PLANT SPECIES IDENTIFICATION

A Project Report
Submitted in the partial fulfilment for the award of the degree of

BACHELOR OF ENGINEERING
IN

C.S.E. (BIG DATA ANALYTICS)

&

C.S.E. (ARTIFICAL INTELENGENCE MACHINE LEARNING)

Submitted by:

KAKKIRALA LOHITH KUMAR	20BCS3880
ANKIT GHOSAL	20BCS6621
LALIT BISHT	20BCS6609

Under the Supervision of:

Ms. MERRY

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

APEX INSTITUTE OF TECHNOLOGY

CHANDIGARH UNIVERSITY, GHARUAN, MOHALI – 140413,

PUNJAB

NOVEMBER 2022

DECLARATION

We, *Kakkirala Lohith kumar, Ankit Ghosal, Lalit Bisht* of fifth semester B.Tech., in the department of Computer Science and Engineering from Chandigarh University, hereby declare that the project work entitled "***PLANT SPECIES IDENTIFICATION***" is carried out by us and submitted in partial fulfillment of the requirements for the award of Bachelor of Technology in Computer Science Engineering, under Chandigarh University (Apex institute of Technology) during the academic year 2020-2024.

ABSTRACT

Species knowledge is essential for protecting biodiversity. The identification of plants by conventional keys is complex, time consuming, and due to the use of specific botanical terms frustrating for non-experts. This creates a hard to overcome hurdle for novices interested in acquiring species knowledge. Today, there is an increasing interest in automating the process of species identification. The availability and ubiquity of relevant technologies, such as, digital cameras and mobile devices, the remote access to databases, new techniques in image processing and pattern recognition let the idea of automated species identification become reality. This paper is the first systematic literature review with the aim of a thorough analysis and comparison of primary studies on computer vision approaches for plant species identification. We identified 120 peer-reviewed studies, selected through a multi-stage process, published in the last 10 years (2005–2015). After a careful analysis of these studies, we describe the applied methods categorized according to the studied plant organ, and the studied features, i.e., shape, texture, color, margin, and vein structure. Furthermore, we compare methods based on classification accuracy achieved on publicly available datasets. Our results are relevant to researches in ecology as well as computer vision for their ongoing research. The systematic and concise overview will also be helpful for beginners in those research fields, as they can use the comparable analyses of applied methods as a guide in this complex activity.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREBVATIONS

SHORT FORM	FULL FORM
KNN	K-Nearest Neighbor
SVM	Support Vector Machine
XGB	Extreme Gradient Boosting
SGD	Stochastic Gradient Descent

1 INTRODUCTION

All life on Earth depends on plants, which are also a vital resource for people's health. Botanists can use this application for therapeutic purposes, but plant recognition is crucial for managing plant species in agriculture. The properties of each plant's leaf can be utilized to categorize the plant. Plants are essential to the balance of nature and in people's lives. They arethe ultimate source of food and metabolic energy for nearly all animals, which cannot manufacture their own food. Thus, the study of plants is vital because they are a fundamental part of life on Earth, and generate the oxygen and food that allow humans and other organismsto exist. All life on Earth depends on plants, which are also a vital resource for people's health.Botanists can use this application for therapeutic purposes, but plant recognition is crucial for managing plant species in agriculture. In a manual identification process, botanist use different plant characteristics as identification keys, which are examined sequentially and adaptively to identify plant species.

By using a computer-aided plant identification system also non-professionals can take part in this process. Therefore, it is not surprising that large numbers of research studies are devoted to automate the plant species identification process. The primary goals of plant taxonomy are to specify characteristics of recently discovered species, to identify characteristics of undiscovered species by comparing them to known species In a manual identification process, botanist use different plant characteristics as identification keys, which are examined sequentially and adaptively to identify plant species. This study uses machine vision and digital image processing to propose a straightforward and computationally effective approach for finding plants.

2 PROBLEM OVERVIEW

Both the harmony of nature and the life of humans depend on plants. Nearly all animals, who are unable to produce their own food, rely on them as their primary source of nutrition and metabolic energy. In light of the fact that plants are an essential component of life on Earth and provide humans with food and oxygen, it is important to study them.as well as other living things. Botanists' work can be automated by using a digital plant identification system to quickly characterize plant species without their knowledge. Implementing the process of identification using different algorithms to get the best result and to identify the best process of identification The taxonomy of Ficus is highly complicated primarily due to its high number ofspecies. In Malesia (a Phyto geographical area that includes Indonesia, Malaysia, Philippines, Singapore, Brunei and Papua New Guinea), 367 species have been recorded where 27% (99 species) of the species is found in Peninsular Malaysia While the taxonomical account for Ficusfor the Milesian region is comprehensive, it is highly complicated and can only be used successfully to identify a species if sufficient plant samples are available. The fig is an essentialpart of the plant for accurate identification but may not be available all year round or branchesbearing figs may not be easily reachable for collection.

Taxonomic keys are also very technical and can only be used or interpreted by highly trained taxonomists or botanists. Such expertise is becoming rare and even if available their familiarity in identifying plants may be limited to a small group of plants they specialize.

A taxonomist needs to refer to herbarium materials to make comparative assessments with the specimen at hand in order to successfully identify a species. Herbarium materials are dried plant parts, mounted on a cardboard-like paper and systematically stored in an herbarium.

Herbaria are onlyfound in a few localities such as in established botanical garden or universities and are usuallyout of reach for non-botanical researches and thus not easily accessible to most people.

This paper explains our method for identifying plants from digital photographs of their leaves. Due to the seasonality of fruits and flowers and the disparity in root and stem characteristics, leaf-based traits are chosen above fruits, flowers, roots, and other aspects.
There are various freely accessible leaf picture databases, including the Image CLEF dataset, Leaf snap dataset, Intelligence dataset, and Flavia dataset. Using the Flavia dataset, the experiment's performanceis assessed.

3 LITERATURE REVIEW

.

Although a significant amount of research has been done studying various aspects of leaf identification in inventory systems, most of it deals with semi-automated systems. A state-of-the-art system which is fully automated and requires least human interaction is yet to be developed.

Pavan et al. proposed an algorithm for identification using multiclass classification based on color, shape volume and cell feature. They performed three stage comparisons: first stage compares redness, greenness, blueness index feature, second stage compare shape feature and the last stage compares cell feature and volume fraction feature. Experiment is performed on a sample of diverse collection of 1000 leaf and flower images. Limitations of this approach is that it semi-automatic approach and its recognition rate is up to 85% percent on an average.

Arun Priya the proposed approach consists of three phases such as pre-processing: transforming to grey scale and boundary enhancement, feature extraction: derives the common DMF from five fundamental features and classification: (SVM) classification for efficient leaf recognition. 12 leaf features which are extracted and orthogonalized into 5 principal variables are given as input vector to the SVM.

Valliamal et al. A probabilistic curve evolution method with particle filters is used to measure the similarity between shapes during matching process. The experimental results prove that the preferential image segmentation can be successfully applied in leaf recognition and segmentation from a plant image.

Dr. H.B.Kekre et al. the method of CBIR is discussed in this paper to filter images based on their content. In this paper feature vector is generated using color averaging technique, similarity measures and performance evaluation. Precision –Recall cross over plot is used as the performance evaluation measure to check the algorithm. The effect due to the size of database and number of different classes is seen on the number of relevancies of the retrievals.

Javed et al. used PNN to classify the plants with broad flat leaves. In this algorithm there were few select point where the user needs to specify the leaf blades and a base point according to which the image is then aligned and compared with other images based on some features like area, eccentricity, etc. They used 1200 sample leaves belonging to 30 different plants to train their system. This system is also semi-automatic and 91.41 percent accurate.

Arora. A et al. categorized the different images and used a variety of novel pre-processing methods such as shadow and background correction, petiole removal and automatic leaflet segmentation for identifying the leaf blobs. Also used complex network framework along with novel tooth detection method and morphological operations to compute several useful features. They used the PlantLeaves II dataset.

4 PROPOSED METHODOLOGY

The following method will be followed to achieve the objectives defined for proposedwork**:**

1. Collection of excellent quality images of leaves
2. Preparing the dataset of plant species
3. Installing the required the set of libraries
4. Giving the image of leaf for identifying the plant species
5. Applying machine learning model to predict the species

The method compares three features in three stages: the redness, greenness, and blueness index feature in the first stage, the shape feature in the second stage, and the cell feature and volume fraction feature in the third stage. The degree of similarity between shapes throughout the matching process is assessed using a probabilistic curve evolution method with particle filters. The experimental findings demonstrate that leaf recognition and segmentation from a plant image may be done successfully via preferential image segmentation. To identify the leaf blobs, we categorized the various photos and employed a range of unique pre-processing techniques, including petiole removal, shadow and background correction, and automatic leaflet segmentation.

Leaf selection

a) Sample images in this study consist of many plant structures, e.g., branches, stems, leaves, others. Where possible, only intact leaves were selected that had no apparenttearing and also free of damage from pest or disease.

b) Young leaves that are evidently small sized were ignored. Selected leaves were cropped out and saved as new images with a standard resolution (1800 × 1800 pixel). The stem was removed as it varies in length and would affect feature extraction. The leaves cropping and stem removing steps were done manually using ImageJ.

Image pre-processing

a) The leaf images have only one object, the leaf. Since all leaves are not perfectly flat, image capturing would always cast a shadow underneath the leaf. The shadow would disruptthe edge detection as it has a huge contrast with the background, confusing the algorithms to draw the boundary based on shadow instead of on the leaf.

b) Thus, it should be removed before image segmentation. Firstly, the image RGB value was changed to HSV value. Then, the channel with the clearest contrast between object and shadow was selected and used to find the object boundary. As HSV value conversion alters the original color, this step serves as a guidance for the subsequent edge detection ofRGB value leaf images, rather than producing a final image for feature extraction.

Feature extraction and description

a) Feature extraction refers to taking measurements, geometric or otherwise, of possibly segmented, meaningful regions in the image. Features are described by a set of numbers that characterize some property of the plant or the plant's organs captured in the images

b) Processed images from previous steps were transformed into a set of parameters that describe the leaf features. There are four classes of features extracted in this study: morphological features (shape), Hu moment invariants feature, texture features and histogram of oriented gradients. These features were selected specifically to obtain the important properties for image leaves, and to obtain the numeric values that can be used to distinguish between the different types of image leaves. Several feature methods, as described below were implemented to improve the accuracy of detection and matching criteria.

c) Shape features are a set of features commonly used in many studies. Shape values included in this study are: area, perimeter, eccentricity, and minor and major axes as shown in Figure

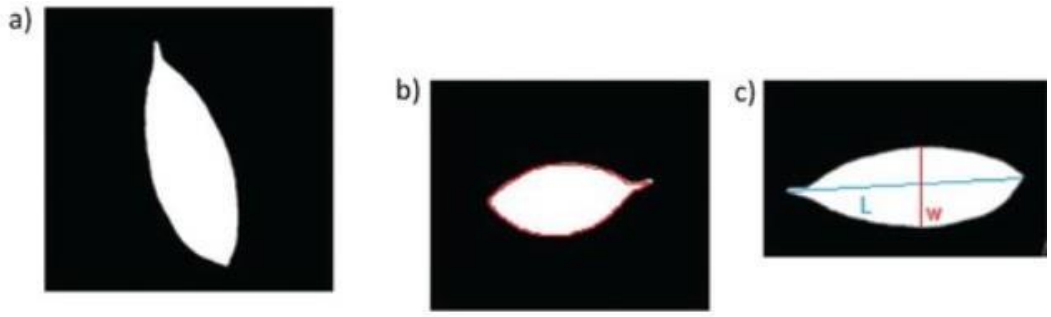

Fig. 1

Classification models training

a) In the classification step, all extracted features are concatenated into a feature vector, which is then being classified Two machine learning classification algorithms, ANN and SVM, were implemented. Data samples were divided into training and testing sets. Twenty images per species were used to build the training dataset, totaling to 60 images and the remaining samples were used for testing the dataset.

b) The ANN was developed using a multilayer perceptron network to train and classify the extracted features values into three classes representing the three species used in this study. The architecture of the ANN is a three-layer feed-forward network. More hidden layers could enhance the classification process; however, it would take more iterations of training the data, which will lead to a higher risk of overfitting.

c) Hence, considering the limited dataset used in this study, only one hidden layer was used. The architecture of ANN consists of 158 input nodes, 10 hidden nodes and three output nodes. The 158 input nodes refer to the number of features, while three output nodes refer to the 3 species we classified. The default setting of the number of hidden nodes in the MATLAB Neural Network Toolbox setting is 10, as this setting generally produces good results. Thus, the default setting of 10 hidden nodes was applied in this study. These neurons and the network were trained with the scaled conjugate gradient back-propagation algorithm

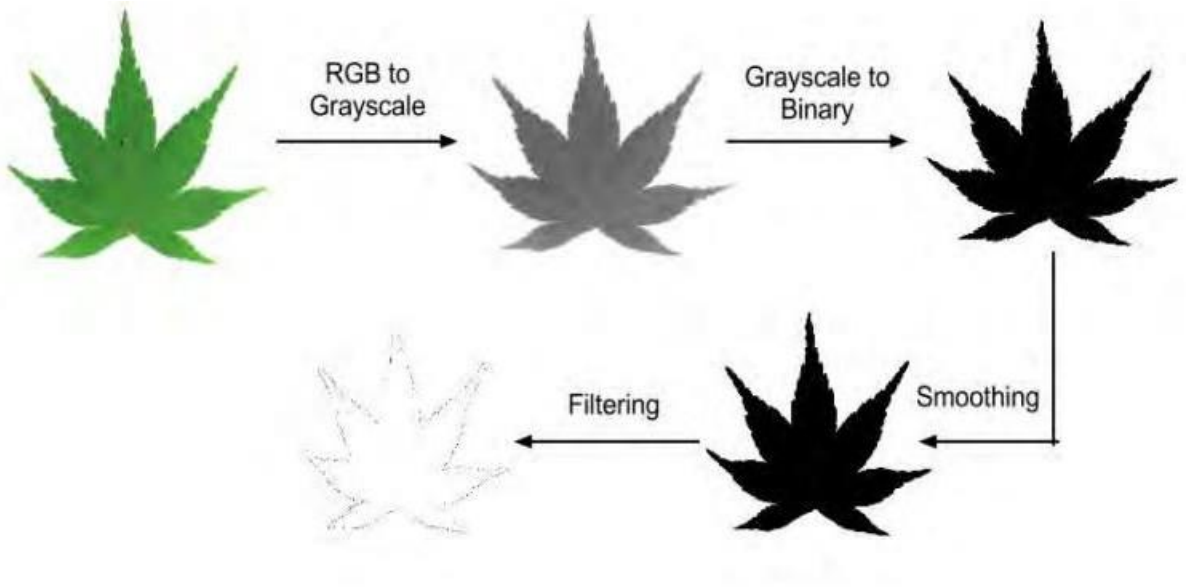

Fig. 2

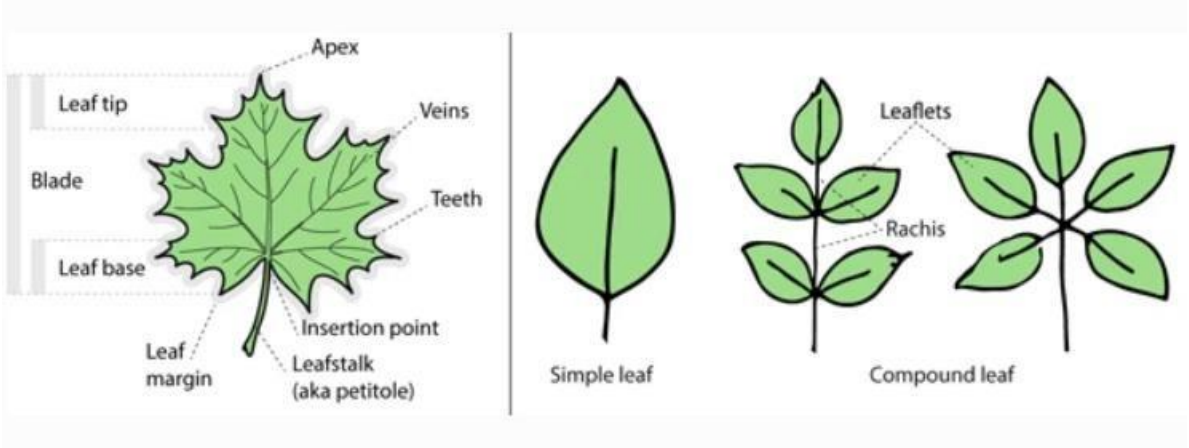

Fig. 3

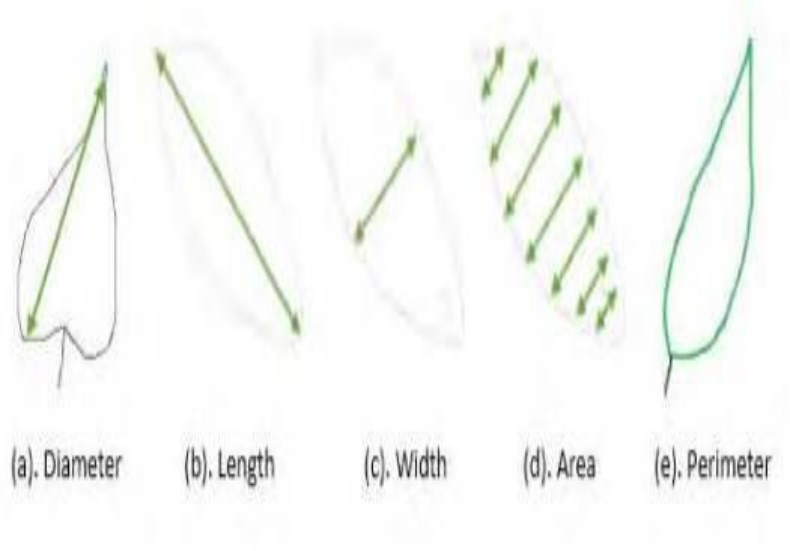

Fig. 4

4.1 ALGORITHMS

An algorithm is a step-by-step process to solve a problem or to do something. We designed our algorithm to complete our work with proper plan, within deadline. We followed the approach mentioned below

4.2 KNN Classifier

The k-nearest neighbors (KNN) algorithm is a simple, easy-to-implement supervised machine learning algorithm that can be used to solve both classification and regression problems. The KNN algorithm assumes that similar things exist in close proximity. In other words, similar things are near to each other. There are other ways of calculating distance, and one way might be preferable depending on the problem we are solving. However, the straight-line distance (also called the Euclidean distance) is a popular and familiar choice.

The KNN Algorithm

- Load the data
- Initialize K to your chosen number of neighbors
- For each example in the data
- Calculate the distance between the query example and the current example from the data.
- Add the distance and the index of the example to an ordered collection
- Sort the ordered collection of distances and indices from smallest to largest (in ascending order) by the distances
- Pick the first K entries from the sorted collection
- Get the labels of the selected K entries
- If regression, return the mean of the K labels

Advantages

- The algorithm is simple and easy to implement.
- There's no need to build a model, tune several parameters, or make additional assumptions.
- The algorithm is versatile. It can be used for classification, regression, and search (as we will see in the next section).

Disadvantages

- The algorithm gets significantly slower as the number of examples and/or predictors/independent variables increase.

4.3 SVM Classifier

Support Vector Machine" (SVM) is a supervised machine learning algorithm that can be used for both classification and regression challenges. However, it is mostly used in classification problems. In the SVM algorithm, we plot each data item as a point in n-dimensional space (where n is a number of features you have) with the value of each feature being the value of a particular coordinate. Then, we perform classification by finding the hyper-plane that differentiates the two classes very well

(Scenario-1): Here, we have three hyper-planes (A, B, and C). Now, identify the right hyper-plane to classify stars and circles.
SVM 2. You need to remember a thumb rule to identify the right hyper-plane: "Select the hyper-plane which segregates the two classes better". In this scenario, hyper-plane "B" has excellently performed this job.
(Scenario-2): Here, we have three hyper-planes (A, B, and C) andall are segregating the classes well.
SVM 3. Here, maximizing the distances between nearest data point (either class) and hyper-plane will help us to decide the right hyper-plane. This distance is called as Margin. Let's look at the below snapshot:
SVM 4. Above, you can see that the margin for hyper-plane C is high as compared to both A and B. Hence, we name the right hyper-plane as C. Another lightning reason for selecting the hyper- plane with higher margin is robustness. If we select a hyper-plane having low margin, then thereis high chance of miss-classification.
(Scenario-3).
SVM5.of you may have selected the hyper-plane B as it has higher margin compared to
A. But here is the catch, SVM selects the hyper-plane which classifies the classes accurately prior to maximizing margin. Here, hyper-plane B has a classification error and A has classified all correctly. Therefore, the right hyper-plane is A.

(Scenario-4) Below, I am unable to segregate the two classes usinga straight line, as one of the stars lies in the territory of other(circle) class as an outlier.
SVM 6. As I have already mentioned, one star at other end is like an outlier for star class. The SVM algorithm has a feature to ignore outliers and find the hyper-plane that has the maximum margin. Hence, we can say, SVM classification is robust to outliers.
SVM 7. Find the hyper-plane to segregate to classes (Scenario-5): In the scenario below, we can't have linear hyper-plane between the two classes, so how does SVM classify these two classes? Till now, we have only looked at the linear hyper-plane.

SVM 8, SVM can solve this problem. Easily! It solves this problem by introducing additional feature. Here, we will add a new feature z=x^2+y^2. Now, let's plot the data points on axis x and z.

The SVM kernel is a function that takes low dimensional input space and transforms it to a higher dimensional space i.e., it converts not separable problem to separable problem.

It is mostly useful in non-linear separation problem. Simply put, it does some extremely complex data transformations, then finds out the process to separate the data based on the labels or outputs you've defined.

Pros

- It works well with a clear margin of separation
- It is effective in high dimensional spaces.
- It is effective in cases where the number of dimensions is greater than the number of samples.
- It uses a subset of training points in the decision function (called support vectors), so it is also memory efficient.

Cons

- It doesn't perform well when we have large data set because the required training time is higher
- It also doesn't perform very well, when the data set has more noise i.e., target classes are overlapping
- SVM doesn't directly provide probability estimates, these are calculated using an expensive five-fold cross-validation. It is included in the related SVC method of Python scikit-learn library.

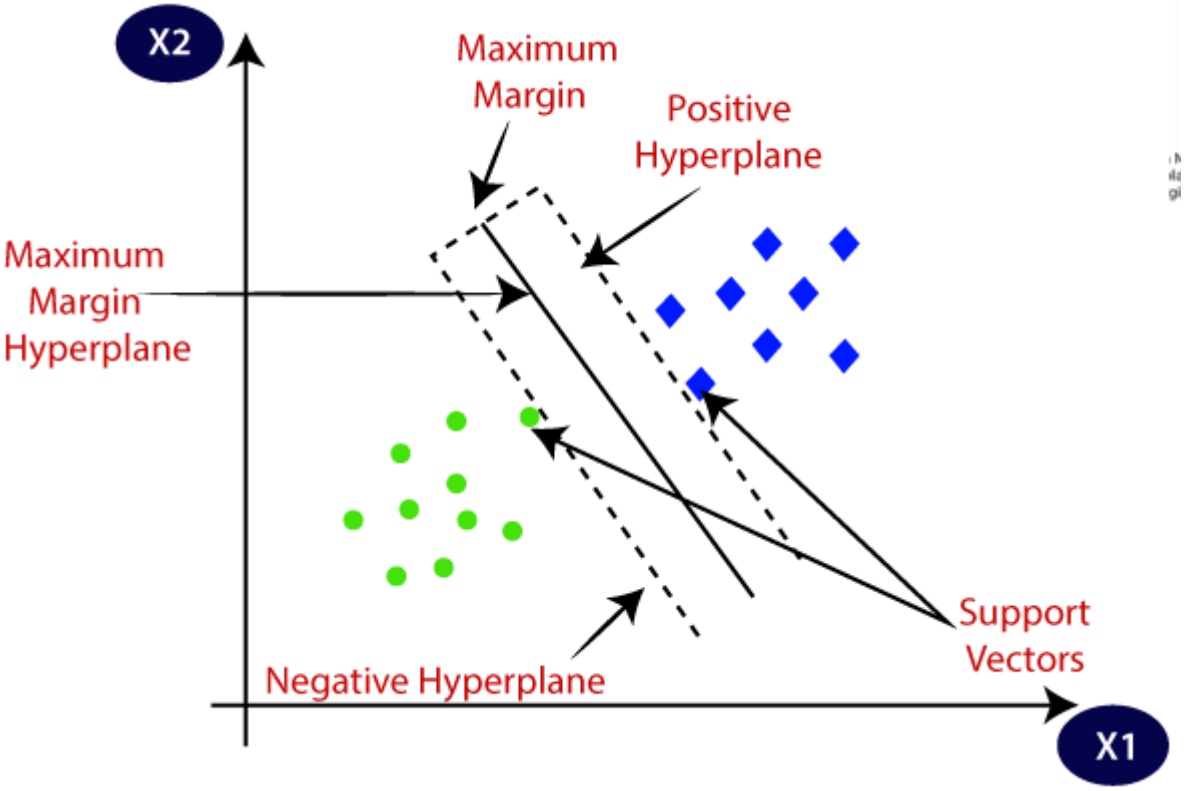

Fig. 5

4.4 XGB Classifier

- XGBoost is an implementation of Gradient Boosted decision trees. XGBoost models majorly dominate in many Kaggle Competitions.
- In this algorithm, decision trees are created in sequential form. Weights play an important role in XGBoost. Weights are assigned to all the independent variables which are then fed into the decision tree which predicts results.

- The weight of variables predicted wrong by the tree is increased and these variables are then fed to the second decision tree. These individual classifiers/predictors then ensemble to give a strong and more precise model. It can work on regression, classification, ranking, and user-defined prediction problems.

4.5 Data Description

The dataset consists approximately 1,584 images of leaf specimens (16 samples each of 99 species) which have been converted to binary black leaves against white backgrounds. Three sets of features are also provided per image: a shape contiguous descriptor, an interior texture histogram, and a fine-scale margin histogram. For each feature, a 64-attribute vector is given per leaf sample.

Note that of the original 100 species, we have eliminated one on account of incomplete associated data in the original dataset.

File descriptions

train.csv – the training

settest.csv – the test set

sample_submission.csv – a sample submission file in the correct format

images – the image files (each image is named with its corresponding id) ## Data

fieldsid – an anonymous id unique to an image

margin_1, margin_2, margin_3, …, margin_64 – each of the 64 attribute vectors for the marginfeature

shape_1, shape_2, shape_3, …, shape_64 – each of the 64 attribute vectors for the shape feature

texture_1, texture_2, texture_3, …, texture_64 – each of the 64 attribute vectors for the texturefeature.

4.6 Random Forest classifier

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of ensemble learning, which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model. As the name suggests, "Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset." Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output.

The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting. Below are some points that explain why we should use the Random Forest algorithm: It takes less training time as compared to other algorithms. It predicts output with high accuracy, even for the large dataset it runs efficiently. It can also maintain accuracy when a large proportion of data is missing.

Advantages of Random Forest

- Random Forest is capable of performing both Classification and Regression tasks.
- It is capable of handling large datasets with high dimensionality.
- It enhances the accuracy of the model and prevents the overfitting issue.

Disadvantages of Random Forest

- Although random forest can be used for both classification and regression tasks, it is not more suitable for Regression tasks.

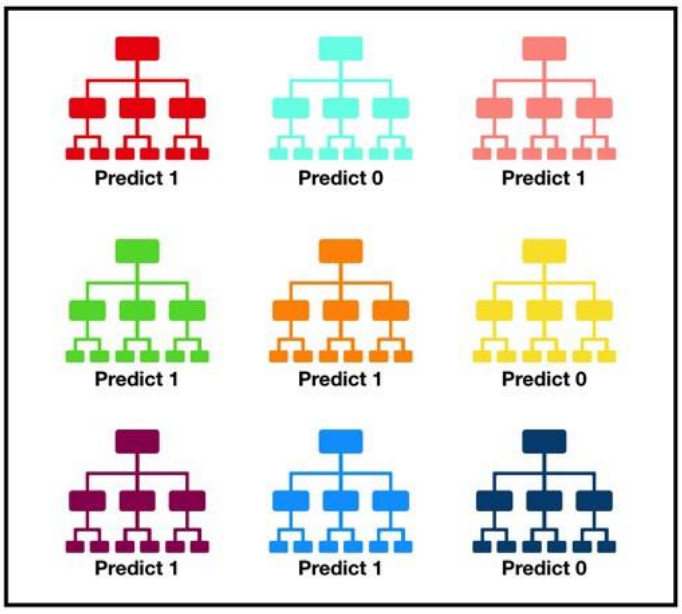

Fig. 6

4.7 SGD Classifier

SGD Classifier is a linear classifier (SVM, logistic regression, .) optimized by the SGD. These are two different concepts. While SGD is an optimization method, Logistic Regression orlinear Support Vector Machine is a machine learning algorithm/model. You can think of that amachine learning model defines a loss function, and the optimization method minimizes/maximizes it.

Stochastic gradient descent (SGD) computes the gradient using a single sample. Mini-batch gradient descent finally takes the best of both worlds and performs an update for every mini-batch of n training examples.

As we can read from the earlier text, SGD allows minibatch (online/out-of-core) learning. Therefore, it makes sense to use SGD for large scale problems where it's very efficient.

The minimum of the cost function of Logistic Regression cannot be calculated directly, so we try to minimize it via Stochastic Gradient Descent, also known as Online Gradient Descent. In this process we descend along the cost function towards its minimum (please have a look at the diagram above) for each training observation we encounter.

Another reason to use SGD Classifier is that SVM or logistic regression will not work if you cannot keep the record in RAM. However, SGD Classifier continues to work.

5 CODE

In [6]:
```
len(os.listdir('images'))
```

Out[6]:
```
1584
```

In [7]:
```
import matplotlib.pyplot as plt
plt.figure(figsize=(20,15))
import cv2 as cv
from keras.preprocessing.image import load_img
for i in range(25):
    j=np.random.choice((os.listdir('images')))
    plt.subplot(5,5,i+1)
    img=load_img(os.path.join('/kaggle/working/images',j))
    plt.imshow(img)
```

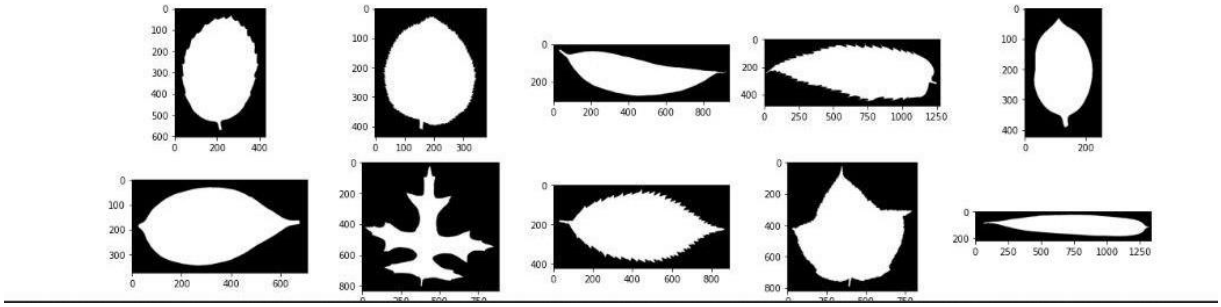

Fig. 7

In [14]:
```
df=df.drop(['id','species'],axis=1)
test_id=dftest.id
dftest=dftest.drop(['id'],axis=1)
```

In [15]:
```
dftest
```

Out[15]:

	margin1	margin2	margin3	margin4	margin5	margin6	margin7	margin8	margin9	margin10	...
0	0.019531	0.009766	0.078125	0.011719	0.003906	0.015625	0.005859	0.000000	0.005859	0.023438	...
1	0.007812	0.005859	0.064453	0.009766	0.003906	0.013672	0.007812	0.000000	0.033203	0.023438	...
2	0.000000	0.000000	0.001953	0.021484	0.041016	0.000000	0.023438	0.000000	0.011719	0.005859	...
3	0.000000	0.000000	0.009766	0.011719	0.017578	0.000000	0.003906	0.000000	0.003906	0.001953	...
4	0.001953	0.000000	0.015625	0.009766	0.039062	0.000000	0.009766	0.000000	0.005859	0.000000	...
...	...	...	...	...	...	...	...	...	...	...	...
589	0.000000	0.000000	0.003906	0.015625	0.041016	0.000000	0.017578	0.000000	0.005859	0.013672	...
590	0.000000	0.003906	0.003906	0.005859	0.017578	0.000000	0.017578	0.005859	0.000000	0.005859	...
591	0.017578	0.029297	0.015625	0.013672	0.003906	0.015625	0.025391	0.000000	0.000000	0.009766	...
592	0.013672	0.009766	0.060547	0.025391	0.035156	0.025391	0.039062	0.000000	0.003906	0.023438	...
593	0.000000	0.117190	0.000000	0.019531	0.000000	0.136720	0.001953	0.005859	0.000000	0.007812	...

594 rows × 192 columns

In [16]:
```
df.info()
```

Fig. 8

In [8]:
```
df=pd.read_csv('train.csv',index_col=False)
dftest=pd.read_csv('test.csv',index_col=False)
```

In [9]:
```
df.head()
```

Out[9]:

	id	species	margin1	margin2	margin3	margin4	margin5	margin6	margin7	margin8	...
0	1	Acer_Opalus	0.007812	0.023438	0.023438	0.003906	0.011719	0.009766	0.027344	0.0	...
1	2	Pterocarya_Stenoptera	0.005859	0.000000	0.031250	0.015625	0.025391	0.001953	0.019531	0.0	...
2	3	Quercus_Hartwissiana	0.005859	0.009766	0.019531	0.007812	0.003906	0.005859	0.068359	0.0	...
3	5	Tilia_Tomentosa	0.000000	0.003906	0.023438	0.005859	0.021484	0.019531	0.023438	0.0	...
4	6	Quercus_Variabilis	0.005859	0.003906	0.048828	0.009766	0.013672	0.015625	0.005859	0.0	...

5 rows × 194 columns

In [10]:
```
dftest
```

Out[10]:

	id	margin1	margin2	margin3	margin4	margin5	margin6	margin7	margin8	margin9	...	te

Fig. 9

In [27]:
```
test_pred = model_5.predict(dftest)
```

In [28]:
```
print(test_pred)
```

```
[51 50  1 19 14  3  3 28 84  8 43 74 75 10 52 46 45 73 13 71 61 68 57 77
  1 70 28 15 35 70 53 74 47 50  4 36 14 55 36 93  8 32  8  9 71 70 38 23
 94 18 17  5 55 94 14 86 62 33 51 98 88 56 21 59 65 11 48  5 13  4 54 57
 29  7 31 98 92 84 25 10 61 43 85 24  1  2 23 83 40 22 48 90 25 21 37 56
 41 95  7 89 98 77  3 12 31 84 53 96 64 72 93 93 67 30  8 88 60 87  6 57
 34 90 60 17 75 27 51 31 39 23 68  2 41 61 24 97 29 28 68 81 42 51 86 62
 60 52 95 64 42 96 95 20 59 35 86  1 26 38 43 75 20 60 46 79 22 79 69 87
 65 97 75 21 29 21 11 10 58 94 27 22 15 45 89 54 43  5 23 94 40 49 89 72
 36 11 81 95 18 91 29 64 80  6 78 45 28  9 78 90 44 89 92 13  2 59  0 96
 70 32 29 78 91 55 44 38  5 60 49 58 94 67 92 88 90 79 25 37 18  0 76 27
 70 71 44 70 32 90 30 82 34 30 82 96 48 65 57 64 26 53 55 73  9  3 83 26
 30 63 17 22 79 63 12 78 36 14 27 25 67 38 20 54 76 69 67 97 80 44 92 69
 23 21 16 51 33 77 16 11 97  1 52 39 24 52 42 17  2 73 96 83 88  9 63 50
 16 38 87 95  3 35 83 60 59 58  0 50 62 38 93 68 55 46 19 46 94 18  0 33
 89 40 62 48 42  6 31 91 73 81 12 85 47  6 45  2 22 35 43  6 80 78 82  5
 61 37 43 33 69 56 71 45 59 42 66 86 98 83 90 64 82 11 79 56 76 49 48 20
 74 15 33 49 89 44  7 35 14 55 23 34 44 32 30 36  9 72 31 61 50 82 34 29
 22 92 72 11 19  4 87 51 71 39 84 32 66 36 41 31 80  4 26 68 96 20 40 34
 39 56 73 76 84  7 67 37  8 95 85 62 10 65 41  2 17 86 41 52  3 49 47 76
 52 16 26 88 63 45 39 66 87 75 74  7 64 65 78 63 56 21 61 88 62 91 59 12
 74 15 85  8 66 57 83 82 72 58 28 17 67 66 57 66 75 35 18  9 54 91 65 19
 15 10 24 71 69 48 39 98 16 19 45 74  6 69 42 34 80 47 85 28 85 47 25 27
```

Fig. 10

In [26]:

```
models = pd.DataFrame({
    'Model': ['SGDClassifier','Random Forest Classifier','K Neighbors Classifier', 'XGB Classifier',
              'svm'],

    'Score': [model_1.score(X_test,y_test)*100,
              model_2.score(X_test,y_test)*100,
              model_3.score(X_test,y_test)*100,
              model_4.score(X_test,y_test)*100,
              model_5.score(X_test,y_test)*100]})
models.sort_values(by='Score', ascending=True)
```

Out[26]:

	Model	Score
3	XGB Classifier	83.333333
0	SGDClassifier	95.959596
2	K Neighbors Classifier	95.959596
1	Random Forest Classifier	97.474747
4	svm	98.484848

In [27]:

```
test_pred = model_5.predict(dftest)
```

In [28]:

```
print(test_pred)
```

Fig. 11

6 OUTPUT

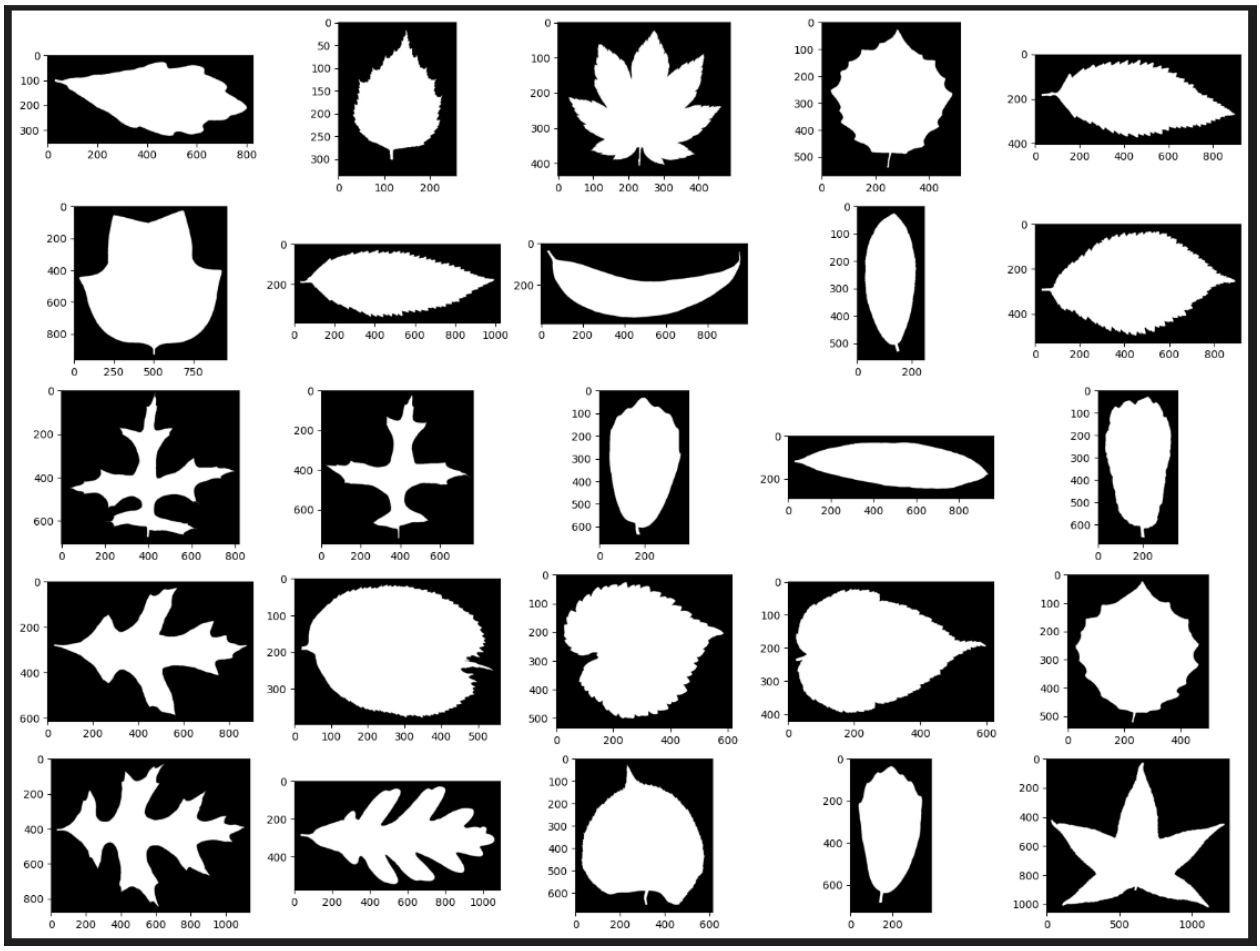

Fig. 12

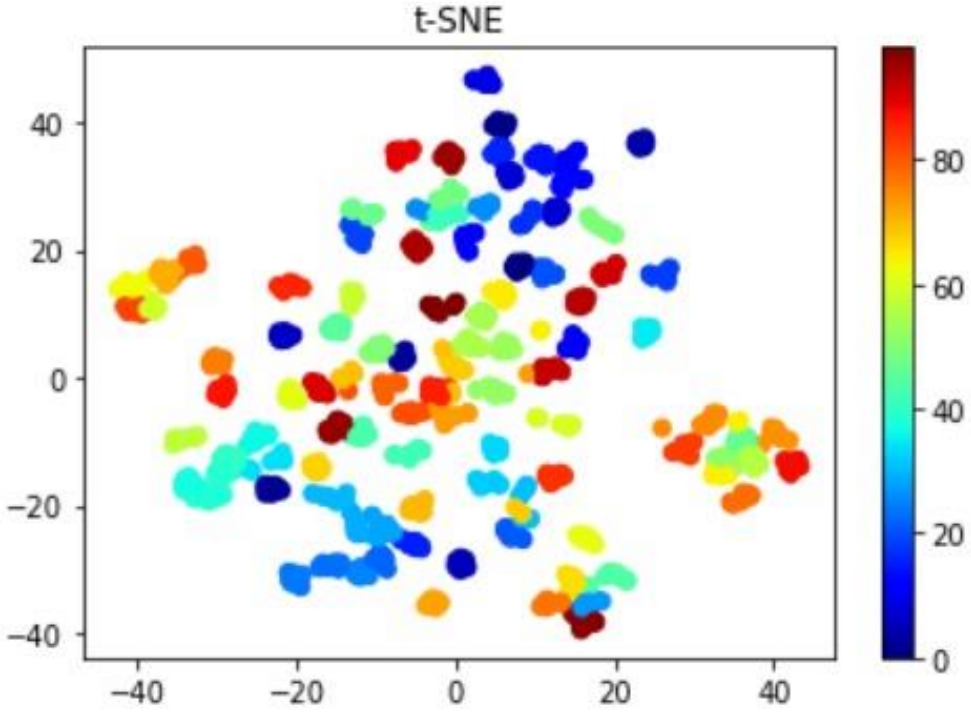

Fig. 13

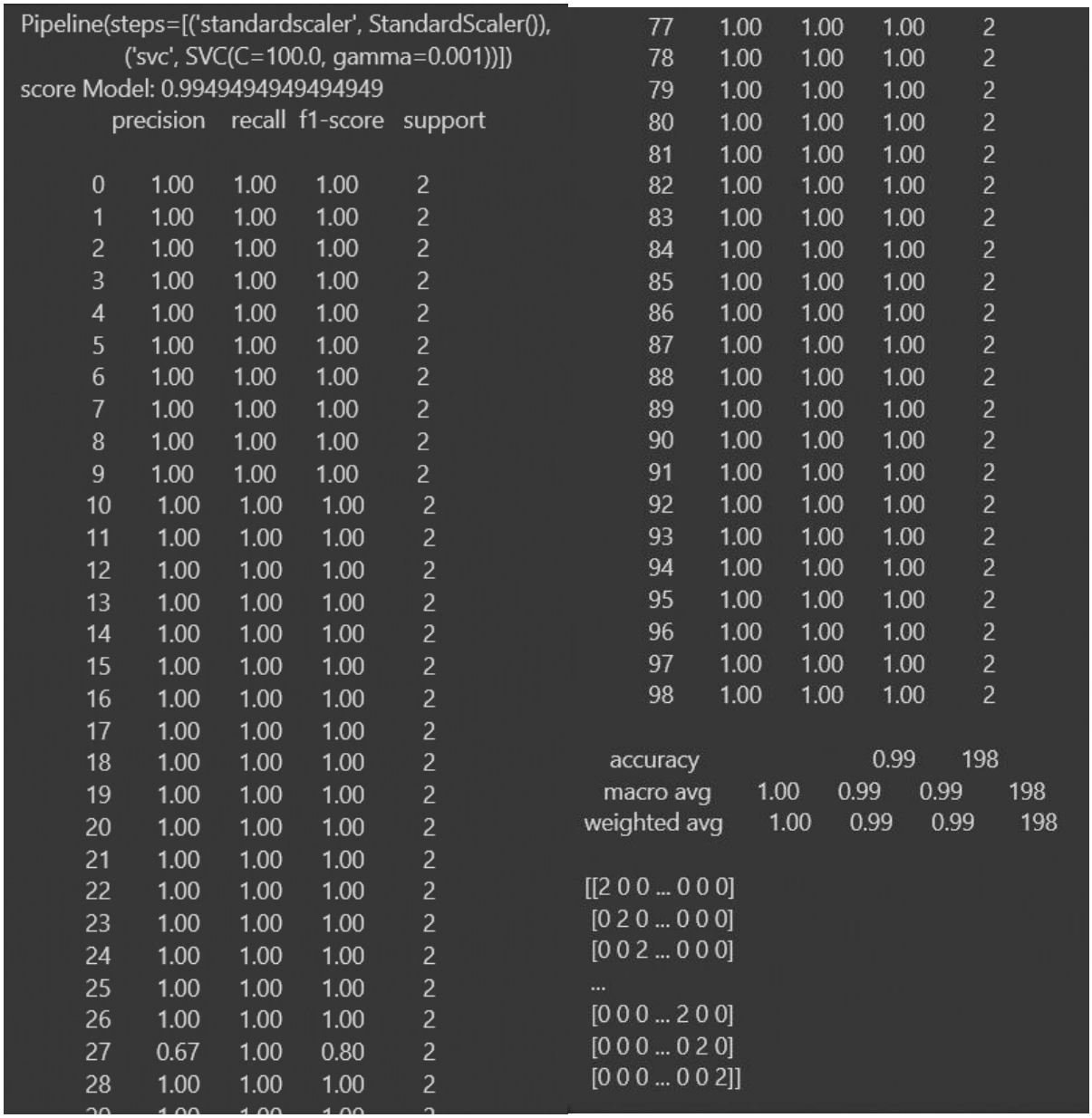

Fig. 14

ALGOROTHIM USED	ACCURACY
K-NEIGHBOR CLASSIFER	95.59
Random Forest Classifier	97.47
SVM	98.98

6 CHALLENGES AND OPPORTUNITIES

Building a Global Dataset: Global biodiversity informatics initiatives such as GBIFF, EOL, and BHL, have successfully built large global databases of biodiversity information that is freely available on the web. GBIF currently provides more than 600 million specimen-level records, EOL over a million species level descriptions, and BHL more than 50 million pages of literature. An analogous dataset of digital images of plant elements (e.g., leaves) does not exist. However, there are several opportunities that should be taken. First of all, digital cameras are now very inexpensive and powerful. Secondly, even though data sharing protocols and standards need to be in place, organizations such as TDWG are devoted to precisely this endeavor. Finally, crowd sourcing offers now excellent opportunities to both, generate large repositories of information, and raise awareness of the general public through citizen science projects. I Naturalist and planet have been very successful and deserve being emulated. The Plant CLEF dataset already demonstrates that this can be done at the European level.

Deep Learning: Deep Learning, particularly using CNN, is a very hot topic in computer vision. The exciting results obtained in events such as ImageNet [9] have generated a lot of expectation. As more data and computational power are now available, this technique has become the most widely used, without substantial algorithmic changes since its inception. Instead of following a gradual path that aims at using images of elements of an organism first (e.g., leaves or flowers of a plant), and then pictures of the whole organism, CNN tackles directly the challenge of identifying organisms by using pictures of the whole or parts of the organism. However, this approach has at least two important limitations. First, it tends to work better with very large sets of images [11]. Secondly, it lacks the explanatory power of other approaches such as landmark-based morphometrics. Nevertheless, as global data sets are developed, it is just a matter of time to overcome the former. Additionally, research work is already under way to overcome the latter [10].

7 CONCLUSION AND FUTURE SCOPE

7.1 CONCLUSION

In this study, we have presented baseline automated identification techniques of plant species based on herbarium leaf images, using the pattern recognition approach. Five machine learning algorithms: Random Forest, SGD, KNN, XGB, and SVM have been used to build identification models. Both models achieved satisfactory results demonstrating their usefulness in identification tasks. The study presented here showed that automated classification of plant species that had similar leaf shapes is feasible based on leaf images. Although the SVM classifier has given us best results Though the developed system is not intended to replace human taxonomists, it may provide a rapid and easily accessible technique to identify plants with acceptable accuracy. We chose to work on species of plants as it is a large genus and species identification can be difficult especially to non-taxonomists

7.2 FUTURE SCOPE

1. we can focus on different techniques to find the plant species using larger datasets. In this research, we use different classifiers to identify the plant species.
2. we can use various parts of plants to identify the species of the plant.
3. We can make a real time image identification
4. We can also make a graphical user interface to make process more user friendly

8.REFERENCES

1. Arora A., Gupta A., Bagmar N., Mishra S., Bhattacharya A.: A Plant Identification System using Shape and Morphological Features on Segmented Leaflets: Team IITK, CLEF 2012 In: CLEF (Online Working Notes/ Labs/Workshop). (2012)
2. Neeraj Kumar, Peter N. Belhumeur, Arijit Biswas, David W. Jacobs, W. John Kress, Ida C. Lopez, and João V. B. Soares, "Leafsnap: A Computer Vision System for Automatic Plant Species Identification", Computer Vision – ECCV 2012, 2012, pp 502-516.
3. Pavan Kumar Mishra, Sanjay Kumar Maurya, Ravindra Kumar Singh, Arun Kumar Misra, "A semi-automatic plant identification based on digital leaf and flower Images", International Conference On Advances In Engineering, Science And Management (ICAESM), 2012, pp. 68-73.
4. ArunPriya C., Balasaravanan T., Antony Selvadoss Thanamani, "An Efficient Leaf Recognition Algorithm for Plant Classification Using Support Vector Machine", Proceedings of the International Conference on Pattern Recognition, Informatics and Medical Engineering, 2012, pp. 428-432.
5. N.Valliammal, Dr. S.N.Geethalakshmi, "Automatic Recognition System Using Preferential Image Segmentation for Leaf and Flower Images", Computer Science & Engineering: An International Journal (CSEIJ), 2011, pp. 13-25.
6. Dr. H.B.Kekre, Mr. Dhirendra Mishra, Ms. Stuti Narula, Ms. Vidhi Shah, "Color Feature Extraction for CBIR", International Journal of Engineering Science and Technology (IJEST), 2011, pp. 8357-8365.
7. Javed Hossain, M. Ashraful Amin, "Leaf Shape Identification Based Plant Biometrics", Proceedings of 13th International Conference on Computer and Information Technology (ICCIT), 2010, pp. 458-463.
8. Joly, A., et al.: LifeCLEF 2015: multimedia life species identification challenges. In: Mothe, J., Savoy, J., Kamps, J., Pinel-Sauvagnat, K., Jones, G., San Juan, E., Capellato, L., Ferro, N. (eds.) CLEF 2015. LNCS, vol. 9283, pp. 462–483. Springer, Heidelberg (2015). doi:10.1007/978-3-319-24027-5_46
9. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Bartlett, P., Pereira, F., Burges, C., Bottou, L., Weinberger, K. (eds.) Advances in Neural Information Processing Systems 25, pp. 1106–1114 (2012)
10. Lee, S.H., Chan, C.S., Wilkin, P., Remagnino, P.: Deep-plant: plant identification with convolutional neural networks (2015). CoRR, vol. abs/1506.08425
11. Simard, P.Y., Steinkraus, D., Platt, J.C.: Best practices for convolutional neural networks applied to visual document analysis. In: Proceedings of the Seventh International Conference on Document Analysis and Recognition, ICDAR 2003, vol. 2, p. 958. IEEE Computer Society, Washington, DC (2003)