representing leaf with a stalk that may be considered as noise. Some algorithms may require to remove the stalk and results with increased computational cost and recognition time. Leaf data set comprises several types of leaves, i.e., single or complex, needle type leaves with a colored background. The performance of the presented technique in this article is computed as average classification accuracy of new test images.The experiments are performed in k-fold cross-validation fashion. The datasets are divided into training and test datasets accordingly. The value of k is taken from the set $\{2, 3, 4, 5, 10\}$.

5 Results and analysis

In this section, we critically analyze the effectiveness and efficiency of the proposed presentation of the leaf classification model. The efficiency proposed model is represented by comparative analysis with other frameworks. Several experiments are performed to optimize

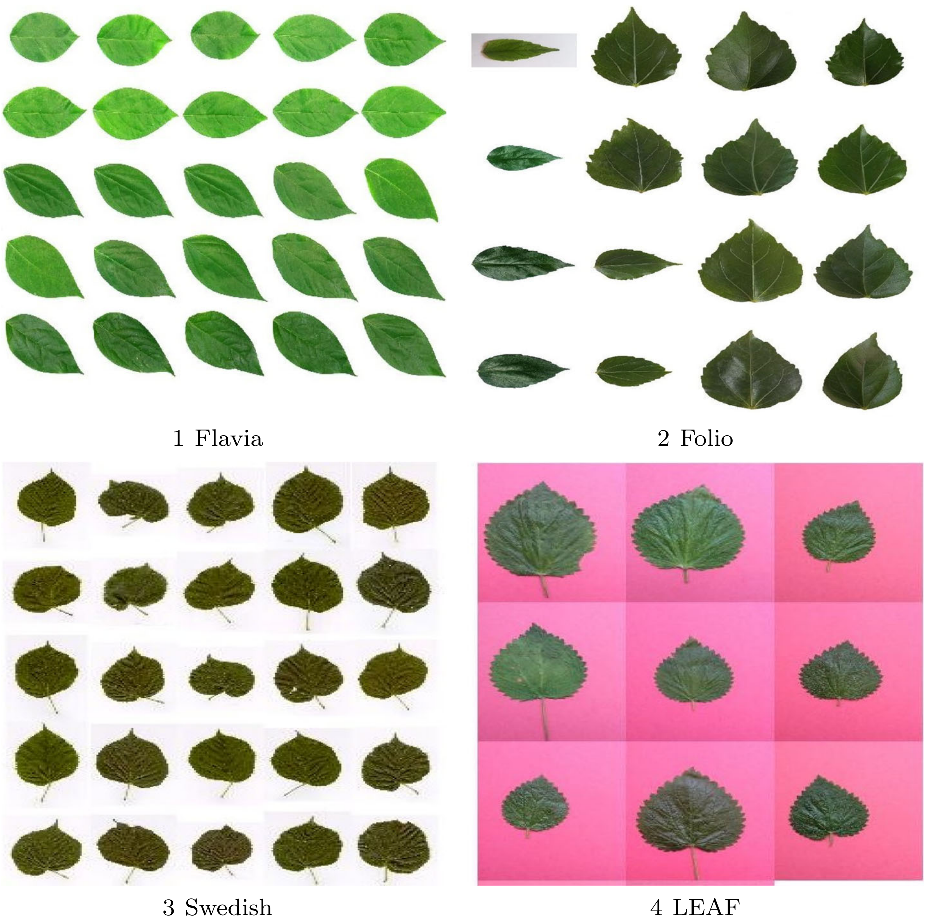


Fig. 6 Sample images from one identity class from each dataset

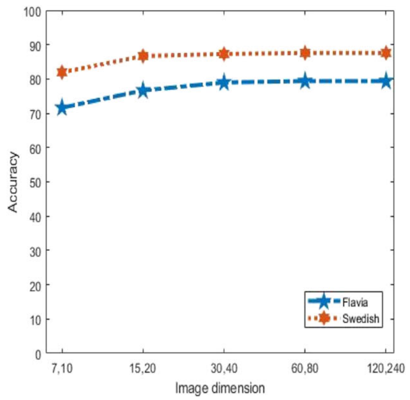
different phases of the proposed framework. The analysis covers the impact of image dimensions on time and performance, the significance of RGB to grayscale conversion method, and improvement in the performance of linear regression with kernel trick.

5.1 Selecting optimum Image dimension

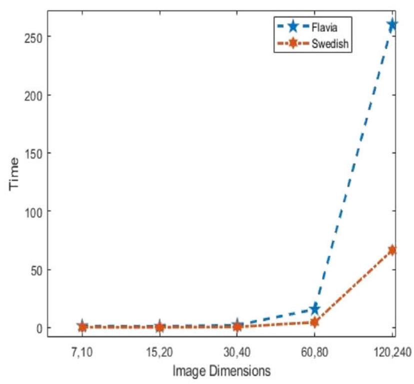
Figure 7 is representing the accuracy and time taken to recognize the leaf images with varying dimensions. The flattened curve of performance after image dimensional 15×20 is ensuring the optimal image dimensions. The time curve is also uplifting with increased dimensions. Further, all the experiments are done with optimize dimensions. leaf images from the different datasets are down-sampled to 15×20 pixel intensity to reduce the overhead of high dimensional space and to maintain aspect ratio.

5.2 Performance comparison on RGB to gray mappings

To examine the impact of the pre-processing phase in the plant recognition system using images. Transforming any three color channel RGB images into one channel grayscale



1 Performance curve with varying image Dimension



2 Time curve with varying image Dimension

Fig. 7 Performance and Time comparison with varying image dimensions (image dimension(row,col)

image is an initial step and proved as a critical step. This conversion is productive in the application of single-channel algorithms on color images. For a comprehensive analysis of the various state of the art methods for conversion are considered, and experiments are performed on every possible combination of the dataset, using linear regression. Table 2 is representing the performance of various experiments with the best accuracy on Linear Regression. From the table, it is interpreted that green color intensity as gray-level intensity is outperforming other methods. The luminance method is performing equally well for the Flavia and the Swedish data set. For the Leaf dataset, the performance of linear regression with the green channel is improving comparatively by 6% and 9% approximately, using Luminance method and PCA based grayscale conversion, respectively.

From Fig. 8, for green channel intensity as a grayscale representation of leaf images, the proposed plant recognition system is recognizing 55% leaf images of the UCI leaf dataset while 87% accurate for Swedish leaves. For the luminance method to convert RGB image to grayscale image, only 50% leaf images from the test sample of the Leaf dataset are recognized correctly. At the same time, for the Swedish dataset, it is performing well with 86% accuracy. For PCA based RGB to gray conversion method recognition system is performing 50% to 87%. From the Fig. 8, it is noted that the green channel is outperforming other state of the art method. Although, Luminance method is a well-known method and utilized mostly as an optimal one. Green channel conversion requires very

Table 2 Performance of Linear regression using Various RGB to gray image decolorization techniques over different datasets

RGB to Gray Conversion	Data set				
	Folio	Flavia	Leaf	Swedish	UCI
Green Channel	79.40	79.81	73.04	86.84	75.54
Luminance method	77.34	79.79	67.84	86.00	71.49
PCA	75.13	78.23	64.67	86.79	67.99

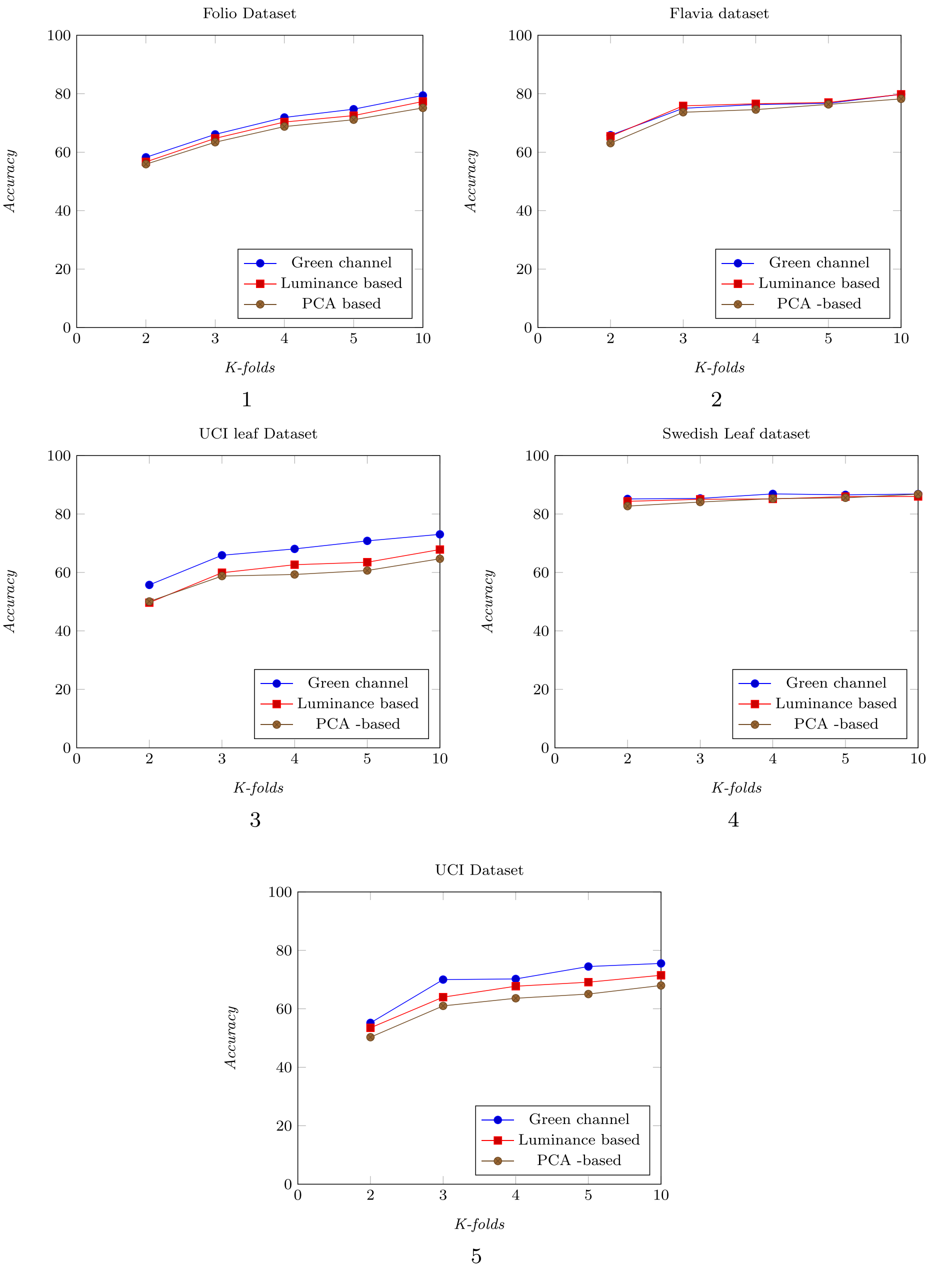


Fig. 8 Performance of Linear regression on different Leaf database using various RGB to gray color mapping techniques

less computation or extracting one channel from three-channel images. And the Luminance method and PCA based grayscale conversion require computational cost and time. From the Table 2 and Fig. 8, it is noted that the Green color intensities are the optimal gray scale intensities for the proposed framework as it is giving better performance and reducing

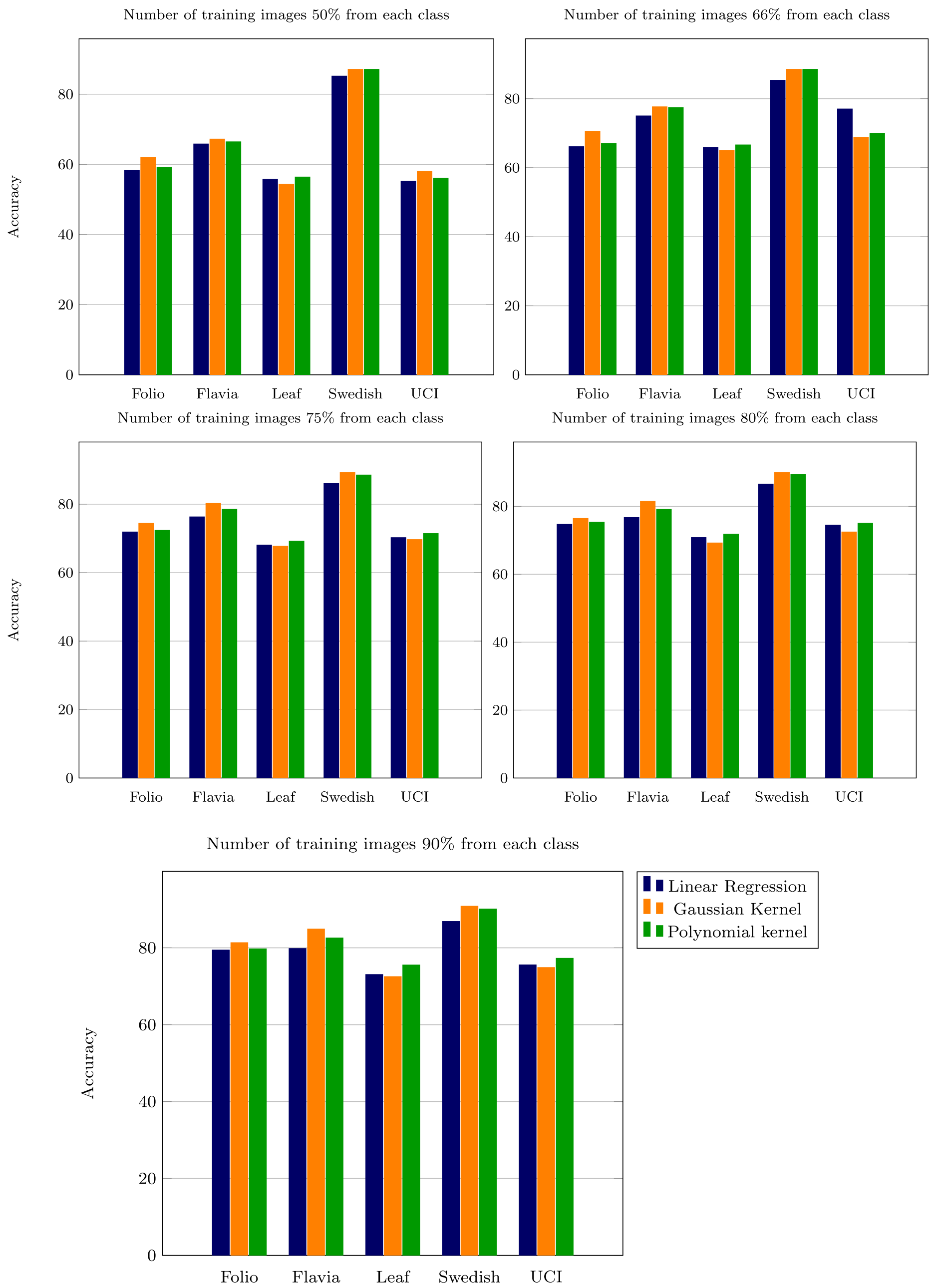


Fig. 9 Performance of Linear Regression and its improvement in kernel space over different dataset and varying number of training images with respect to green channel as gray scale conversion method

the cost of computation requited in remaining two methods. The rest of experiments are analyzed with green channel as gray scale image and predefined image dimensions, i.e., 15×20.

5.3 Analysis of Linear regression in kernel space Over traditional linear regression

Here, we are analyzing the performance of the plant recognition framework with improvements over traditional linear regression. All the experiments are performed with optimized image dimensions and green channel as grayscale pixel intensities. For improvement, linear regression is integrated with the kernel trick.

The kernel trick maps the data in high dimensional space with the purpose that data in higher-dimensional space could become better structured. With this motive, we are using the Gaussian Kernel and polynomial kernel function as a mapping function. Comparative analysis of Linear regression and regression with the Gaussian kernel and the polynomial kernel is represented in Fig. 9. Figure 9 represents the average classification accuracy over the different datasets and the varying number of training images. From the figure, it can be inferred that the Gaussian kernel is outperforming the linear regression and polynomial kernel. For the Leaf dataset and UCI dataset, Linear regression is performing better than the Gaussian kernel. But the polynomial kernel is performing well for the Leaf and UCI datasets.

The results discussed using Figs. 8 and 9, can be summarized in terms of highest accuracy on every dataset achieved in Table 3. From the table, we can see the improvement in recognition rate, and amount of training data requires to find best function to identify the corresponding classes for given test images. For all dataset, maximum number of training images will help to find best fit line as mapping function. Table 4 represents the summarized results of the green channel with a varying number of images, and improvements represent the impact of the training sample requires for any recognition system. Minimum accuracy represents the performance with the least information given as training data. From the table, the performance of the framework with Swedish leaves is comparable with the least information provided. Adequate information and quality of training images impact the recognition information.

5.4 Comparative analysis of proposed framework

The subsection analyses the performance of the proposed framework. Comparative analysis of the proposed framework is reported by comparing the framework presented in the literature that follows the sequential process from image preprocessing algorithm to feature extraction and classification. Tables 5 and 6 is representing the classification accuracy using a general framework and proposed framework using various feature extraction method, pixel intensities as a feature vector and classification algorithm. Table 5 represents the performance of the proposed framework, and the recognition system extracts various handcrafted

Table 3 Performance analysis and improvement of linear regression in kernel space over different dataset with optimal settings

Data set	RGB to gray	Fold(k)	LR	Improved LR	Improvement
Flavia	Green	10	79.81	84.86 (RBF)	5.05
Folio	Green	10	79.4	81.32 (RBF)	1.92
Leaf	Green	10	73.04	75.51 (Poly)	2.47
Swedish	Green	10	86.84	90.8 (RBF)	3.96
UCI	Green	10	75.54	77.26 (Poly)	1.72

Table 4 Classification accuracy over different dataset with respect the green pixel intensity as gray scale intensity and increasing the training images

Dataset	Linear Regression with Kernel	Min Accuracy	Max Accuracy	Improvement
Folio	LR	58.24	79.40	21.16
	LR & RBF	62.01	81.32	19.32
	LR & Polynomial	59.18	79.71	20.53
Flavia	LR	65.80	79.81	14.01
	LR & RBF	67.21	84.86	17.66
	LR & Polynomial	66.43	82.54	16.11
Swedish	LR	85.16	86.84	1.68
	LR & RBF	87.10	90.80	3.69
	LR & Polynomial	87.10	90.07	2.96
Leaf	LR	55.74	73.04	17.31
	LR & RBF	54.32	72.49	18.17
	LR & Polynomial	56.39	75.51	19.12
UCI	LR	55.20	75.54	20.34
	LR & RBF	58.02	74.86	16.84
	LR & Polynomial	56.07	77.26	21.19

features from leaf images instead of using pixel intensity and used linear regression as a classification model. The features descriptor employed to compute texture characteristic of leaf are Hog [6], GLCM [13], Tamura [1, 38], LTE [9], and LBP [4]. The last row in the Table 5 is the performance with a proposed framework of the leaf classification model that follows a simple approach. The proposal does not require any explicit feature descriptors for feature vector representation. For each descriptor, the upper row is the classification performance of LR, and the second row is discussing accuracy over linear regression with kernel trick. Bold

Table 5 Comparative analysis of proposed work with plant species identification framework using hand-crafted features extraction and Linear Regression

Feature Descriptor	Dimensionality	Data set				
		Flavia	Swedish	Folio	UCI	Leaf
GLCM	22	66.93	81.19	63.70	44.19	43.08
		65.06	77.08	58.54	43.35	42.67
HoG	32	67.78	47.71	42.70	20.51	24.45
		81.56	75.92	50.54	27.06	30.20
LBP	59	**93.03**	95.90	75.96	64.79	58.48
		89.70	**96.71**	64.67	69.48	61.56
LTE	9	61.66	77.17	45.48	40.95	36.62
		69.85	81.66	54.60	52.80	51.31
Tamura	3	11.21	9.16	17.73	13.41	10.72
		46.56	55.02	36.24	37.37	31.67
Pixel intensity	300	79.81	86.84	79.40	75.54	73.04
		84.86	**90.8**	**81.32**	**77.26**	**75.51**

Table 6 Comparative analysis of proposed work with plant species identification framework using handcrafted features extraction and classification techniques

Dataset	Feature descriptor	Classification	Accuracy	References
Framework with handcrafted feature extraction and various classification				
Folio	Color+HU+LBP+Haralick	LDA	79.77%	[25]
Swedish	Halalick+HU	LDA(LR)	92.10%(74.56%)	[25]
Flavia	Color+HU+LBP+Haralick	LDA(LR)	89.17%(78.96%)	[25]
Swedish	Margin features	Feed forward Neural Network	82.40%	[28]
Swedish	Fourier	1-NN	89.6	[19]
Swedish	shape context+DP	1-NN	88.12	[19]
ICL	MDM(FD)	SVM	74.20%(66.56%)	[14]
Swedish	Global shape feature (Margin Feature)	SVM	89.15%(92.40%)	[42]
Proposed framework with various classification approach(90% of training data and 10% testing data)				
Flavia	Pixel intensities	SVM	79.27%	
Flavia	Pixel intensities	Logistic Regression	20.26%	
Swedish	Pixel intensities	Logistic Regression	50.96%	
Swedish	Pixel intensities	LDA	94.24%	
Swedish	Pixel intensities	MTSVM	44.68%	
Swedish	Pixel intensities	Linear Regression	91.36%	
Folio	Pixel intensities	SVM	60.94%	
Folio	Pixel intensities	LDA	78.13%	
Folio	Pixel intensities	KNN	62.5%	
Folio	Pixel intensities	Linear Regression	81.25%	
Proposed Framework modelled on Linear Regression with Kernel Trick				
Flavia	Pixel intensity	Linear Regression (RBF)	88.08%	
Folio	Pixel intensity	Linear Regression (RBF)	87.5%	
Swedish	Pixel intensity	Linear Regression (RBF)	90.8%	
UCI	Pixel intensity	Linear Regression (RBF)	80.65%	
Leaf	Pixel intensity	Linear Regression (RBF)	73.17%	

entries in table signifies the maximum accuracy on dataset with different feature descriptor. From the table it is noted that pixel intensity as feature descriptor is performing well with proposed framework. Table 6 is representing the performance of various studies discussed in the literature and follows the framework described in Fig. 1. It also represents the classification accuracy of several experiments performed in our own experimental setting with pixel intensities as a feature vector and several classification approaches. Table 7 is representing the computational time taken by one leaf to classify using approaches discussed in Table 5.

The results confirm that the proposed framework is outperforming and comparable to the existing framework with the least computational cost and time. Only LBP Features descriptor is outperforming for Flavia and Swedish dataset (Table 5). A significant difference in the accuracy of LR and LR with kernel trick using HoG, LBP, and Tamura is representing the benefits of using kernel methods. In the Table 7, There is a noteworthy difference in the time required to compute the features. Here Tamura features are costliest as compared

Table 7 Time required to extract feature of a leaf image and classification(in seconds

Classification	Feature Descriptor					
	GLCM	HoG	LBP	LTE	Tamura	Gray Scale Intensity
Linear Regression	0.1680	0.0451	0.1177	0.1000	29.9825	0.0013
Kernel LR	0.1881	0.0465	0.1427	0.1190	30.0004	0.0361

to other methods of feature extraction while grayscale pixels intensities are having no computational times and evaluation cost and getting second highest accuracy as compared to feature descriptors.

5.5 Analysis of the results

The article presented a novel framework to make plant species identification that represents the whole framework as a system of linear equations. It requires lesser operations to classify leaf images in the predefined category. The framework portrays several benefits being simple, cost-effective, comparable to the existing framework, incremental approach, easy to deploy on a small platform, i.e., android. The framework does not require any domain knowledge to identify feature to distinguish similar plant's leaf. We are using linear regression to model the leaf classification framework. Linear regression aims to find a projection matrix for individual plant species in the dataset to project the new leaf image in as a linear combination of training leaves from any class. It is easy to add more classes at later stages without affecting the mapping function of existing classes. Mapping function for existing classes in database affects only those classes in which training samples are increased from a few to more. Linear regression can be changed to an incremental approach which makes it a more flexible framework when the number of undocumented species exists and the database requires regular up-gradation. From Table 7, it is clearly stated that framework require very less time to identify plant names for a given leaf. From Table 5 LBP features are performing better as compared to proposed framework but time required to extract feature and then classification is 100 times more than time required to classify only one leaf image using proposed framework linear version of linear regression while it takes at least four times more computational time for classification kernel version of linear regression.

6 Conclusion and future work

The article proposes a novel framework for leaf classification that follows a simple approach and model it using linear regression. The proposed presentation is efficient with comparable accuracy and requires lesser time to classify one leaf image. It is an automated framework that does not require any optimization and domain knowledge to identify notable features to differentiate similar objects. The entire process requires no preprocessing algorithm for illumination, rotation, scale, and transformation variant images that make the framework robust for images captured through mobile devices or uncontrolled environments. The framework can be extended to the probabilistic model and semi-supervised framework to make better classification of leaf images using unlabelled data. Besides several benefits form the framework, the proposed presentation is modeled on the lazy classifier. It reduced the time for optimization but required space to store all the training images at a testing time or while classifying new leaf. As the proposed algorithm does not require training but testing requires sufficient time based on the number of the training sample in the dataset. As an improvement, the model can be designed to select most informational images to reduce the space requirement and testing time. Overall we can conclude that the framework is a robust and efficient framework with several benefits.

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. Bagri N, Punit KJ (2015) A comparative study on feature extraction using texture and shape for content based image retrieval. Int J Adv Sci Technol 80.4:41–52
2. Bruno OM, et al. (2008) Fractal dimension applied to plant identification. Inform Sci 178.12:2722–2733
3. CCorlett RT (2016) Plant diversity in a changing world: status, trends, and conservation needs. Plant Diversity 38.1:10–16
4. Chen S, Liu C (2012) Various discriminatory features for eye detection. In: Cross disciplinary biometric systems. Springer, Berlin, pp 183–203
5. Cortes C (1995) Support-vector networks. Machine learning 20(3):273–297
6. Dalal N, Triggs B, Schmid C (2006) Human detection using oriented histograms of flow and appearance. In: European conference on computer vision. Springer, Berlin, pp 428–441
7. Dikbas S, Arici T, Altunbasak Y (2007) Chrominance edge preserving grayscale transformation with approximate first principal component for color edge detection. In: 2007 IEEE international conference on image processing, IEEE, vol 2
8. Du J-X, Xiao-Feng W, Guo-Jun Z (2007) Leaf shape based plant species recognition. Appl Math Comput 185.2:883–893
9. Ertugrul OF, et al. (2015) A vision system for classifying butterfly species by using law's texture energy measures. Int J Biomed Data Min pp 1
10. Everitt BS, Dunn G (1991) Applied multivariate data analysis. No. 519.5076 E9
11. Goettsch B, et al. (2015) High proportion of cactus species threatened with extinction. Nat Plants 1.10:15142
12. Goyal N, Kapil G, Nitin K (2019) Multiclass twin support vector machine for plant species identification. Multimed Tools Appl pp 1–24
13. Haralick RM, Karthikeyan S (1973) Its' Hak Dinstein. textural features for image classification. IEEE Trans Syst Man Cybern 6:610–621
14. Hu R, Jia W, Ling H (2012) Multiscale distance matrix for fast plant leaf recognition. IEEE Trans Image Process 21(11):4667–4672
15. Jin Z, Ng MK (2015) A contrast maximization method for color-to-grayscale conversion. Multidimen Syst Signal Process 26.3:869–877
16. Kadir A, et al. (2012) Experiments of Zernike moments for leaf identification. J Theoret Appl Inform Technol (JATIT) 41.1:82–93
17. Kulkarni AH, et al. (2013) A leaf recognition technique for plant classification using RBPNN and Zernike moments. Int J Adv Res Comput Commun Eng 2.1:984–988
18. Le T-L, Tran D-T, Hoang V-N (2014) Fully automatic leaf-based plant identification, application for Vietnamese medicinal plant search. In: Proceedings of the fifth symposium on information and communication technology. ACM
19. Ling H, Jacobs DW (2007) Shape classification using the inner-distance. IEEE Trans Patt Anal Mach Intell 29.2:286–299
20. Liu Q, et al. (2017) Extended RGB2Gray conversion model for efficient contrast preserving decolorization. Multimed Tools Appl 76.12:14055–14074
21. Munisami T, Ramsurn M, Kishnah S, Pudaruth S (2015) Plant leaf recognition using shape features and colour histogram with k-nearest neighbour classifiers. Procedia Comput Sci (Elsevier) J 58:740–747

22. Naseem I, Togneri R, Bennamoun M (2010) Linear regression for face recognition. IEEE Trans Patt Anal Mach Intell 32.11:2106–2112
23. Otsu N (1979) A threshold selection method from gray-level histograms. IEEE Trans Syst Man Cybern 9(1):62–66
24. Park J, EenJun H, Yunyoung N (2008) Utilizing venation features for efficient leaf image retrieval. J Syst Softw 81.1:71–82
25. Pearline S, Anubha V, Sathiesh K, Harini S (2019) A study on plant recognition using conventional image processing and deep learning approaches. Journal of Intelligent & Fuzzy Systems Preprint 1-8
26. Pimm SL, Joppa LN (2015) How many plant species are there, where are they, and at what rate are they going extinct? Ann Missouri Bot Gard 100.3:170–177
27. Pornpanomchai C, et al. (2011) Thai herb leaf image recognition system (THLIRS)
28. S"oderkvist O (2001) Computer vision classification of leaves from swedish trees
29. Seo Ja-Won, Kim SeongDae (2013) Novel PCA-based color-to-gray image conversion. In: 2013 IEEE international conference on image processing, IEEE
30. Silva PedroFB, Marcal AndreRS, Rubim M, da Silva A (2013) Evaluation of features for leaf discrimination. In: International conference image analysis and recognition. springer, Berlin
31. Sowmya V, Govind D, Soman KP (2017) Significance of incorporating chrominance information for effective color-to-grayscale image conversion. Signal Image Video Process 11.1:129–136
32. Sulc Milan, Matas Jiri (2014) Texture-based leaf identification. In: European conference on computer vision. Springer, Cham
33. Tricot C (1994) Curves and fractal dimension. Springer Science & Business Media, New York
34. Turkoglu M, Davut H (2019) Recognition of plant leaves: an approach with hybrid features produced by dividing leaf images into two and four parts. Appl Math Comput 352:1–14
35. Waldchen J, Patrick M (2018) Plant species identification using computer vision techniques: a systematic literature review. Arc Comput Meth Eng 25.2:507–543
36. Waldchen J, Patrick M (2018) Machine learning for image based species identification. Meth Ecol Evol 9.11:2216–2225
37. Waldchen J, et al. (2018) Automated plant species identification—Trends and future directions. PLos Comput Biol 14.4:e1005993
38. Wang B, et al. (2013) Crowd density estimation based on texture feature extraction. J Multimed 8:4
39. Wu SG, et al. (2007) A leaf recognition algorithm for plant classification using probabilistic neural network. In: 2007 IEEE international symposium on signal processing and information technology, IEEE
40. Yan J, et al. (2006) A scalable supervised algorithm for dimensionality reduction on streaming data. Inform Sci 176.14:2042–2065
41. Yigit Enes, et al. (2019) A study on visual features of leaves in plant identification using artificial intelligence techniques. Comput Electron Agricult 156:369–377
42. Zhang X, et al. (2019) Plant recognition via leaf shape and margin features. Multimed Tools Appl 78.19:27463–27489