Multimedia Tools and Applications (2024) 83:40323–40336
https://doi.org/10.1007/s11042-023-17311-8

A novel transfer learning-based approach for plant species prediction using leaf images

Silky Sachar[1] · **Anuj Kumar**[1]

Received: 7 September 2021 / Revised: 10 June 2022 / Accepted: 28 September 2023 /
Published online: 13 October 2023

Abstract
There is extensive research going on in the field of automatic identification of plants in order to preserve the plant species which are on the verge of extinction and also to educate people or new generation farmers about the various plants growing in their vicinity. Few plants also possess certain medicinal properties which can be used by the layman to treat commonly occurring ailments. We suggest a novel leaf identification approach using combination of deep learning and conventional machine learning techniques. In this approach, the leaf image features are extracted using a neural network pre-trained on the ImageNet and then fed into the machine learning classifiers for predictions. We prepared three different models and analyzed their performance. Thereafter, we propose an ensemble approach based on stacking classifiers where the predictions of multiple classifiers were used to train a meta-classifier. This approach achieved an accuracy of 99.16% and 98.13% on the unseen samples of Swedish and Flavia datasets respectively.

Keywords Convolutional Neural Networks · XGBoost · Machine Learning · Leaf identification · Stacking classifier · Ensemble learning

1 Introduction

Human beings rely heavily on plants for their survival. The benefits of plants are well known but a lot of work is being done to automatically identify them. The automatic identification reduces the chance of human error and helps to educate people about the plants growing in their surroundings, which they can use for their benefit. An excellent review of various plant identification techniques has been discussed in [1] and another recent review in this field have been presented in [2]. Most studies rely on leaf images for the automatic identification of the plant. The images could be captured by a smartphone camera or scanned. These images were then input into the machine learning model to extract features. The features in case of leaf images refer to shape, color, texture and venation. After the

✉ Silky Sachar
silky.sachar@gmail.com

[1] Department of Computer Science and Application, Panjab University, Chandigarh, India

features have been extracted in the form of a feature vector, it is input into the classifier. The classifier classifies the feature vector into its relevant class.

The recent trend in the field of artificial intelligence has been shifted to deep learning where Convolutional neural Networks (CNN) are used. The CNN consists of layers which extract the image details/features by assigning learnable weights/bias to the various objects to differentiate one class from another. Convolutional neural networks can be developed from scratch [3] to classify the tree images into their respective species or features obtained from multiple CNNs could be concatenated to form 1 feature vector to be classified by conventional machine learning classifier [4]. In another contribution, a CNN based on ResNet Inception V2 was proposed along with a texture based leaf image descriptor to classify leaves into their respective classes [5]. The CNNs work very well on large datasets but face few issues like overfitting on small datasets if not properly dealt with. To classify the leaf images into their respective classes and to achieve a performance which is at par with the CNNs, we conducted this study to propose a novel approach of transfer learning using the VGG16 CNN architecture pretrained on the ImageNet. The features extracted from the leaf images using the CNN architecture were classified by using XGBoost, NuSVC and the Ensemble based on stacking classifiers consisting of the K-NN, Decision Tree, Naïve Bayes as the base learners and the Logistic Regression classifier as the meta-learner. The models were evaluated using 5-Fold Cross validation and also using the hold-out validation technique on two publicly available datasets namely Swedish and Flavia. Performance measures used were accuracy, precision and recall. The Ensemble achieved promising results. Section 2 provides the related work in this field. Section 3 provides proposed model; Section 4 specifies the experiments and results. Section 5 presents Conclusion and future work.

2 Related work

It has been observed in the literature that different conventional machine learning techniques utilized by various authors in the field of automatic identification of plants by using leaf images, have given some exemplary results. In the work proposed in [6], the authors have identified leaves based on their shape and have proposed a shape descriptor called Integral Contour Angle(ICA). For each boundary point on the leaf image, two vectors project to left and right boundary sides forming an integral contour angle. These angles can also be obtained at different scales to form multiscale ICA. The multiscale ICA's form a set which can be used to distinguish between two leaf shapes by calculating the Hausdorff distance between them. They were able to obtain 98.4% accuracy on Swedish dataset and 89.03% on CVIP100 leaf dataset [7] . A modified texture extraction approach was developed in [8] where the authors used the red and green color channels of the input image as opposed grey tones while calculating Local Binary Patterns(LBP). When the feature vector was input into the Extreme Learning machine classifier, it was observed to provide 98.94 % accuracy in Flavia, 99.4 % in Swedish, 83.7% in ICL and 92.92 % in Foliage datasets. For the automatic identification of lawn plants, authors in [9] used shape, color and texture features of the plants. Shape was obtained using Fourier Harmonic functions, texture features obtained using Gabor filter were local gradient, second moment matrix and energy. For the extraction of color, Hue, Saturation, Value (HSV) color space was used. The features were then input into the least square-Support Vector Machines (SVM) for classification and segmentation.

The conventional machine learning approaches have proven to be efficient but the latest trend in image recognition is shifting to deep learning as it eliminates the need of human intervention. Authors in [10] have proposed the Mask RCNN for automatic segmentation of *Arabidopsis* plants. This model consists of feature extractor whose output is sent to Region Proposal Network (RPN). The output of RPN is then fed into the box classifier, box regressor and object mask. The model obtained highest accuracy of 90% on combination of synthetic and real data which outperformed various state-of-the-art techniques. Another work based on deep learning was proposed in [11] where authors have developed a 50 layer deep network to identify plants from the LeafSnap dataset. Transfer learning was used in [12] to extract the features of input leaf images using MobileNet CNN architecture and then classified using Logistic regression classifier. The authors obtained an accuracy of 99.6% on LeafSnap dataset and 90.54% on Flavia dataset. In [13], the authors proposed D-Leaf CNN architecture and compared it's feature extraction capabilities to pre-trained CNN model namely AlexNet and fine-tuned AlexNet. For the purpose of classification, authors used conventional machine learning classifiers (Support Vector Machines (SVM), k-Nearest-Neighbor (k-NN), Naïve Bayes (NB) and deep learning classifiers (Artificial Neural network (ANN) and CNN). The performance of these were compared to the standard Sobel segmented veins. The D-Leaf achieved an accuracy of 94.88% which was at par with the AlexNet and fine-tuned AlexNet. The features extracted from the CNN model also worked better with ANN classifier.

3 The proposed model

The introduced model consists of mainly two modules as shown in Fig. 1. First module is the feature extractor module which is based on the deep learning CNN architecture VGG-16 by using transfer learning. The output from the first module constitutes the feature vector which acts as input to the second module. The feature vector forms the basis on which one species will be differentiated from the other species. The second module is called the classification module which is based in the traditional machine learning classifiers like XGBoost, NuSVC, and the ensemble. The ensemble learning technique adopted in

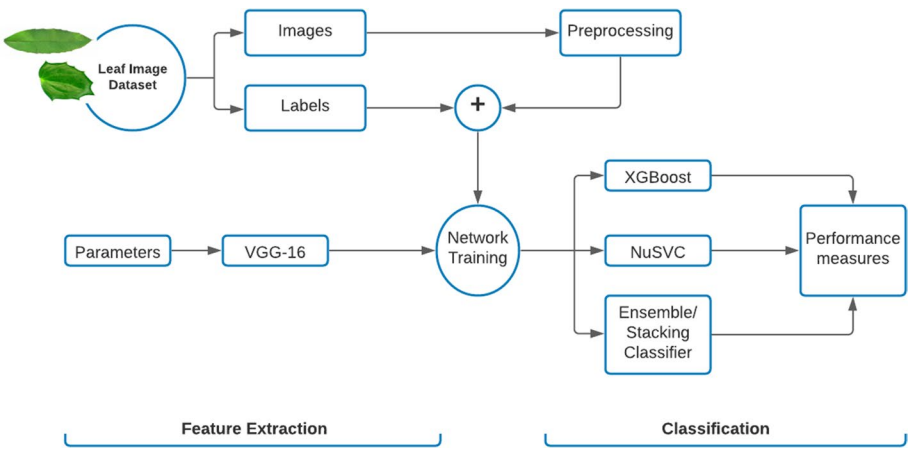

Fig. 1 Transfer learning methodology used in the experiments

this work is the stacking classifiers approach which is explained in detail in the following subsections.

3.1 Datasets

In this work, we have used two freely available leaf image datasets namely Swedish [14], Flavia [15] . Publicly available datasets allow other researchers to develop their algorithms and test them too. Swedish dataset consists of leaf images from 15 classes of trees. Flavia consists of 32 classes of cleaned leaf images taken against white background. There are about 50 images in each class of the above-mentioned datasets. Few images from each dataset are shown in Fig. 2

3.2 Extraction of features using deep learning frameworks

Deep learning has proven to provide exceptional results in all fields of machine learning including compute r vision, pattern recognition and image identification. The frameworks based on deep learning use layered architectures which include deep neural and deep belief networks, recurrent neural networks and CNN. CNN have been used in the literature to classify leaf images into their respective classes and also to extract features which can be fed into other conventional classifiers. Deep learning feature extraction eliminates the need of hand-crafted feature extractors leaving less scope of human errors.

It has also been observed in the literature that the outcome of machine learning models can be improved when input features are extensively engineered.

CNN architecture is comprised of various layers which take part in feature extraction and classification. The feature extraction part is achieved using convolution, ReLu and Pooling layers. ConvNets have the inherent ability to learn characters/features with training. Since,

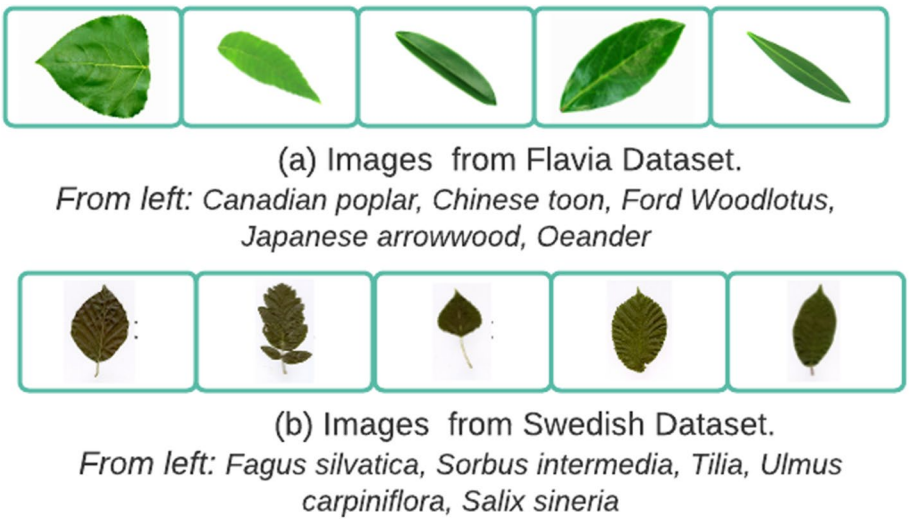

Fig. 2 **a** Images from Flavia Dataset. From left: *Canadian poplar, Chinese toon, Ford Woodlotus, Japanese arrowwood, Oeander.* **b** images from Swedish Dataset. From left: *Fagus silvatica, Sorbus intermedia, Tilia, Ulmus carpiniflora, Salix sineria*

the images used in this experiment were colored, the image information was also distributed across RGB channels. CNNs reduce the images to form which is less computationally extensive, while preserving the inherent information. Initial Conv layers extract low-level information like edges, color and as it moves to last layers, it begins to capture high level information which helps distinguish one image from another. Pooling layers help to extract the governing features from the outputs of conv layers by reducing dimensions. The two important variants of pooling layer are Max pooling and average pooling. The max pooling is considered superior as it suppresses noise. Another important layer is the Rectified Linear Units (ReLu) layer which turn all the negative activations to zero and introduces non-linearity to the network. It is also considered to its counterparts (tanh and sigmoid) as it trains faster without loss in accuracy. CNN also have softmax layer which acts as a classifier as part of the fully connected layer. In the experiments, we have removed the classification part of the CNN architectures. Various powerful and efficient CNN models have been developed and made freely available for deep learning community.

3.2.1 Transfer learning using VGG-16

The term stands for Visual Geometric Group from the Oxford University. This model won the 1st and 2nd place in the object detection and classification in 2014 ImageNet Large Scale Visual Recognition Challenge (ILSVRC). It was developed to classify objects into 1000 classes. This network is composed of 16 layers, divided into 13 convolution, 5 max pooling layers and 3 fully connected layers. The 13+3 layers consist of trainable parameters. The architecture map is shown in Fig. 3.

Transfer learning was employed to extract features of the input images using this pre-trained CNN model. The colored images were first read from the datasets and resized to 224x224 to maintain a standard size for the model. The model was used without the classification layers as the network layers are used to extract features of the input images. To control overfitting, EarlyStopping method of Keras was used to monitor the loss to halt the training. The feature vector thus obtained was further input into the classifier for classification into their respective classes. The main contribution of this study was to construct XGBoost, NuSVC and a stacking based ensemble algorithm that aims to solve the leaf classification problem without the need of computationally intensive hardware systems or the use of very deep CNNs to overcome the memory constraint. Stacking based algorithm effectively combines the outputs from simple machine learning classifiers to provide its own species prediction.

3.2.2 Min-max scaling for data normalization

Min-Max scaling is a popular normalization technique. This method makes sure that all the features are transformed into the range [0 ,1] or in the range [-1,1] if there are negative values of features in the training data. The main reason behind using normalizing the data is the fact that the variables measured at different scales do not contribute equally to the fitting process of

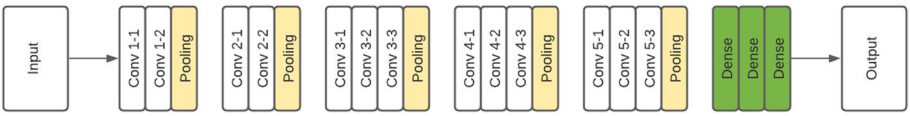

Fig. 3 Architecture map of VGG-16

the model and the chances of creating a biased model may increase. The equation used in the formulation is depicted in Eq. 1

$$X_{scaled} = \frac{X - \min(X)}{\text{Max}(X) - \text{Min}(X)} \tag{1}$$

3.3 Extreme gradient boosting

Extreme gradient boost (XGBoost) [16] has recently been popular among ensemble techniques due to its high performance and speed. Ensemble learning is a process in which a collection of predictor models works to provide better classification accuracy. Boosting models improve the accuracy by improving on the mis-classifications of previous models. In gradient boosting technique, in order to improve the mis-classifications of previous branches, the loss function is optimized. XGBoost models can run on single and distributed systems, has inbuilt cross validation, can handle missing values and outliers, can be used in regression and classification problems and has efficient memory management.

3.4 NuSVC

The Support Vectors machine classifier first defines the hyperplanes while maximizing the margin/distance between the classes. NuSVC works similar to the SVC but it provides a parameter Nu which is bound between the values 0 and 1 and is easily interpretable. In this technique, we have the flexibility to provide the number of support vectors. The Nu parameter represents the number of examples that the algorithm is allowed to misclassify or it can be defined as the maximum value of fraction of margin errors and minimum value of fraction of support vectors.

3.5 Stacking classifier based on K-NN, decision trees, Naïve Bayes and logistic regression

Stacking classifier is an Ensemble based technique where the output from multiple base classifiers is passed as input into the meta-classifier for the task of final prediction. The methodology is shown in Fig. 4. This technique has proved to be useful when the size of dataset is small. It can be very efficient in solving multi-classification problems. Stacking is different from bagging as the base learners are typically different and fit on the same dataset. It is also different from boosting as in stacking a single model is used to learn how to get the best combination of predictions from the base models. The base learning models (Level 0): K-Nearest Neighbors, Decision Trees and Naïve Bayes fit on the training data and the predictions are compiled. The meta-classifier Logistic Regression (Level 1) learns how to best combine the predictions of the base models.

3.5.1 Base Learner K-nearest neighbors

K nearest neighbors is a simple supervised learning algorithm that works by computing the distance between the query and all the samples in the dataset selecting the specific number of samples closest to the query and then votes for the most frequently occurring label.

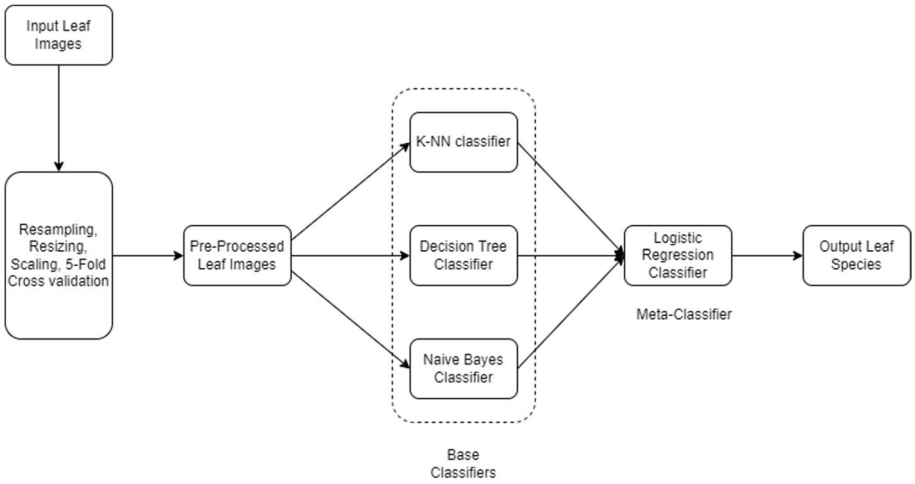

Fig. 4 Workflow of Stacking classifier to predict leaf species

3.5.2 Base learner decision trees

A decision tree is drawn upside down i.e the root is at the top. The tree grows as it splits into branches/edges based on an internal split condition. The end of the branch which does not split is the output/prediction. The cost function tries to find the greatest number of branches with similar responses. Gini score can be calculated as shown in Eq. 2

$$Gini\ score = sum(pk * (1 - pk)) \tag{2}$$

Gini score provides an idea about how good the split of decision tree is by calculating how mixed the output classes are in the groups created by the split. *pk* in Eq. (2) is the proportion of same class inputs present in the particular group.

3.5.3 Base learner Naïve Bayes

In classification problem, the hypothesis (h) is a class/species to assign to a new instance (d). Naïve bayes is based on the Baye's theoram which can be stated as in (3)

$$P(h|d) = \frac{(P(d \mid h) * P(h)\Big)}{P(d)} \tag{3}$$

Where, P(h | d) called Posterior probability is the probability of hypothesis h given data d

P(d | h) is the probability of d given h is true

P(h) called Prior probability of h is the probability of h being true (whatever may be the data)

P(d) is the probability of data (whatever the hypothesis may be)

3.5.4 Meta-classifier logistic regression

In case of stacking technique to create an ensemble of base learners, the input dataset for the meta-classifier (logistic regression shown in Fig. 4), is formed from the outputs of the base learners and the actual leaf species/class is the target variable for the meta-classifier. The meta-classifier is trained using the output of base learners which means that the output of the base learners forms the features set of the meta-classifier. Logistic regression is used here to solve multiclass classification problem by using one vs all approach. One-vs-All method denotes 1 to the class it is working on and 0 to all other classes. Sigmoid function is used to predict the output as a value from 0 to 1 with threshold at 0.5.(Eq. 4)

$$h = \frac{1}{1 + e^{-\Theta X}} \tag{4}$$

Where X is the input feature
Θ is the randomly initialised value
The above Eq. 4 can also be expressed as the following:

$$h = \Theta_0 + \Theta_1 X_1 + \Theta_2 X_2 + \Theta_3 X_3 + \Theta_4 X_4 + \ldots \tag{5}$$

where, X_1, X_2, X_3, are input feature and Θ is initialised randomly for each X.

This algorithm aims to update the value of Θ with each iteration so as to determine a relationship between the input features and the output label.

Cost function determines how far is the predicted value from the actual value shown in Eq. 6

$$cf = \frac{1}{m}\left[\sum y^i \log h^i + (1 - y)\log\left(1 - h^i\right)\right] \tag{6}$$

where m: Number of images in the training set; y is the actual output label; h is the predicted output label

The gradient Descent technique used to update the Θ value is given by:

$$\Theta = \Theta - \alpha \sum \left(h^i - y^i\right) X^i_{cf} \text{where } \alpha \text{ represents the learning rate} \tag{7}$$

4 Experimental results

The experiments were conducted on 11th Gen Intel core (TM) i7-11370H @ 3.30 GHz equipped with NVIDIA GeForce RTX 3070 GPU GDDR6 @ 8GB.

- Two Datasets (Swedish- 15 classes, Flavia -32 classes)
- Feature Extraction using VGG-16. Number of features extracted 25088.
- Division of dataset into training and testing.
- Classification using classifiers [XGBoost, NuSVC, Stacking(K-NN, Decision Tree, Naïve Bayes)]
- 5-Fold cross Validation on the training set.
- Hold out validation using the test set.

Table 1 Result of 5-Fold Cross Validation using XGBoost Classifier

Metric Name	Swedish (%)	Flavia (%)
Accuracy	97.40	93.94
Precision	97.62	94.47
Recall	97.40	93.67

Table 2 Result of XGBoost classifier on the unseen samples of test set

Metric Name	Swedish (%)	Flavia (%)
Accuracy	95.83	94.41
Precision	95.99	95.37
Recall	95.83	94.34

To evaluate the performance of the classifiers, we investigated three performances metrics through this research. The performance metrics Accuracy, Precision, Recall is presented from Eqs. (9) to (10).

$$Accuracy = \frac{True\ Postive + True\ Negatives}{(True\ Positive + False\ Positive) + (True\ Negative + False\ Negative)} \tag{8}$$

$$Precision = \frac{True\ Postive}{True\ Positive + False\ Positive} \tag{9}$$

$$Recall = \frac{True\ Postive}{True\ Positive + False\ Negative} \tag{10}$$

Subsection 4.1 will present the performance of the XGBoost classifier. Subsection 4.2 will present the performance of the NuSVC classifier. Subsection 4.3 will present the performance of the stacking classifier.

4.1 Performance of the XGBoost classifier

The experiments were conducted on two datasets Swedish and Flavia separately. The validation was applied on two levels one being 5-Fold cross-validation and other being hold out validation. Both datasets were first divided into training and testing datasets to perform hold out validation and then the training set was divided into K folds to perform K-Fold cross validation. Table 1 presents the results of applying 5Fold cross validation on both datasets using XGBoost classifier. The mean accuracy obtained was 97.40% on Swedish dataset with 15 classes and 93.94 % accuracy on Flavia dataset with 32 classes, across the 5 folds. When the experiment was performed on the unseen data of the testing set then the accuracies obtained were 95.83% on the Swedish dataset and 94.41% on Flavia dataset (Table 2).

4.2 Performance of the NuSVC classifier

The same experiments that were conducted on the two datasets using the XGBoost classifier were performed using the NuSVC classifier. Table 3 presents the results of applying 5-Fold cross validation on both datasets using NuSVC classifier. The mean accuracy obtained was 99.11% on Swedish dataset with 15 classes and 95.57 % accuracy on Flavia dataset with 32 classes, across the 5 folds. When the experiment was performed on the unseen data (refer Table 4) of the testing set then the accuracies obtained were 93.33% on the Swedish dataset and 95.81% on Flavia dataset. It can be observed that both the classifiers have performed at par with each other as the NuSVC performed better than XGBoost on both datasets during the 5-Fold cross validation and vice versa while testing on the test set.

4.3 Performance of the stacking classifier

Table 5 shows the accuracy, precision and recall obtained when the species of the Swedish leaf dataset were classified using the stacking classifier and validated using the 5-Fold cross validation. The dataset consists of 15 species. For the hold out validation, after dividing the dataset into testing and training, the training set images were further divided into 5-Folds for 5 fold cross validation. It was observed that K-NN performs slower than others and decision tree is faster. Though Decision tree is more flexible but it is pruned in the learning process and thus leads to loss of accuracy. K-NN also performs better in this case as the dataset has 15 classes and there are around 50 images per class. Table 6 shows the accuracy, precision and recall obtained when the species of the Flavia leaf dataset were classified using the stacking classifier and validated using the 5-Fold cross validation. The dataset consists of 30 species. For the hold out validation, after dividing the dataset into testing and training, the training set images were further divided into 5-Folds for 5-fold

Table 3 Result of 5-Fold Cross Validation using NuSVC Classifier

Metric Name	Swedish (%)	Flavia (%)
Accuracy	99.11	95.57
Precision	99.18	96.05
Recall	99.11	95.42

Table 4 Result of NuSVC classifier on the unseen samples of test set

Metric Name	Swedish (%)	Flavia (%)
Accuracy	93.33	95.81
Precision	95.11	95.84
Recall	93.33	95.83

Table 5 Result of 5-Fold Cross validation on Swedish Dataset using Stacking

Metric Name	K-NN	Decision Tree	Naïve Bayes	Stacking
Accuracy	98.18	77.92	95.18	98.62
Precision	98.33	79.24	95.54	98.78
Recall	98.18	77.92	95.18	98.62

Table 6 Result of 5-Fold Cross validation on Flavia Dataset using Stacking

Metric Name	K-NN	Decision Tree	Naïve Bayes	Stacking
Accuracy	91.67	74.63	94.43	95.77
Precision	93.37	75.77	95.02	96.04
Recall	91.16	73.88	94.17	95.61

cross validation. It was observed that Decision trees gives same low performance compared to K-NN and Naïve Bayes. Naïve bayes gives better performance when compared to individual classifiers. It is relatively simple classifier, converges faster and works well even with less training data. From both Tables 5 and 6, it was observed that the stacking classifier outperforms the individual classifiers as in this case, the predictions from the component classifiers are input into the Logistic regression classifier for the final prediction. From Tables 5 and 6 it is noticed that we can leverage the advantages of Logistic regression like its simplicity. The convex loss function helps to overcome the problem of local minima. Thus, it proves to be very useful when the number of features are large (as in this case, where the features are derived from the VGG deep learning model) and the training data is not huge. The stacking classifier showed an accuracy of 98.62% in the case of Swedish dataset and an accuracy of 95.77% in case of Flavia dataset.

Further in order to carry out hold out validation where the model is tested on samples that were not used in training, called the unseen samples, the data was divided into training and testing data. The model was tested on the testing data to check if it can detect the patterns and predict the actual species of the plant. An accuracy of 99.16% and 98.13% was observed in case of Swedish and Flavia datasets respectively. The performance comparison of three classifiers namely XGBoost, NuSVC and stacking on both datasets was depicted in Fig. 5. As shown, stacking classifier outperforms others on both datasets thus showing promising results (Table 7).

As the datasets in the study are freely available, various other techniques have been proposed in the literature as shown in Table 8. A comparable accuracy has been

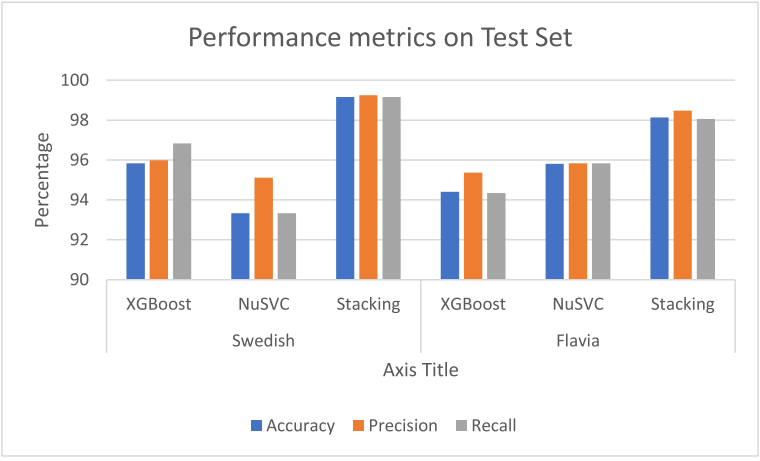

Fig. 5 Performance of the classifiers on the unseen samples of the test set (Hold out validation)

Table 7 Result of Stacking classifier on the unseen Test Set

Metric Name	Swedish	Flavia
Accuracy	99.16	98.13
Precision	99.25	98.48
Recall	99.16	98.06

Table 8 Comparison with Existing works on the same datasets

Reference	Dataset	Technique	Accuracy (%)
The proposed study	Flavia	VGG+Stacking classifier	98.13
	Swedish		99.16
[5]	Flavia	Texture based Ffirst and Inception-resNet-V2	99.6
	Swedish		99.7
[4]	Flavia	Concat(MobileNet, DenseNet)+ LDA, LR, CART, bagging, MLP	98.71
	Swedish		99.4
[3]	Flavia	CNN-26 layer deep	99.6
[7]	Swedish	ICL+ Hausdorff distance	98.4
[17]	Flavia	Modified Local Binary Pattern + KNN	97.55
	Swedish		96.83
[18]	Flavia	Shape descriptors+ ANN	96
[19]	Flavia	Multiclass SVM	97
[20]	Flavia	CNN	96.5

observed in few papers where in one the authors used concatenation of feature vectors from 2 CNNs MobileNet and DenseNet121[4] to input into conventional classifiers, in another [5] a texture based feature descriptor and a ResNet Inception V2 model was proposed and in another a 26 layer deep CNN model with residual blocks was proposed [3]. When compared to the forementioned work, our approach is less memory intensive as the size of the VGG is 16 layers compared to other deeper layered models like ResNet and DenseNet and so the number of features generated are less in comparison. The choice of classifiers taken into the stack are simple classifiers which do not require too many hyperparameters. The stacking approach makes use of logistic regression classifier to predict the final result from the predictions of other classifiers in the stack which makes the approach more robust.

As the proposed method makes use of Convolutional neural networks, there arises a need for high hardware demands like GPUs in order to enhance the performance. The training time also depends on the depth of the model taken into account. There is a scope for more experiments based on other CNN models for extraction of features and other combination of classifiers on other leaf image datasets.

The experiments performed in the mentioned setting shows that the stacking classifier outperformed the machine learning classifiers like local binary patterns, KNN, shape descriptors and Artificial neural networks, multiclass SVM and Convolutional neural networks.

5 Conclusion and future scope

The use of plants for any purpose requires their robust identification. Through this work, we have proposed a stacking classifier to classify plant leaves into their respective species. The hybrid model makes use of deep learning technique to extract features from the plant leaves and then the classification is performed using the traditional Machine learning classifiers namely XGBoost, NuSVC and the Stacking classifier consisting of the meta classifier (Level 1) Logistic Regression being trained by the output of the Level 0 classifiers (K-NN, Decision Trees and Naïve Bayes). The experiments were performed on two freely available datasets Swedish and Flavia which consists of 15 and 32 classes respectively. We have also performed the validation of the techniques on two levels. First the dataset was divided into training and testing and then K-Fold Cross validation was employed on the training set. The validation accuracy achieved on the unseen samples of testing set was 99.16 % and 98.13 % on Swedish and Flavia respectively. The comparison with existing works (which contains deep learning and machine learning techniques) has been presented in Table 8.

It can be concluded that the combination of deep learning and classical machine learning techniques can help solve many real-world problems and that too with greater accuracy. The future scope of our research will be to create a medicinal leaf dataset and also to detect the medicinal leaves in their natural habitats.

Data Availability The datasets used for the study are publicly available. Name of datasets and references have been provided under Section 3.1

Declarations

Conflicts of interests Authors declare no competing interests.

References

1. Wäldchen J, Mäder P (2018) Plant species identification using computer vision techniques: a systematic literature review. Arch Comput Methods Eng 25:507–543. https://doi.org/10.1007/s11831-016-9206-z
2. Sachar S, Kumar A (2020) Survey of feature extraction and classification techniques to identify plant through leaves. Expert Syst Appl 114181. https://doi.org/10.1016/j.eswa.2020.114181
3. Sun Y, Liu Y, Wang G, Zhang H (2017) Deep learning for plant identification in natural environment
4. Pearline A, Sobitha S, Vajravelu SK (2019) DDLA : dual deep learning architecture for classification of plant species. https://doi.org/10.1049/iet-ipr.2019.0346
5. Šulc M, Matas J (2017) Fine - grained recognition of plants from images. Plant Methods 1–14. https://doi.org/10.1186/s13007-017-0265-4
6. Ni F, Wang B (2018) Integral contour angle: An invariant shape descriptor for classification and retrieval of leaf images. Proc - Int Conf Image Process ICIP:1223–1227. https://doi.org/10.1109/ICIP.2018.8451605
7. Wang B, Gao Y (2014) Hierarchical string cuts: A translation, rotation, scale, and mirror invariant descriptor for fast shape retrieval. IEEE Trans Image Process 23:4101–4111. https://doi.org/10.1109/TIP.2014.2343457
8. Turkoglu M, Hanbay D (2019) Leaf-based plant species recognition based on improved local binary pattern and extreme learning machine. Phys A Stat Mech its Appl 527:121297. https://doi.org/10.1016/j.physa.2019.121297
9. She Y, Wang X, Gui T, Cai J (2019) Lawn plant identification and segmentation based on least squares support vector machine and multifeature fusion. J Electron Imaging 28:1. https://doi.org/10.1117/1.jei.28.2.023034
10. Ward D, Moghadam P, Hudson N (n.d.) Deep leaf segmentation using synthetic data

11. Bodhwani V, Acharjya DP, Bodhwani U (2019) Deep residual networks for plant identification. Procedia Comput Sci 152:186–194. https://doi.org/10.1016/j.procs.2019.05.042
12. Beikmohammadi A, Faez K (2018) Leaf classification for plant recognition with deep transfer learning. Proc - 2018 4th Iran Conf Signal Process Intell Syst ICSPIS 2018 21–26. https://doi.org/10.1109/ICSPIS.2018.8700547
13. Tan JW, Chang SW, Abdul-Kareem S et al (2020) Deep learning for plant species classification using leaf vein morphometric. IEEE/ACM Trans Comput Biol Bioinforma 17:82–90. https://doi.org/10.1109/TCBB.2018.2848653
14. Söderkvist OJO (2001) Computer vision classification of leaves from swedish trees. 74
15. Wu SG, Bao FS, Xu EY, et al (2007) A leaf recognition algorithm for plant classification using probabilistic neural network. ISSPIT 2007 - 2007 IEEE Int Symp Signal Process Inf Technol 11–16. https://doi.org/10.1109/ISSPIT.2007.4458016
16. Chen T, Guestrin C (2016) XGBoost: A scalable tree boosting system. In: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Association for Computing Machinery, pp 785–794
17. Naresh YG, Nagendraswamy HS (2016) Classification of medicinal plants: An approach using modified LBP with symbolic representation. Neurocomputing 173:1789–1797. https://doi.org/10.1016/j.neucom.2015.08.090
18. Aakif A, Khan MF (2015) Automatic classification of plants based on their leaves. Biosyst Eng 139:66–75. https://doi.org/10.1016/j.biosystemseng.2015.08.003
19. Ibrahim Z, Sabri N, Nabilah N, Mangshor A (2018) Leaf Recognition using Texture Features for Herbal Plant Identification. Indones J Electr Eng Comput Sci 9:152–156. https://doi.org/10.11591/ijeecs.v9.i1.pp152-156
20. Jasitha P, DIleep MR, DIvya M (2019) Venation Based Plant Leaves Classification Using GoogLeNet and VGG. 2019 4th IEEE Int Conf Recent Trends Electron Information, Commun Technol RTEICT 2019 - Proc 715–719. https://doi.org/10.1109/RTEICT46194.2019.9016966