

## Article

# Plant Recognition Using Morphological Feature Extraction and Transfer Learning over SVM and AdaBoost

Shubham Mahajan <sup>1,\*</sup>, Akshay Raina <sup>2</sup> , Xiao-Zhi Gao <sup>3</sup> and Amit Kant Pandit <sup>1</sup>

<sup>1</sup> School of Electronics & Communication, Shri Mata Vaishno Devi University, Katra 182320, India; amit.pandit@smvdu.com

<sup>2</sup> School of Electrical Engineering, Shri Mata Vaishno Devi University, Katra 182320, India; 18bee003@smvdu.ac.in

<sup>3</sup> School of Computing, University of Eastern Finland, 70210 Kuopio, Finland; xiao.z.gao@uef.fi

\* Correspondence: 19dec001@smvdu.ac.in; Tel.: +91-94196-10760

**Abstract:** Plant species recognition from visual data has always been a challenging task for Artificial Intelligence (AI) researchers, due to a number of complications in the task, such as the enormous data to be processed due to vast number of floral species. There are many sources from a plant that can be used as feature aspects for an AI-based model, but features related to parts like leaves are considered as more significant for the task, primarily due to easy accessibility, than other parts like flowers, stems, etc. With this notion, we propose a plant species recognition model based on morphological features extracted from corresponding leaves' images using the support vector machine (SVM) with adaptive boosting technique. This proposed framework includes the pre-processing, extraction of features and classification into one of the species. Various morphological features like centroid, major axis length, minor axis length, solidity, perimeter, and orientation are extracted from the digital images of various categories of leaves. In addition to this, transfer learning, as suggested by some previous studies, has also been used in the feature extraction process. Various classifiers like the kNN, decision trees, and multilayer perceptron (with and without AdaBoost) are employed on the opensource dataset, FLAVIA, to certify our study in its robustness, in contrast to other classifier frameworks. With this, our study also signifies the additional advantage of 10-fold cross validation over other dataset partitioning strategies, thereby achieving a precision rate of 95.85%.

**Keywords:** plant species recognition; SVM; AdaBoost; non-separable data; feature extraction; feature selection; transfer learning



**Citation:** Mahajan, S.; Raina, A.; Gao, X.-Z.; Kant Pandit, A. Plant Recognition Using Morphological Feature Extraction and Transfer Learning over SVM and AdaBoost. *Symmetry* **2021**, *13*, 356. <https://doi.org/10.3390/sym13020356>

Academic Editor: Peng-Yeng Yin

Received: 19 January 2021

Accepted: 10 February 2021

Published: 22 February 2021

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

There are potentially hundreds of thousands of species of plants that exist on earth presently, out of which a large number contribute to medicinal use to human beings, while others are poisonous. There are many other uses in this regard as well. This corresponds to the necessity of recognizing such species using the resources available. This task of identification and classification can be solved in most promising way using various tools in the domain of artificial intelligence (AI).

There are a number of methods in ML and deep learning that can be implemented to address this task, such as regression-based models as supervised machine learning models, or computer vision models using convolution neural networks. For instance, Sun et al. [1] proposed a binary generator network to solve the problem of how a generative adversarial network (GAN) generates a lesion image with a specific shape, and used the edge-smoothing and image pyramid algorithm to solve the problem that occurs when synthesizing a complete lesion leaf image where the synthetic edge pixels are different and the network output size is fixed, but the real lesion size is random. With this, the authors clearly accomplished proving the necessity as well as effective implementation of AI-based

techniques at tasks like plant leaf recognition or disease recognition. The same can also be verified from studies like that of Yang et al. [2].

For some of these computational models, there exist certain challenges. One of them is the extraction and selection of features from leaf images for accurate recognition. As mentioned in next section, some studies used artificially developed features, while others made use of texture-based, morphological features-based, or color-based features. Therefore, the architectural structure of these models is preprocessing, feature extraction and selection, and finally classification. Some works have also been done which propose the use of convolution-based neural networks like the residual network. Because more smart and better processing-based cameras are available these days, the dataset has become more informative about the above-said features, thereby leading these models to perform to a greater extent, empowering human perception and intelligence to classify and recognize plant species.

With this, our study aims to propose a novel classification model to identify plant species using morphological features, as well as from transfer learning and adaptive boosting, with high and competitive accuracy. The main contributions of our study are

- Proposing a robust, precise, and fast plant-species recognition model;
- Making use of morphology-based features from leaf images with low dimensionality;
- Using features by transfer learning from low-dimensional ConvNet architecture;
- Evaluation of different classifiers using controlled comparison;
- Enhancing the classification results via adaptive boosting.

This paper has been sub-divided into seven sections. Existing models and studies done addressing this task are presented in Section 1.1. The methodology discussed in detail, including pre-processing techniques used, features extracted, and model implemented, is discussed in Section 2. Section 3 presents the dataset used, split into training and testing sets, and models the hyper-parameter selection and the flow of the methodology. In Section 4, the analysis of results obtained, and their comparison with other existing models is done. Finally, in Section 5, we exploit the credibility of the proposed methodology with regards to the detailed discussion of the implementation in this paper, and we make some concluding remarks in Section 6.

### 1.1. Literature Review

Fortunately, there has been intensive work done in past decade for this task, thereby giving more and better insights to present studies like this one. In one of such proposals, by Zhang Y et al. [3], a bag of features (BOF) was implemented, and the authors achieved an overall maximum accuracy of 94.22%. Azlah et al. [4] provided a controlled and well-analyzed comparative study on various studies and models on various bases, including CNN-based, mathematical learning-based, etc. They reviewed different techniques for plant leaf classification, including CNNs, support vector machines (SVMs), and kNNs, and provided important insights for development of methodologies at this task. They summarized characteristics of these classifiers and stated that CNNs are disadvantageous for intensive computation and incapable of generalization.

Another important study has been done by Munish [5], in which they used different classifier models like decision tree, kNN, and multilayer perceptron, and implemented the AdaBoost technique for improving precision. They achieved a precision rate of 95.42%. Jeon and Rhee [6] implemented convolution neural network-based classifier Google Net and provided good insights on the use of CNNs at this task, achieving a recognition rate of above 94% of their system. In another study by Pankaja and Suma [7], texture- and shape-based features were extracted and PCA classifier was used. They claimed 96.66% achievable accuracy. An accuracy of 94.0% was achieved by using the multilayer perceptron (MLP) classifier [8] trained over morphological features. From the same morphological features, Kadir [9] achieved an accuracy of 93.75%. From a combination of morphology-based, vein structure-based, and geometrically-based features, Arun [10] implemented the support vector machine classification model and got to an accuracy of 94.20%. Using edge and color histograms and the area of leaves, Anami [11] proposed a leaf recognition model

and achieved an accuracy of 93.6%. The elliptic Fourier analysis was used by Ekshinge and Andore [12], which got them to an accuracy of 85% from shape-based features. Sun [13] achieved a recognition rate of 91.78% using a deep-learning model consisting of 26 layers with eight residual blocks on the BJFU100 dataset.

## 2. Proposed Methodology

### 2.1. Support Vector Machine (SVM)

SVM [14] is a supervised machine learning model for classification and regression analysis. If  $\hat{S}$  is the number of features,  $\hat{H}$  is a set of  $\hat{S}$  dimensional hyperplanes, and  $\hat{h} \in \hat{H}$  is where  $\hat{h}$  has maximum margin, i.e., the maximum distance between data points of both classes. An SVM aims at finding an  $\hat{h}$  in  $\hat{H}$  that can classify all of the examples as points in space distinctly. This model is both memory-efficient and highly effective in multi-dimensional feature space. Traditionally, this was a binary-class classifier. However, later many new efficient approaches were introduced for multiclass classifications. Out of these, Duan and Keerthi [15] suggest using the PWC PSVM, among others, which is pairwise coupling implementation [16] of one-versus-one classifiers, and refers to the PWC implementations with PSVM.

There are certain tuning parameters in an SVM model that help it optimize the results with respect to the specific data points available. One of them is Kernel, which is a mathematical function and operates in such a way as to take data as input and transform it into a necessary form. These functions return the inner product between two points in an acceptable space and may be of various forms, e.g., linear, nonlinear, polynomial, radial base function (RBF), and sigmoid. These kernels are based on fact that non-separable features become linearly separable (often) upon their mapping to highly dimensional feature space. One of the commonly used kernels is the RBF kernel, which on two samples,  $x$  and  $\bar{x}$ , is represented as feature vectors in some input space, as defined in Equation (1) below:

$$K(x, \bar{x}) = e^{\left(\frac{-||x - \bar{x}||^2}{2\sigma^2}\right)} \quad (1)$$

where  $K(x, \bar{x})$  is also called Gaussian RBF if it is parametrized using  $\gamma = -\frac{1}{\sigma^2}$ , where  $\gamma > 0$ ,  $\sigma$  is a free parameter, and  $||x - \bar{x}||^2$  is considered as the squared Euclidean distance between two feature vectors.

Another parameter in SVM modelling is the regularization (denoted by  $C$ ), which generically represents the tolerance level of mis-classifications. A higher value of  $C$  means that the model will not tolerate ideally any misclassifications, and will try to fit all of the points to its potential. There is another parameter, namely gamma ( $\gamma$ ), which describes the effect of the training example. This implies that with a low gamma, focuses a long way from dependable separation lines are considered for computation when fitting, while a high gamma implies that focuses near conceivable lines are considered for calculation.

### 2.2. Adaptive Boosting (AdaBoost)

Boosting [17], in general, is a method of converting a family of weak learners into a strong learner. We combine these weak classifiers (slightly better than deciding from a coin toss) so that they have minimum correlation with each other, perform democratic voting (figuratively), and the result turns out to be a strong classifier. A few types of boosting method are gradient tree boosting, Adaptive Boosting (AdaBoost), and Extreme Gradient Boosting (XGBoost).

AdaBoost was developed by Yoav and Robert, and they won Gödel Prize in 2003 for this study. It was implemented in combination with various types of learning algorithms. The output of these weak classifiers is summed using weights, which serve as the representation for output of the boosted classifier. A representation of algorithm of AdaBoost is shown in Figure 1a. AdaBoost is adaptive in the sense that succeeding weak learners adjust (adapt) in favor of those samples that were wrongly classified by earlier classifying-blocks. It is sensitive to noisy data and outliers [18]. Traditionally, it has been based on binary-class

problem. However, a few later studies have suggested its use for multi-class classification problems as well [19,20]. The algorithm, one of these problems [19], is shown in Figure 1b.

**Algorithm 1: AdaBoost**

1. Initialize the observation weights  $w_i = \frac{1}{n}$ ,  
where  $i = \{1, 2, \dots, n\}$
  2. For  $m = 1$  to  $M$ :
    - a) Fit a classifier  $\tilde{T}^m(x)$  to the training data using weights  $w_i$ .
    - b) Compute  $err^{(m)} = \frac{\sum_1^n w_i I(c_i \neq \tilde{T}^m(x_i))}{\sum_1^n w_i}$
    - c) Compute  $\alpha^m = \log\left(\frac{1 - err^{(m)}}{err^{(m)}}\right)$
    - d) Set  $w_i \leftarrow w_i \cdot e^{\alpha^{(m)} \cdot I(c_i \neq \tilde{T}^m(x_i))}$   
for  $i = \{1, 2, 3, \dots, n\}$
    - e) Re-normalize  $w_i$ .
  3. Output  $C(x) = \arg \max_k \sum_{m=1}^M \alpha^{(m)} \cdot I(\tilde{T}^m(x) = k)$
- (a)

**Algorithm 2: Multiclass AdaBoost [13]**

- Stagewise Additive Modeling using a  
Multi-class Exponential loss function.
1. Initialize the observation weights  $w_i = \frac{1}{n}$ ,  
where  $i = \{1, 2, \dots, n\}$
  2. For  $m = 1$  to  $M$ :
    - a) Fit a classifier  $\tilde{T}^m(x)$  to the training data using weights  $w_i$ .
    - b) Compute  $err^{(m)} = \frac{\sum_1^n w_i I(c_i \neq \tilde{T}^m(x_i))}{\sum_1^n w_i}$
    - c) Compute  $\alpha^m = \log\left(\frac{1 - err^{(m)}}{err^{(m)}}\right) + \log K - 1$
    - d) Set  $w_i \leftarrow w_i \cdot e^{\alpha^{(m)} \cdot I(c_i \neq \tilde{T}^m(x_i))}$   
for  $i = \{1, 2, 3, \dots, n\}$
    - e) Re-normalize  $w_i$ .
  3. Output  $C(x) = \arg \max_k \sum_{m=1}^M \alpha^{(m)} \cdot I(\tilde{T}^m(x) = k)$
- (b)

**Figure 1.** Representations of two AdaBoost Methods as algorithms (a) Shown algorithm of AdaBoost, (b) Shows Multiclass AdaBoost [19].

### 2.3. Feature Extraction and Selection

Features are the most crucial component of a Supervised Machine Learning Model. They represent the structure and information content in the dataset. Since our model is not based on convolution operators, i.e., there is no direct convolution-based learning in our network, images will not be processed directly, as the pixel-intensity values of images and hence numeric-type features need to be fed. As there is a large variety of features that can be used for this purpose, one must be cautious and make sure they are highly informative and distinct for efficient convergence of a network. It must be noted that for the task of recognizing plant species from corresponding leaf images, the most important characteristics observable in a leaf are shape- or structure-based, the others being botanical characteristics like vein structure. There have been a number of studies in botanical sciences that prove the fact that although vein structure is important in distinguishing plant species, it cannot be entirely relied upon for the task, as some patterns in these structures in different species of plants have also been noticed. Certainly, this does not play a major role in the dataset used in this study, as can be applied in general to flora only; yet, we considered this fact, and with the success of previous studies in similar tasks utilizing shape-based features, we fed our model with morphological features, like minor and major axis length, solidity, perimeter, centroid, and orientation, which are discussed in detail in this section.

Solidity is area fraction of the region, compared to its convex hull or the ratio of the pixels that are present in both the object and the convex hull. The more bays and corners the object has, the lower its solidity. Its mathematical formulation is

$$S = \frac{\text{Area}}{\text{Convex Area}} \quad (2)$$

Solidity-filled images consist of single color. Generally, the leaf images are of two types. The first is hollow, wherein pixels around the focus of mass are just halfway filled—for



example, small pixels are left unfilled, and the one in which all pixels around center of mass are loaded up with colors of high intensity.

$$\frac{c}{s} = \frac{10}{\left(\frac{D_h}{D_i}\right) \left(\frac{N_s}{1000}\right)^{1.5}} \quad (3)$$

where,  $c$  is centroid,  $s$  represents solidity,  $D_h$  is the diameter, and  $N_s$  is the pixel count in a specific region.

While interfacing all centroid points, a line is drawn. The entire image's focal point of mass stands around this centroid point.

A centroid is interpreted as the center of mass of a region or area. The centroid of the polygon formed can be found by segmenting the image into smaller regions, and each small region can have its individual centroid point. Summing up these individual centroids, the centroid of a polygon is calculated using

$$C_x = \frac{\sum C_{ix} A_i}{A_i} \quad (4a)$$

$$C_y = \frac{\sum C_{iy} A_i}{A_i} \quad (4b)$$

here,  $C_x$  and  $C_y$  represent centroids where  $A_i$  is the contour area, given as

$$A = \frac{1}{2} \sum_{i=1}^{N-1} x_i y_{i+1} + x_{i+1} y_i \quad (5)$$

The major axis length is a line that connects one end, referred to as the base point, to tip of leaf. The line is drawn to two points that are selected. This represents the main perpendicular axis of image. This main axis length calculates length of image in width-wise as

$$\text{Major Axis Length} = \frac{(x_1 - x_c)^2}{rx^2} + \frac{(y_1 - y_c)^2}{ry^2} \quad (6)$$

where  $x_1$  and  $y_1$  are points along the major axis,  $x_c$  and  $y_c$  are center points, and  $r_y$  and  $r_x$  represent the radius along the  $x$ -axis and  $y$ -axis, respectively. The minor axis is a line drawn perpendicular to the major axis.

The perimeter or circumference of region  $\check{R}$  represents the path length of any shape externally. To determine the perimeter, the whole of separations between progressive limit pixels are estimated in equation below. The basic measure of the perimeter is attained by calculating the number of limit pixels that belong to an object. The perimeter is the total diameter that surrounds the image of leaf. The total number of pixels across boundary points will show the total number of pixels utilized to fill the boundary pixels.

$$P = 2L + 2W \quad (7)$$

where  $P$  is the perimeter,  $L$  is the length of the major axis, and  $W$  is the length of the minor axis.

Orientation is measurement of the angle between the  $x$ -axis and the main axis of ellipse. It displays the orientation of the picture with the main axis and minor axis. The direction of coordinate axis will immediately minimize the length of the major and minor axis, as seen in the equation below.

$$O = \cos m_j + \sin m_j + \sqrt{\tan(m_j \times m_i)} \quad (8)$$

where  $O$  is the orientation,  $m_i$  is the length of the minor axis, and  $m_j$  is the length of the major axis.

Other than these morphological features, we used the power of transfer learning and deep convolutional neural networks for our task. For this, first we implemented the traditional transfer learning technique to feed the proposed classifier model, which can be highlighted as first considering the convolutional network as a fixed feature extractor, and using other classifiers like logistic regression or SVM for classification tasks. Later, we fine-tuned the convolutional network by first training the last  $Q$  levels with  $N - Q$  fewer levels frozen (the higher the level, the lower will be the overfitting to target data) [21].

Primarily, there were two possibilities as options of the architecture that would already be trained against the renowned ImageNet, and whose weights were available online for use. These were VGGNet and the ResNet50. Also, it is an unavoidable fact that the ImageNet dataset is not related or similar to the dataset used for this task, though ImageNet, in its total collection of 21m841 synsets, includes 70 leaf-related synsets. Therefore, the usage of transfer learning is not an ideal decision. However, Yosinski et al. stated and proved experimentally that even features transferred from distinct tasks are improved compared to random weights [22]. In addition, certain later layers of these architectures are not used for transferring features, which further disregards the non-ideal condition mentioned above, because earlier layers are more generic when it comes to features, while the later layers are more dataset-specific [21]. ResNet50 was chosen for this task, as it provides less overfitting and better results [23]. It is relatively deep and less complex [24]. With the pooling size of  $3 \times 7 \times 7$ , it generates feature vectors of lower dimensions. ResNet50 consists of 49 convolutional layers with one fully-connected layer. Forty-eight convolutional layers compose 16 “residual” blocks in four stages. Like in [23], we also used the bottleneck features or the CNN codes (in the terminology of transfer learning). Also, the layer after the last convolutional or residual block has to be dropped off. These extracted feature vectors are of the 2048 dimension. We acknowledge Yosinski et al. [22] for good insights into transfer learning and its use for this task.

A code snippet as the MATLAB implementation algorithm for visually depicting a leaf sample image in the dataset in different stages of processing can be observed in Figure 2 below. Subsequent images printed on the output screen after running the code as in Figure 2 are shown in Figure 3.

### **Algorithm 3. Pre-Processing, Resizing Images, extracting Morphological Features**

```

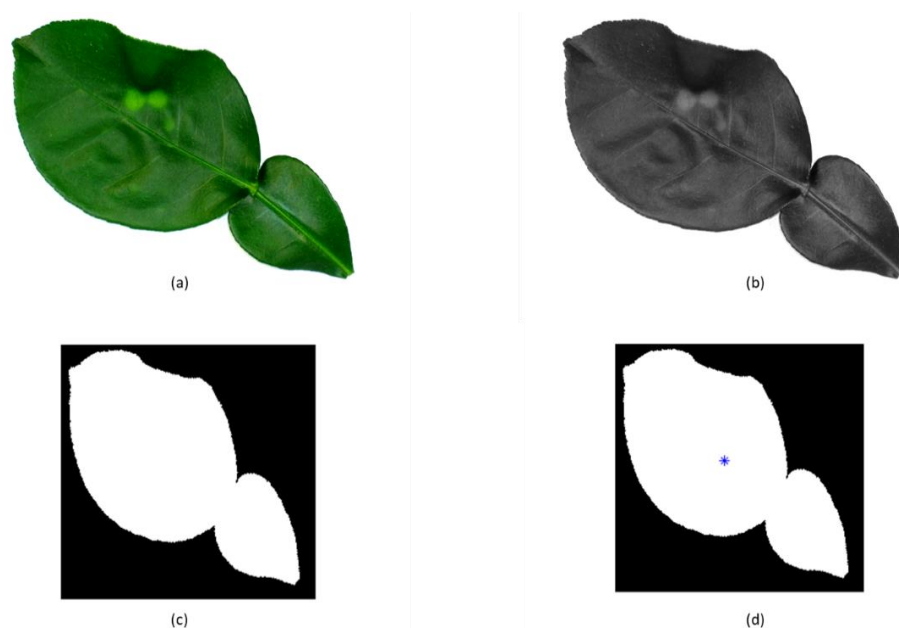
Convert the RGB image into Binary Format and Resize to [256 256]
rgb_img=imread("dataset/train_images/3621.jpg"); imshow(rgb_img)
gry_img = rgb2gray(rgb_img); imshow(gry_img)
resized = imresize(gry_img,[256,256]);
mask = zeros(size(resized));
mask(25:end-25,25:end-25) = 1;
binary_img = activecontour(resized,mask,400); imshow(binary_img)

Extract Features and display:

features =
regionprops(bwconncomp(binary_img),'Area','MinorAxisLength','MajorAxisLength',
"Centroid","Orientation","Perimeter","Solidity");
centroids = cat(1,features.Centroid);
figure
imshow(binary_img)
hold on
plot(centroids(:,1),centroids(:,2),'b*')
struct2table(features)

```

**Figure 2.** Shows code snippet implemented on MATLAB for the processing of images before feeding into the model.



**Figure 3.** Leaf sample image in different stages of pre-processing. (a) Original RGB Image (Size: [1200, 1600]). (b) Grayscale Converted Image (Size: [1200, 1600]). (c) Binary Converted and Resized Image (Size: [256, 256]). (d) Centroid location of the leaf sample

#### 2.4. Model Highlights

To sum up this study, we have used the multiclass support vector machine as a classifier and introduced a non-linear kernel for unavoidable reasons in the separability of the datapoints. To improve the network testing results, we have implemented the multiclass AdaBoost. Though AdaBoost works efficiently with relatively weak classifiers for improving the results, and SVMs are strong classifiers, we used AdaBoost because there have been various studies that show SVM to be a weak learner (a little better than a coin toss, in such cases), with some other modifications can act optimally with AdaBoost [25–27].

For the input data, we extracted morphological features from the leaf images in the available open-source FLAVIA dataset, as well as the features from the transfer learning technique. These features are described in Table 1 below.

**Table 1.** Features used for training the classifier.

Feature	Explanation	Formula
Solidity	Area fraction of the region compared to its convex hull, or the extent of the pixels in the convex hull that is additionally in the area.	$S = \frac{Area}{Convex\ Area}$
Centroid	Center of mass of a region or area.	$C_x = \frac{\sum C_{ix} A_i}{A_i}; C_y = \frac{\sum C_{iy} A_i}{A_i}$ $A = \frac{1}{2} \sum_{i=1}^{N-1} x_i y_{i+1} + x_{i+1} y_i$
Perimeter	Path length of the shape externally.	$P = 2L + 2P$
Major axis length	Line connecting one end, called a base point, to the tip of the leaf.	$Major\ Axis\ Length = \frac{(x_1 - x_c)^2}{rx^2} + \frac{(y_1 - y_c)^2}{ry^2}$
Minor axis length	Line drawn perpendicular to the major axis.	$\left( \left( \frac{(x_1 - x_c)^2}{rx^2} + \frac{(y_1 - y_c)^2}{ry^2} \right) \right)^{\perp}$
Orientation	Angle between the x-axis and the major axis of the ellipse.	$O = \cos m_j + \sin m_j + \sqrt{\tan(m_j \times m_i)}$
ConvNet via transfer learning	Features extracted from ResNet50 via transfer learning	N/A

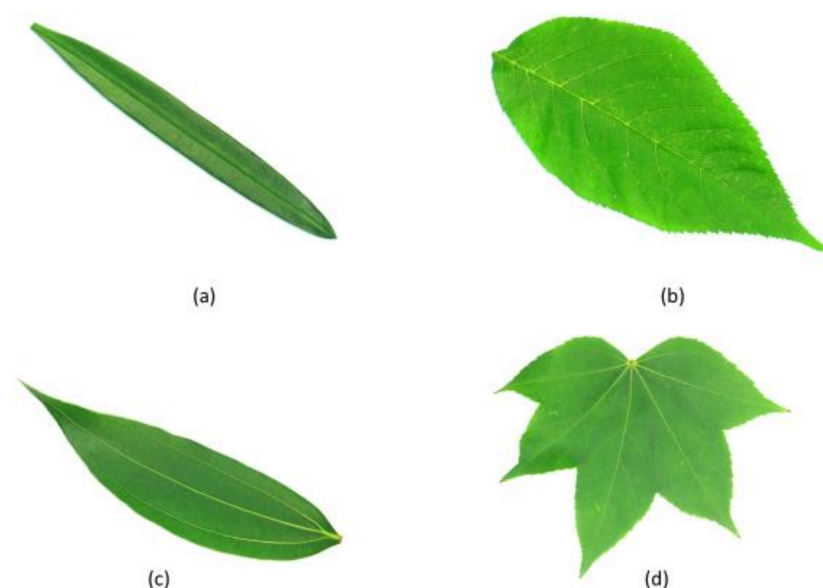
The parameters, decisive elements of the methodology, have been summarized in Table 2 below.

**Table 2.** The hyperparameters and elements of the methodology selected.

Element	Value/Detail
Fundamental algorithm	Support vector machine (polynomial kernel)
Secondary algorithm	Multiclass adaptive boosting
Dataset	FLAVIA (32 classes distributed among 1907 instances)
Features	Morphological and from transfer learning (ResNet50)
Partitioning strategy	80:20 split, five-fold CV and 10-fold CV (best: 10-fold CV)
Fundamental model	Support vector machine and adaptive boosting

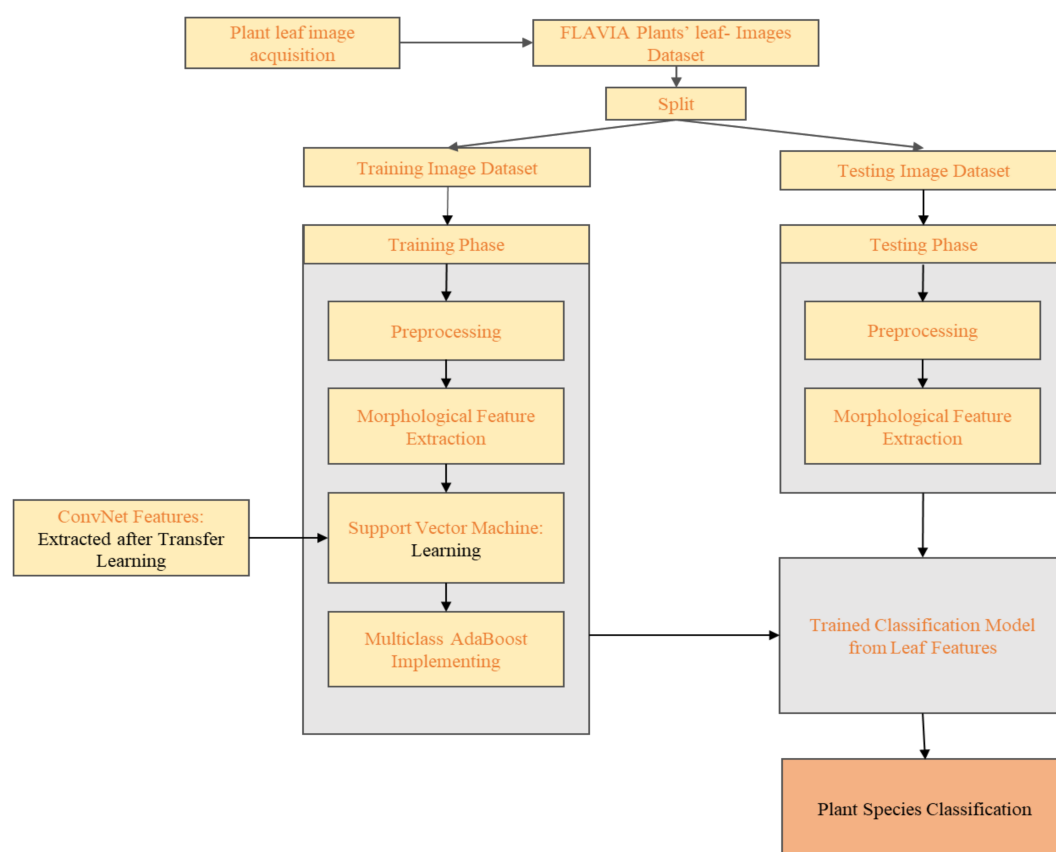
### 3. Data and Training

The open-source dataset [28] that we used contains 1907 leaf images of 32 different species of plants. Samples from that dataset are shown in Figure 4 below. With the availability of good quality photography, the images are of good quality with a white/transparent background, with almost no signs of pixel deformity and little or no variations of luminance or color. The features extracted from these images, as discussed in Section 2.3, were highly scattered and non-separable. Therefore, we had to use a non-linear kernel with the multiclass SVM. Sangeetha and Kalpana [29] provided good insights about selection of kernel functions for optimal performance in a multiclass SVM. Based on that study, and the dataset structure used for this study, we used the polynomial kernel function.



**Figure 4.** Shows samples from the FLAVIA leaf image dataset (a) Image of species, label *Podocarpus macrophyllus* (Thunb.) Sweet. (b) Image of species, label *Prunus serrulata* Lindl. var. *lannesiana* auct. (c) Image of species, label *Cinnamomum japonicum* Sieb. (d) Image species, label *Kalopanax* s. Koidz.

The implemented architecture has been represented using the flow chart in Figure 5 below.



**Figure 5.** Shows the flow diagram of the implemented algorithm

#### 4. Evaluation and Results

We performed experiments with same dataset and its partitioning strategy with three other previously proposed models using kNN, MLP + AdaBoost, and decision tree, to provide sufficient data that can quantitatively validate the credibility of this study. To show the validity of the proposed system, as well as the decisions like AdaBoost and transfer learning, we did experiments in the absence and presence of these models. Firstly, an 80/20 approach was used, in which 80% of the images are considered randomly as a training dataset, and remaining 20% are considered as a testing dataset. Another approach is 5-fold and 10-fold cross-validation. In five-fold cross-validation, the whole dataset is randomly partitioned into five groups. Training and testing are done as 3:2 on these groups iteratively. A similar approach is used for the 10-fold cross-validation. Firstly, we compared the proposed model with three other models, as their precision rates vs dataset partitioning strategy is as depicted in Figure 6 below. These results indicate that our framework outperforms the other three while utilizing 10-fold validation. This is also evidence of the fact that the MLP + AdaBoost performs slightly better with the five-fold cross-validation strategy, and is nearly consistent with varying dataset partitioning strategies. Secondly, the confusion matrix of proposed model is shown below in Figure 7.



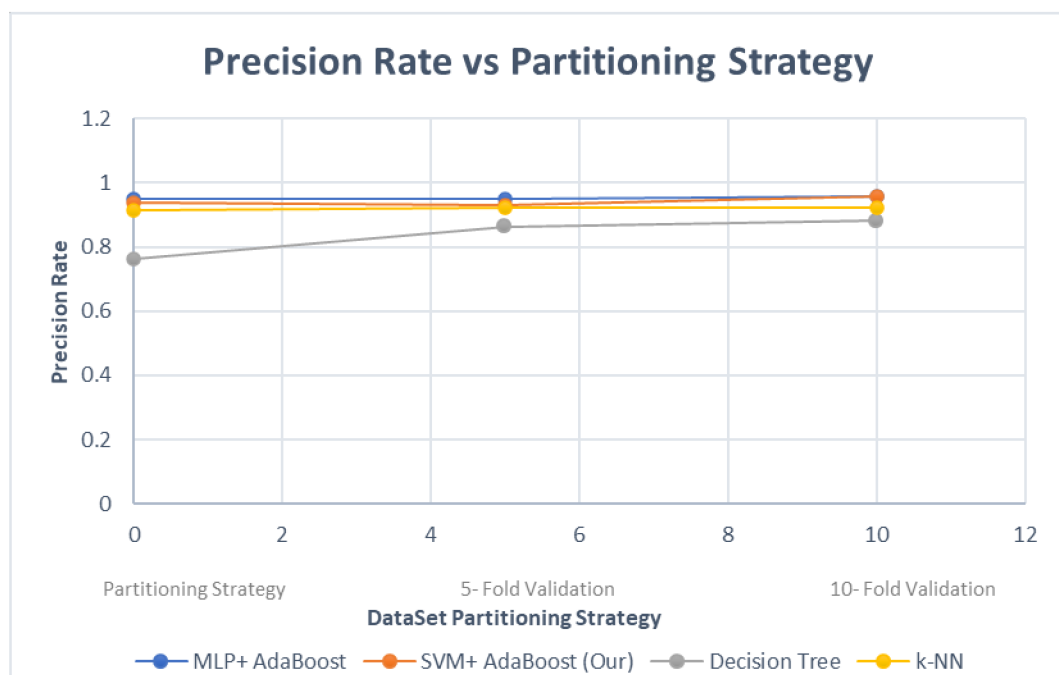


Figure 6. Precision rates vs dataset partitioning strategy for four different models.

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	aa	ab	ac	ad	ae	af	← classified as
9	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Phyllanthus edulis (Carr.) Hou.
0	9	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	2	0	0	Aecleus chinensis
0	0	7	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Bebetis andweensis Alendri
0	0	0	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Cercis chinensis
0	0	0	0	9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Indigofera tinctoria L.
0	0	0	0	0	7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Acer Palmatum
0	1	0	0	0	0	7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Phoebe nana (Oliv.) Gamble
0	0	0	0	0	0	0	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Kalopanax septemlobus (Thunb. ex A.Murr.) Koidz.
0	0	0	0	0	0	0	0	11	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Cinnamomum japonicum Sieb.
0	0	0	0	0	0	0	0	0	11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Koeleria paniculata Laxm.
0	0	0	0	0	0	0	0	0	0	11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Ilex macrocarpa Oliv.
0	0	0	0	0	0	0	0	0	0	0	11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Pittosporum tobira (Thunb.) Ait. f.
0	0	0	0	0	0	0	0	0	0	0	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Chimonanthus praecox L.
1	0	0	0	0	0	0	0	0	0	1	0	0	12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Cinnamomum camphora (L.) J. Presl
0	0	0	0	0	0	0	0	0	0	0	0	0	0	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Viburnum avahiki K.Koch
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Osmorhiza fragrans Lour.
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	15	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Cedrus deodora (Roth.) G. Don
0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Ginkgo biloba L.
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	11	0	0	0	0	0	0	0	0	0	0	0	0	0	Lagerstroemia indica (L.) Pers.
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	13	0	0	0	0	0	0	0	0	0	0	0	0	Podocarpus nagebiellus (Thunb.) Sweet
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	12	0	0	0	0	0	0	0	0	0	0	0	Prunus serotina Lindl. var. lamellosa auct.
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	13	0	0	0	0	0	0	0	0	0	0	Ligustrum lucidum Ait. f.
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10	0	0	0	0	0	0	0	0	0	Torreia sinensis M. Roem.
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	11	0	0	0	0	0	0	0	0	Prunus persica (L.) Batsch
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8	0	0	0	0	0	0	0	Manglietia fordiana Oliv.
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	11	0	0	0	0	0	Acer buergerianum Miq.
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Mahonia bealei (Fortune) Carr.
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Magnolia grandiflora L.
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Populus canadensis Meisch
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Liriodendron chinense (Hems.) Sarg.
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Citrus reticulata Blanco

Figure 7. The model's performance as a confusion matrix obtained from the proposed model.

It can be computed from this confusion matrix that the overall accuracy of the framework on all 32 classes is

$$Acc \% = \left( \frac{323}{323 + 18} \right) \times 100 = 94.72141\% \quad (9)$$

If compared with the CNN-based study [6], in which the recognition rate (RR) achieved was almost above 94%, our framework still outperforms. The authors in this study used convolution neural network-based classifier Google Net, which is considered to be a fairly well-performing classifier. Also, considering the fact that these classifiers are developed and used specifically for visual data, and hence should be more dedicated to the problem at hand, this is a clear indication of the robustness of the proposed architecture in this article. Next, Sun et al. [13] achieved 91.78% RR with their 26-layered CNN network, which further evidence of the credibility of our framework. An accuracy of 94.22% was achieved by Zhang et al. [3] using BOF and DPCNN, and is a close competitor, but our methodology validates its effectiveness by its outperformance. An overall analysis of these studies in contrast to ours has been tabulated below in Table 3.

**Table 3.** Accuracy values of various research studies.

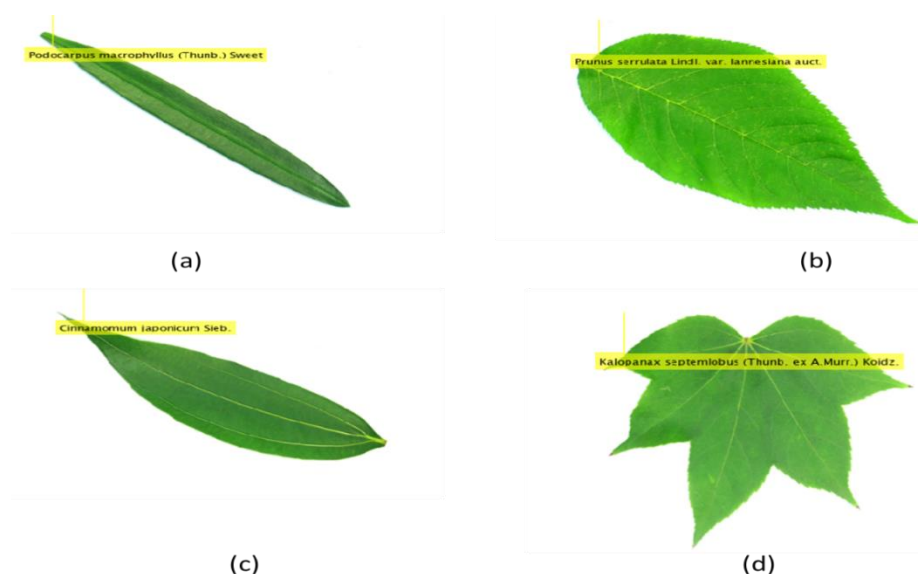
Serial	Research Study	Accuracy (%)
1	CNN-based [6]	94.00
2	CNN-based [13]	91.78
3	Bag of features-based [3]	94.22
4	Our study (hybrid)	94.72

The precision rates of each of the 32 classes in the FLAVIA dataset are shown in the Table 4 below.

**Table 4.** Shows values of each individual class in the dataset.

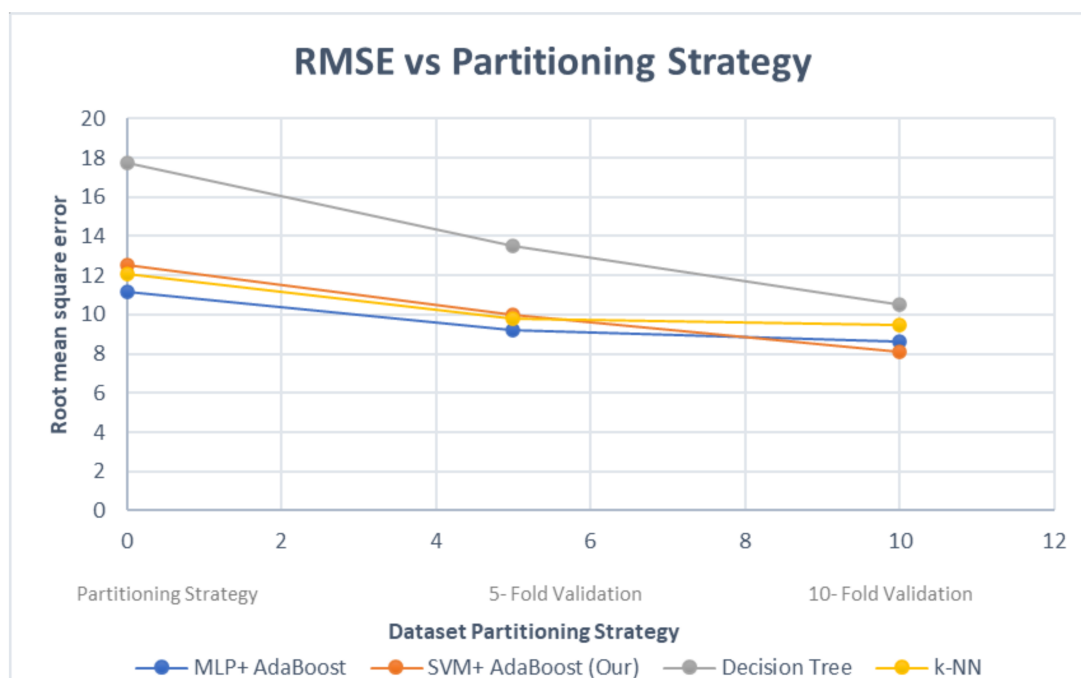
Serial	Class	Precision	Class	Precision
1	<i>Phyllostachys edulis</i> (Carr.) Houz.	1.0000	<i>Cedrus deodara</i> (Roxb.) G. Don	0.93750
2	<i>Aesculus chinensis</i>	0.9000	<i>Ginkgo biloba</i> L.	0.92860
3	<i>Berberis anhwaiensis</i> Ahrendt	1.0000	<i>Lagerstroemia indica</i> (L.) Pers.	0.91660
4	<i>Cercis chinensis</i>	0.9090	<i>Nerium oleander</i> L.	0.92860
5	<i>Indigofera tinctoria</i> L.	0.9000	<i>Podocarpus macrophyllus</i> (Thunb.) Sweet	0.92300
6	<i>Acer palmatum</i>	1.0000	<i>Prunus s. Lindl var. l. auct.</i>	0.92857
7	<i>Phoebe nanmu</i> (Oliv.) Gamble	1.0000	<i>Ligustrum lucidum</i> Ait. f.	1.00000
8	<i>Kalopanax s. Koidz.</i>	1.0000	<i>Tonna sinensis</i> M. Roem.	1.00000
9	<i>Cinnamomum japonicum</i> Sieb.	1.0000	<i>Prunus persica</i> (L.) Batsch	1.000000
10	<i>Koeleruteria paniculata</i> Laxm.	0.9166	<i>Manglietia fordiana</i> Oliv.	0.90000
11	<i>Ilex macrocarpa</i> Oliv.	0.9166	<i>Acer buergerianum</i> Miq.	1.00000
12	<i>Pittosporum tobira</i> Ait. f.	1.0000	<i>Mahonia bealei</i> (Fortune) Carr.	0.90900
13	<i>Chimonanthus praecox</i> L.	0.9090	<i>Magnolia grandiflora</i> L.	1.00000
14	<i>Cinnamomum camphora</i> (L.) J. Presl	0.9231	<i>Populus × canadensis</i> Moench	0.92307
15	<i>Viburnum awabuki</i> K. Koch	1.0000	<i>Liriodendron chinense</i> Sarg.	1.00000
16	<i>Osmanthus fragrans</i> Lour.	1.0000	<i>Citrus reticulata</i> Blanco	1.00000

The same leaf samples shown in Figure 4, resulting from model, are shown in Figure 8 below.



**Figure 8.** Model's output samples of four different species of leaves. (a) Output of *Podocarpus macrophyllus* (Thunb.) Sweet class, (b) Output of *Prunus s.* Lindl var. l. auct., (c) *Cinnamomum japonicum* Sieb. (d) *Kalopanax s.* Koidz.

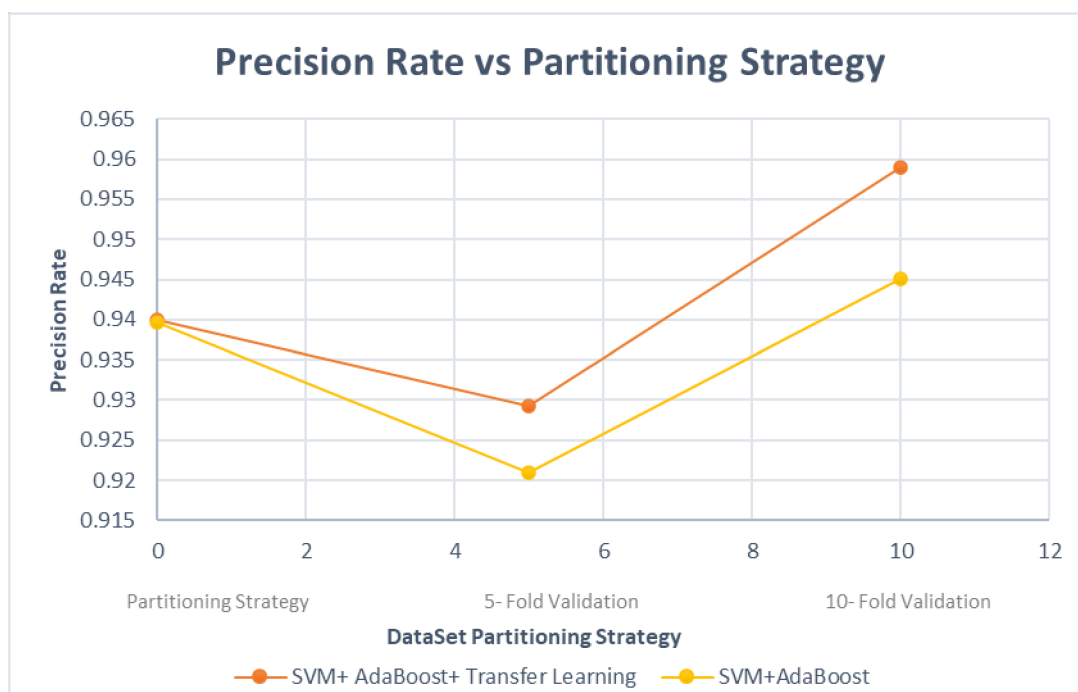
Next, we compared the root mean squared error (RMSE) of all models at different partitioning strategies, and as depicted in Figure 9 below, the RMSE falls most gradually for the kNN, while the rate of fall is almost consistent with the other three models, with the lowest being for our framework at 10-fold CV.



**Figure 9.** All four models' performance as root mean squared error (RMSE) vs partitioning strategy of the proposed systems.

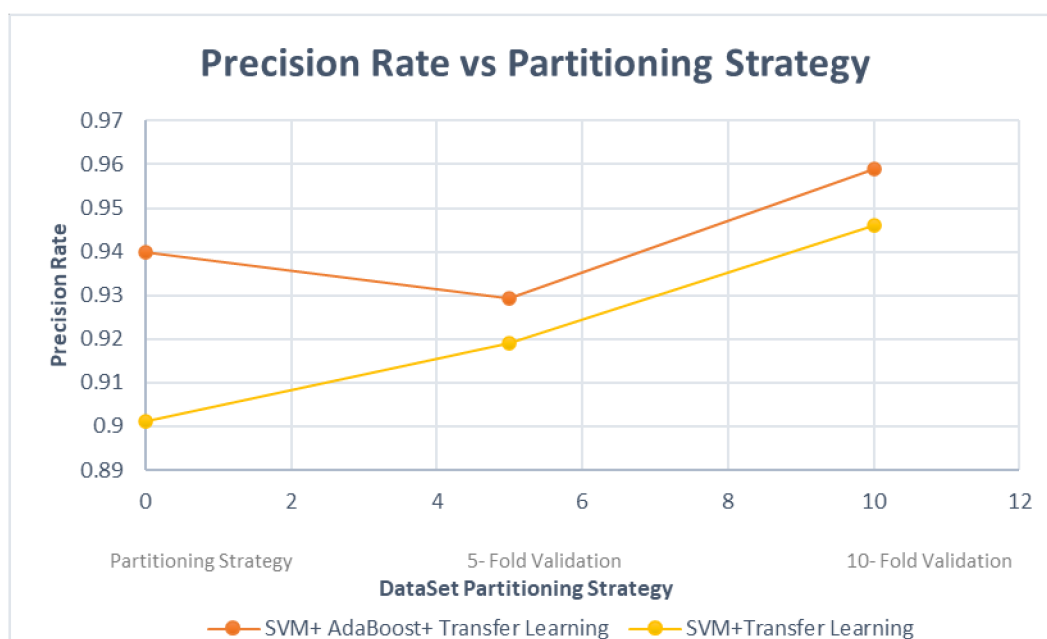
In another experiment, aiming to prove the validity of transfer learning for extracting features, as mentioned in previous sections, we found that the proposed architecture performs better with the introduction of these additional transfer learning-based features. This can be proven from the plot in Figure 10 below, in which the flow of model's performance has been depicted. It is certain from this plot that the decision for including CNNs is vital in the consolidated performance of the methodology. Also, the precision rate of the

methodology without transfer learning or CNNs is also considerably good. This also clarifies the fact that the high superiority in performance of the methodology is primarily dependent upon transfer learning or CNNs.



**Figure 10.** Models' performance in presence and absence of transfer learning features.

Finally, we did similar experiments in the absence of AdaBoost, using the strong SVM classifier itself, and the results are depicted in the Figure 11 below. These results clearly indicate that although by marginal amounts, the adaptive boosting strategy plays a role in enhancing the framework.



**Figure 11.** Models' performance in the presence and absence of AdaBoost.

## 5. Credibility of the Methodology

It is an undeniable fact that the SVM was first introduced way before some of the high-performing modern classifier models. Even if some modifications to the SVM are considered, it still might not outperform some of the CNN-based networks for most tasks. This study is primarily aimed at improving SVM implementation for plant species recognition. For example, the architecture proposed includes the use of AdaBoost, which is not conventionally preferred with SVM (as discussed in this paper already), yet we observed some improvements while using it. There already have been numerous options proposed at this task clearly, but the architecture in our study suggests that even traditional networks like SVM can compete with some of the well-performing models of the time, particularly for this task. This will enable researchers to go not only for CNN-based models while working on such problems, because as the results indicate, there is a high future scope of such models, though only with potentially useful modifications. Also, we utilized one of the potentials that CNNs exhibit, i.e., extracting features from images, with the decision to use transfer learning. In effect, we enhanced a traditional classifier like SVM, utilizing CNNs and some other modifications as well. Lastly, in the manuscript, as the conclusion and Priya et al. [10] also state, SVMs are disadvantageous with speed and size constraints, as well as with complex algorithmic structures, whereas CNNs are disadvantageous with intensive computations and are incapable of better generalization. Clearly, for the task at hand—plant species recognition—there arises no issue of urgency to detect the class of a plant, i.e., the primary disadvantages of SVM are not much of a loss for the task, and moreover, after the modifications proposed, this quantification of disadvantages gets even lower. However, this is not the case with CNNs.

## 6. Conclusions

In this study, we presented an efficient and robust plant species classification model using features extracted from leaves, transfer learning, and adaptive boosting. We also experimented with the model for realizing the effects to final results with the presence and absence of some architectural highlights, such as AdaBoost and transfer learning, to provide supplementary evidence of the right choices for these decisions, thereby making this proposal a novel study. In this work, an average precision rate of 95.85% was achieved for 32 plant species. This gives this model better performance than most existing models for the task. This study has the full potential to be extended, and be used in medicinal or agricultural research.

With regard to the limitations of this study, as already discussed above, SVMs might be disadvantageous in terms of their speed and size constraints. Although this does not play a primary role in improvements to this framework, it can be considered. Secondly, as also discussed in the manuscript, the enhanced AdaBoost technique was utilized, although boosting in general is not preferred with SVMs. Therefore, a future scope of our framework would be to develop a better boosting strategy for the SVM that also lightens the processing/computations to be done by it. Also, the data is not processed much before feeding into the classification modules in the framework. Although most of the existing methods for this task will not make much of a difference, a customized and dedicated block for processing data before classification can be added to the overall flow of the framework. Our model also does not specifically account for the fact that the sheet can be angled to the scan plane. We did not experiment with the possibility of geometric distortions or deformations of images due to non-ideal camera angles. In addition, surely “affinity” might solve the problem in that case. Certainly, this is an interesting aspect of the study to be worked upon.

**Author Contributions:** Conceptualization, A.R., S.M. and A.K.P.; methodology, S.M.; software, S.M. and A.K.P.; validation, S.M., A.R., X.-Z.G. and A.K.P.; formal analysis, S.M. and A.K.P.; investigation, S.M. and A.K.P.; resources, S.M. and A.K.P.; data curation, S.M. and A.K.P.; writing—original draft preparation, S.M. and A.K.P.; writing—review and editing, S.M. and A.K.P. and X.-Z.G.; visualization, S.M.; supervision, A.K.P.; project administration, A.K.P.; funding acquisition, X.-Z.G. All authors have read and agreed to the published version of the manuscript.



**Funding:** The authors received no specific funding for this study.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

**Acknowledgments:** I would like to acknowledge almighty GOD, my parents, friends, my supervisor, and all the co-authors for valuable suggestions and support.

**Conflicts of Interest:** The authors declare that they have no conflicts of interest to report regarding the present study.

## References

1. Sun, R.; Zhang, M.; Yang, K.; Liu, J. Data enhancement for plant disease classification using generated lesions. *Appl. Sci.* **2020**, *10*, 466.
2. Yang, K.; Zhong, W.; Li, F. Leaf segmentation and classification with a complicated background using deep learning. *Agronomy* **2020**, *10*, 1721. [CrossRef]
3. Zhang, Y.; Cui, J.; Wang, Z.; Kang, J.; Min, Y. Leaf Image Recognition Based on Bag of Features. *Appl. Sci.* **2020**, *10*, 5177. [CrossRef]
4. Azlah, M.A.; Chua, L.S.; Rahmad, F.R.; Abdullah, F.I.; Wan Alwi, S.R. Review on techniques for plant leaf classification and recognition. *Computers* **2019**, *8*, 77. [CrossRef]
5. Kumar, M.; Gupta, S.; Gao, X.Z.; Singh, A. Plant Species Recognition Using Morphological Features and Adaptive Boosting Methodology. *IEEE Access* **2019**, *7*, 163912–163918. [CrossRef]
6. Jeon, W.S.; Rhee, S.Y. Plant leaf recognition using a convolution neural network. *Int. J. Fuzzy Log. Intell. Syst.* **2017**, *17*, 26–34.
7. Pankaja, K.; Suma, V. Leaf Recognition and Classification Using GLCM and Hierarchical Centroid Based Technique. In Proceedings of the 2018 International Conference on Inventive Research in Computing Applications (ICIRCA), Coimbatore, India, 11–12 July 2018; IEEE: Piscataway, NJ, USA; pp. 1190–1194.
8. Kumar, S. Leaf Color, Area and Edge features-based approach for Identification of Indian Medicinal Plants. *Int. J. Comput. Sci. Eng.* **2012**, *3*, 436–442.
9. Kadir, A.; Nugroho, L.E.; Susanto, A.; Santosa, P.I. Performance improvement of leaf identification system using principal component analysis. *Int. J. Adv. Sci. Technol.* **2012**, *44*, 113–124.
10. Priya, C.A.; Balasaravanan, T.; Thanamani, A.S. An efficient leaf recognition algorithm for plant classification using support vector machine. In Proceedings of the International conference on pattern recognition, informatics and medical engineering (PRIME-2012), Salem, India, 21–23 March 2012; IEEE: Piscataway, NJ, USA; pp. 428–432.
11. Anami, B.S.; Nandyal, S.S.; Govardhan, A. A combined color, texture and edge features-based approach for identification and classification of indian medicinal plants. *Int. J. Comput. Appl.* **2010**, *6*, 45–51. [CrossRef]
12. Sambhaji, E.S.; Andore, D.B. Leaf recognition algorithm using neural network-based image processing. *Asian J. Eng. Technol. Innov.* **2014**, *2*, 10–16.
13. Sun, Y.; Liu, Y.; Wang, G.; Zhang, H. Deep learning for plant identification in natural environment. *Comput. Intell. Neurosci.* **2017**, *22*, 2017. [CrossRef] [PubMed]
14. Cortes, C.; Vapnik, V. Support-vector networks. *Mach. Learn.* **1995**, *20*, 273–297. [CrossRef]
15. Duan, K.B.; Keerthi, S.S. Which is the best multiclass SVM method? An empirical study. In *International Workshop on Multiple Classifier Systems, Proceedings of the Multiple Classifier Systems*; Springer: Berlin/Heidelberg, Germany, 2005; pp. 278–285.
16. Hastie, T.; Tibshirani, R. Classification by pairwise coupling. In *Advances in Neural Information Processing Systems*; MIT Press: Cambridge, MA, USA, 1998; pp. 507–513.
17. Freund, Y.; Schapire, R.E. A decision-theoretic generalization of on-line learning and an application to boosting. *J. Comput. Syst. Sci.* **1997**, *55*, 119–139. [CrossRef]
18. Boosting Algorithms: AdaBoost, Gradient Boosting and XGBoost. 5 May 2018. Available online: [hackernoon.com](https://hackernoon.com) (accessed on 4 January 2020).
19. Hastie, T.; Rosset, S.; Zhu, J.; Zou, H. Multi-class adaboost. *Stat. Its Interface* **2009**, *2*, 349–360. [CrossRef]
20. Kim, T.H.; Park, D.C.; Woo, D.M.; Jeong, T.; Min, S.Y. Multi-class classifier-based adaboost algorithm. In *International Conference on Intelligent Science and Intelligent Data Engineering, Proceedings of the Intelligent Science and Intelligent Data Engineering*; Springer: Berlin/Heidelberg, Germany, 2011; pp. 122–127.
21. CS231n: Convolutional Neural Networks for Visual Recognition. Available online: <http://cs231n.github.io/transfer-learning/> (accessed on 10 January 2021).
22. Yosinski, J.; Clune, J.; Bengio, Y.; Lipson, H. How transferable are features in deep neural networks? *arXiv* **2014**, arXiv:1411.1792.

- 
23. Plant Leaf Recognition. Available online: <http://cs229.stanford.edu/proj2016/report/LiuHuang-PlantLeafRecognition-report.pdf> (accessed on 5 January 2021).
  24. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015.
  25. Li, X.; Wang, L.; Sung, E. AdaBoost with SVM-based component classifiers. *Eng. Appl. Artif. Intell.* **2008**, *21*, 785–795. [[CrossRef](#)]
  26. Govindaraj, D. Can Boosting with SVM as Weak Learners Help? *arXiv* **2016**, preprint. arXiv:1604.05242.
  27. García, E.; Lozano, F. Boosting Support Vector Machines. *MLDM Posters*. 2007, pp. 153–167. Available online: <https://www.sciencedirect.com/science/article/abs/pii/S0957417420301457> (accessed on 10 January 2020).
  28. Flavia (at a Glance). Available online: <http://flavia.sourceforge.net/> (accessed on 2 January 2021).
  29. Sangeetha, R.; Kalpana, D.B. Identifying efficient kernel function in multiclass support vector machines. *Int. J. Comput. Appl.* **2011**, *28*, 18–23. [[CrossRef](#)]