# Piper Plant Classification using Deep CNN Feature Extraction and Hyperparameter Tuned Random Forest Classification

***A. Pravin***[1]***, C. Deepa***[2]

[1]Research Scholar, Department of Computer Science, Sri Ramakrishna College of Arts and Science, Coimbatore, India.

[2]2Head, Department of Computer Science (Artificial Intelligence & Data Science), Sri Ramakrishna College of Arts & Science, Coimbatore, Tamilnadu, India.

**Correspondence:** Email: pravinresearchscholar1@gmail.com

**Received** 10 July, 2022; Revised August 27, 2022; Accepted August 27, 2022

**Available online** August, 2022 at www.atlas-tjes.org, doi: 10.22545/2022/00202

*The plant has numerous uses in medicine, food, and industry and plays a major role in environmental protection. Hence it is crucial to identify and classify the specific plant species. In the agriculture production and botanical area, plant classification for images of leaves is considered the basic research. Due to the higher dimensionality and nature complexity of leaf image data, various effective algorithms are required to perform the classification of specific plant species. Hence, in this study, all plant types and specific piper plant types are considered and classified based on the Hyperparameter tuned random forest algorithm due to its effective optimal hyperparameter tuning. Piper plants are selected in this research since they possess significant medicinal applications. Significantly the Deep CNN approach is considered an effective feature extraction of all plants such as tomato, apple, cherry, and others and also piper plants like piper mulesa, piper nigrum, and others. However, initially the effective pre-processing of data augmentation to reduce overfitting and increase the amount of data and feature scaling for data features normalization are established. The experimental results show that the proposed hyperparameter tuned random forest classifier shows better results of showing an accuracy value of 0.94 for all plants and 0.88 value for piper plant compared with other machine learning algorithms like SVM, naïve Bayes, and Logistic regression.*

**Keywords**: Data mining, piper plant classification, hyper parameter tuned random forest, medicinal applications, and deep CNN.

## 1 Introduction

Identifying plant species is not only exclusive to plant ecologists and botanists. It is valuable or vital for society from various professionals to the public. However, identifying plants through traditional means is time-consuming and difficult due to diverse botanical terms. This also makes it complex to overcome the obstacles for novices who are interested in attaining knowledge of plant species. Recently, research on computer science encompassing pattern recognition, image processing and DL (Deep Learning) has been incorporated into diverse plant classification for solving the deficiency in people's ability to identify plant species (Wäldchen, Rzanny, Seeland, & Mäder, 2018). At present, applying DL in classifying plants has become the active research area in which the image dataset has been the pre-requirement. Accordingly, a special focus on identifying piper plant types is also essential as it possesses significant medicinal applications. So, DL has been recommended to identify piper plants. DL algorithm has been the ML (Machine Learning) class algorithm which utilized multiple layers for extracting high-level features from input. Through this method, various types of plants could be identified more easily than in the traditional system to precision and speed. Recommended system has been applied to datasets and empirical images that explored its efficacy (Pravin & Deepa, 2021; Rajesh & Bhaskari, 2021).

Different plant types have been focused on by conventional research. Correspondingly, a CNN architecture has been suggested for classifying leaf images of ladies' fingers into three kinds such as disease, leaf burn, and healthy(Chen, Zhang, Nanehkaran, & Li, 2020; Selvam & Kavitha, 2020). The dataset comprised 1088 samples that have been taken from farms of various villages in Tiruvannamalai. Results explored better accuracy (Hang, Zhang, Chen, Zhang, & Wang, 2019). On contrary, a few-shot learning methodology relying on the Siamese network has been endorsed to resolve the leaf classification issues. Initially, features of varied images have been extracted by CNN (Convolutional Neural Network) with weight-sharing. Subsequently, this network utilized the loss function for learning metric space. Through this, identical and different leaf samples have been partitioned.

Additionally, SSO (Spatial Structure Optimizer) technique has been suggested to build metric space that will assist to enhance leaf classification accuracy. Lastly, K-NN (K-Nearest Neighbour) has been employed for classifying leaves in the learned space of metrics. Performance has been analyzed by using Swedish, Leafsnap, and Flavia datasets for evaluating it. Empirical outcomes reveal that the suggested system could accomplish maximum accuracy for classification with minimum supervised samples (Wang & Wang, 2019). As different approaches have been employed by traditional works, sinuosity coefficients have been recommended to describe leaf shape and perform classification based on it. Accuracy is satisfactory through this system (Kala & Viriri, 2018). On contrary, SVD (Singular Value Decomposition) and SR (Sparse Representation) have been integrated to recognize plant types. Better results have been attained.

However, other features like texture, color, and shape of leaf images have to be included in sparse representation for improving performance (Zhang, Zhang, Wang, & Kong, 2018). In addition, MSF-CNN (Multi-Scale Fusion CNN) has been suggested to recognize plant leaves. Accuracy has been satisfactory (Hu, Chen, Yang, Zhang, & Cui, 2018). Moreover, an approach having five stages has been suggested for classifying leaf shapes for identifying plant species through venation detection. These stages include canny edge detection, leaf boundary removal, curve extraction, produce hue image normalization, and finally, image fusion. Outcomes showed 91.06% accuracy for the Acer dataset (Kolivand, Fern, Saba, Rahim, & Rehman, 2019).

Though existing studies attempted to perform plant type classification, to the best of our knowledge, only a few studies tried to classify piper plants. While other studies focus on the classification of plant diseases and their diagnosis. Traditional works which focussed on classifying piper plants have also been limited in terms of accuracy. Considering this, the present study intends to perform a classification of piper plants and identify them with good accuracy than conventional systems by accomplishing the below objectives based on DM (ML and DL) and image processing. Typically, DL eliminates data labeling and provides high-quality results with effective accuracy. It also removes the necessity of feature engineering and analyses unstructured data efficiently. It also possesses several applications such as image processing, speech recognition, etc. Automatic learning could be accomplished by DL-based methods which makes it robust. Deep CNN proposed in this study is computationally effective and performs automatic learning. Moreover, ML-based methods like RF (Random Forest) perform better data handling and perform effective classification. All these advantages make the proposed system explore better outcomes than traditional systems which are proved through results.

The major contributions of this study are:

- To perform feature extraction using the proposed Deep CNN model to obtain only significant features for improving the classification performance.
- To classify piper plants through the introduced hyperparameter tuned RF (Random Forest) Classifier for enhancing the prediction accuracy.
- To analyze the performance of the introduced system by comparing it with other ML algorithms in terms of significant metrics for validating the efficiency of the proposed system thereby exploring its efficacy in piper plant classification.

# 2 Review of Existing Work

Various traditional works are intended to classify the plant types through different approaches and methodologies. The results explored in these systems are presented in this section.

A method based on CNN named D-Leaf has been suggested for classifying the varied plant types from leaf images. These images have been pre-processed. Then, features have been extracted through three CNN models termed fine-tuned Alex-Net, D-Leaf, and pre-trained Alex-Net. Finally, classification has been performed by five ML (Machine Learning) methods such as SVM (Support Vector Machine), K-NN (K-Nearest Neighbour), CNN, NB (Naïve Bayes), and ANN (Artificial Neural Network). Moreover, the traditional morphometric technique calculated morphological measurements by relying on Sobel segmented veins. This has been undertaken for the benchmark. Analysis revealed the efficient performance of CNN models to conventional morphometric measurements at a rate of 66.55%. Extracted features from CNN fitted better with ANN. It has been concluded that D-Lea could be an efficient system to identify plant species as explored by empirical outcomes (Wei Tan, Chang, Abdul-Kareem, Yap, & Yong, 2018).

In addition, data augmentation has been undertaken by employing CNN to find twelve plant species through various image transforms like rotate, scaling, histogram equalization, flip and resize. The outcome has been significant subsidization to persistent progress in the agricultural domain and the widespread aim of augmenting agricultural yield worldwide (Alimboyong, Hernandez, & Medina, 2018). Various approaches have been utilized by different researchers, accordingly, DL (Deep Learning) and the conventional system has been undertaken to recognize plant species(Ferentinos, 2018). In a conventional system, feature extraction has been undertaken through LBP (Local Binary Pattern), CCS (Colour Channel Statistics), haralick texture, and shape features. These features were then classified by classifiers like LR (Logistic Regression), NB, RF (Random Forest), Bagging, K-NN, LDA (Linear Discriminant Analysis, regression, and classification. Outcomes revealed the more effective performance of CNN models than conventional techniques (Anubha Pearline, Sathiesh Kumar, & Harini, 2019). Further, DenseNet-121 and Mobile-Net have been utilized as the feature extraction method and ML classifiers have been employed for classification. Swedish leaf, Folio, and Flavia have been used as benchmark datasets.

The suggested methodology accomplished maximum accuracy for custom and standard datasets (Raj & Vajravelu, 2019). Similarly, an approach to specifically classify tomato has been explored with a dataset encompassing 5,266 images having seven species of tomatoes. CNN has been utilized for this classification which showed 93% testing accuracy (Alajrami & Abu-Naser, 2020). Likewise, valuable features of the leaf have been learned from raw input data representation through CNN to attain a good perception of the selected features relying on DN (De-Convolutional Network). Few unexpected results have been reported. Different venation orders are the effective representative features in comparison to outline shape(Mostajer Kheirkhah & Asghari, 2019).

Moreover, multi-level representation explored the feature's hierarchical transformation of abstraction from low-level to high-level based on the species classes. These findings have been suitable for defining hierarchical leaf characters which assisted in gaining insights into the design of hybrid models for feature extraction. It also can further enhance the classification performance of plant systems (Lee, Chan, Mayo, & Remagnino, 2017). As leaves vary in shape, surface, and hue, few studies attempted to classify heterogeneous leaves through the collection of methods by utilizing different visual attributes. A progressive method has been considered that encompasses pre-processing scheme for scale normalization and introduces different leaves, a hue analysis stage that is comprised of hue highlight extraction, a shape analysis stage that includes shape relying on the representation, and a surface analysis scheme to explore surface samples of the leaf surface. Each layer includes modules to treat different sorts of leaf and classification to pick the appropriate module for further preparation. Grouping based on NFC has been done for exploiting fuzzy likeness amongst the overall leaves for shape layers and hue.

While Euclidean separation has been used to segregate surface elements. The results have explored the better performance of the suggested system (Chaki, Parekh, & Bhattacharya, 2020). In addition, LeafNet has been endorsed in (Barré, Stöver, Müller, & Steinhage, 2017)(Goyal & Kumar, 2021). Publicly available datasets like Flavia, Foliage and LeafSnap have been used for evaluation. Assessing the detection accuracy corresponding to LeafNet on Flavia, Foliage and LeafSnap explored the better performance of the endorsed system than conventional works. On the contrary, plant species that have been captured as photos have been identified through CNN. Different factors impacting the performance of the networks have been assessed by DL architectures like VGGNet, AlexNet, and GoogLeNet. Pre-trained models have been fine-tuned by TL (Transfer Learning) through LifeCLEF 2015 dataset. To minimize the overfitting issue, data augmentation methods have also been employed that rely on image transforms like translation, scaling, rotation and reflection. Furthermore, the parameters of the network have been adjusted and varied classifiers have been fused for enhancing the overall performance(Yang, Zhong, & Li, 2020). The best-integrated system accomplished an 80% accuracy rate (Ghazi, Yanikoglu, & Aptoula, 2017).

To classify plants on a large scale, a DL model having 26 layers, and 8 residual building blocks have been recommended that achieved a 91.78% recognition rate revealing that DL has been a promising system (Sun, Liu, Wang, & Zhang, 2017; Zhang, Wang, & Huang, 2017). Moreover, to classify plant species even from an occluded image of a leaf, an approach has been suggested that uses a database consisting of known plant species with various leaves. This identification has been accomplished by DCE (Discrete Contour Evolution) and parameters of similarity transformation like rotation, uniform scaling, and translation which exhibited better results (Chaudhury & Barron, 2018). Additionally, a classifier has been recommended to classify plant species through MLP (Multi-Layer Perceptron)-Adaboosting.

This approach encompassed various stepwise processes like pre-processing which has been employed to set the leaf images for further processing. This is then followed by feature extraction and selection. Several morphological features like axis length, orientation, perimeter, and centroid have been extracted from various leaf digital images. Finally, classification has been performed with classifiers such as K-NN, MLP, and DT (Decision Tree) for testing the algorithm's accuracy. Furthermore, Adaboost has been used to enhance the precision of the suggested system (Kumar, Gupta, Gao, & Singh, 2019).

Correspondingly, features of plant leaves have been extracted and the species have been identified that rely on image processing. Initially, leaf images have been segmented by different techniques, and subsequently, the feature extraction method has been utilized for extracting features of texture and leaf shape from sample leaf images. Then, complete characteristic features of the plant leaves have been formed based on characteristic information. Testing has been carried out by 50 leaf databases which have then been compared with the K-NN classifier, SVM, and Kohonen network relying on SOM (Self-Organizing Mapping).

Concurrently, leaves of varied plans have been compared. Outcomes revealed that ginkgo leaves have been identified easily. Good recognition impact has been accomplished for images of the leaf having complex backgrounds. Test set samples have been taken as input to attain reconstruction errors. Test set class label could be attained by reconstruction of DL model with small error sets. Results revealed that the suggested methodology possessed minimum recognition time and maximum rate of recognition. But, Accuracy needs to be improved further (Huixian, 2020)(Chau et al., 2017).

## 2.1 Problem Identification

Various problems identified through the analysis of the traditional works are listed below.

- Traditional work (Wei Tan et al., 2018) suggested including many plant species having compound leaves. It also stated to include several taxonomic features for enhancing the identification process of plant species that could be utilized by botanists.

- The conventional system (Alimboyong et al., 2018)endorsed to use model by training with other plant kinds such as herbal and medicinal plants.
- Accuracy needs to be further improved for the correct identification of plant species. Existing works showed 66.55% accuracy (Anubha Pearline et al., 2019), 80% accuracy (Ghazi et al., 2017), and 93% (Alajrami & Abu-Naser, 2020)based on the utilized methods.
- The study recommended employing the recently successful DL method to solve the accuracy issues associated with the classification of plant types (Chaudhury & Barron, 2018).

# 3 Materials and Methods

In the proposed model, the piper and other plant datasets are utilized to classify the piper plant types shown in Figure 1. Initially, the pre-processing is performed with data augmentation and feature scaling approach. Further, the feature extraction is performed for the pre-processed data using the Deep CNN approach and further classification is performed using Hyperparameter tuned Random Forest classifier. The plant type is predicted and checked as a piper or other plants. If it is not the piper plant then the performance analysis is exhibited for the proposed model whereas if it is the piper plant then again feature extraction is performed for the data to extract and classify the piper plant type using the similar Deep CNN and hyperparameter tuned Random Forest classifier. In this case, the proposed model is enabled with feature extraction that is embedded with the process called fine-tuning, this process is attained on the raw data directly. The degree of feature extraction is observed to be high from the training data and also from the data that have reduced variance.

Notably, CNN is capable of performing a better feature selection process when compared with the existing feature selection algorithms. ML algorithms are observed to be easy in overfitting massive data environments. Thus, CNN would be the best fit for the massive network environment. When compared with other deep learning algorithms, a huge advantage of CNN is that it shares similar convolutional kernels which will aid in reducing the calculation and the number of parameters. This in turn helps the recommended random forest to learn effectively and accurately. In order to achieve high accuracy and efficiency, the proposed work has considered CNN for feature extraction and RF as a classifier. Finally, the performance of the proposed model is checked with piper plants and other plant types based on their leaves. The piper plant and other plant leaves used in the model are shown in the following Figure 2.

In this study piper and other plants, leaf database is utilized comprising 3663 images of piper plant and other plant species as shown in figure.3. Every piper plant leaf type contains nearly 100 images. Every leaf image is considered a white background RGB image and 75*75 pixels is the size of the image. The image resolution is 75*75.

## 3.1 Pre-processing

In this stage, initially, the white background is converted into a black color background for removing the feature extraction effect. The thresholding algorithm is used to complete the background color modification. After the three color channels threshold value setting, if the RGB channel value is higher compared with the threshold value then the value is determined as 0 and the value is determined as reserved if it is smaller than the threshold value. All images are resized into 64x64 since the network needs all images to be of similar size. Image scaling should reach a suitable accuracy rate. To improve the efficiency of training, the mean value is removed by every image. All the image mean value is calculated and then every image is subtracted from the mean value. Every leaf character is highlighted in this stage and thus the accuracy rate is improved.

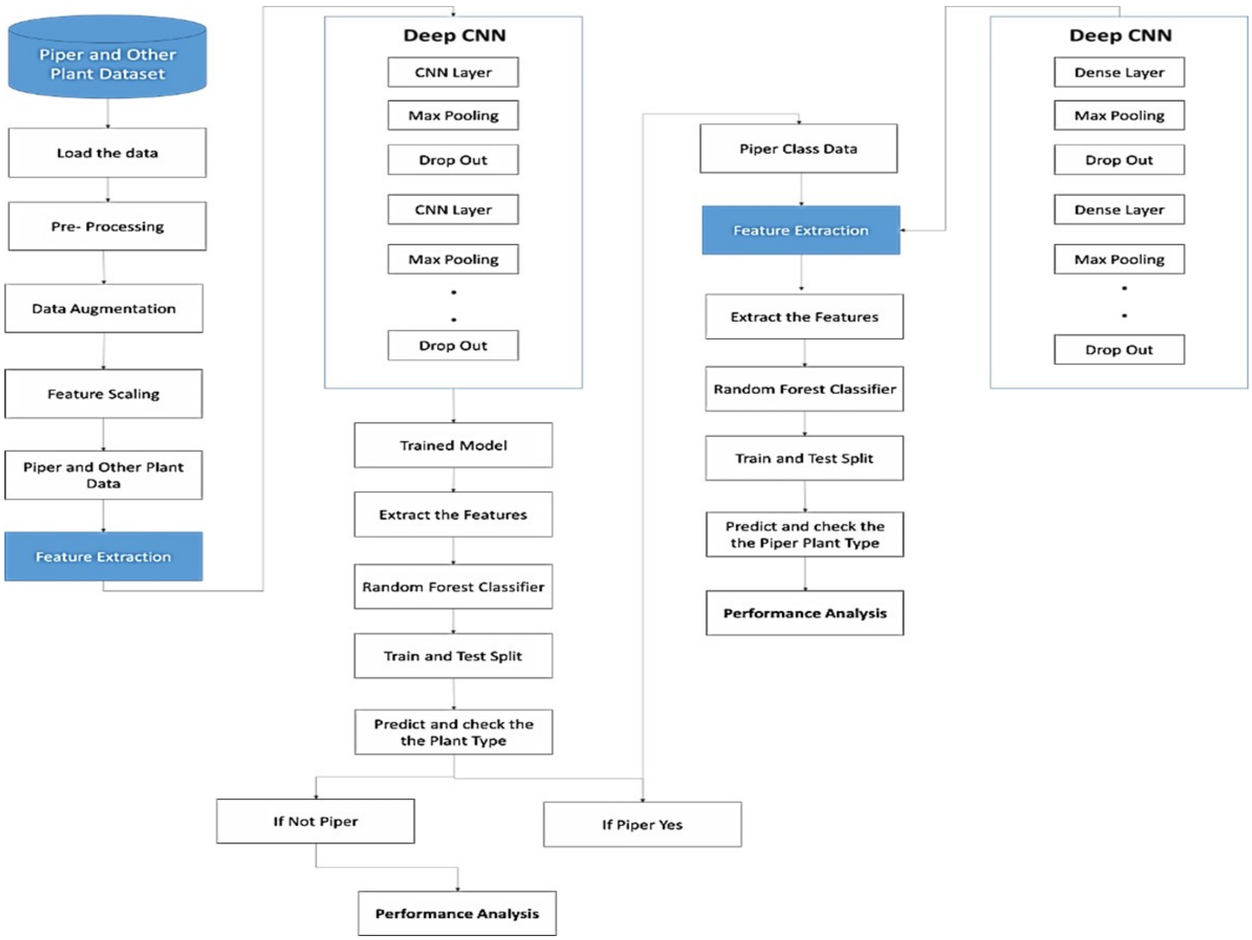


**Figure 1:** *Proposed flow.*

### 3.1.1 Data Augmentation

CNN performance is affected directly due to the database size. A huge amount of data is needed as the small size might over-fit the training set easily. Data augmentation enlarges the training set and addresses the discussed problem. Certain approaches like scaling, vertical flip, noise, rotation, horizontal flip, and color jittering are engaged for data augmentation. For reliable predictions, a lot of training data is required for deep learning models and it is not available always. To make a better-generalized model, the existing data is augmented.

### 3.1.2 Feature scaling

Feature scaling is an approach for independent features standardizing presented in the fixed range data. During the data pre-processing it has been performed to address greatly differentiating magnitudes, units, or values. If feature scaling is not performed then the greater values are weighed by the machine learning algorithm and the lower values are considered by smaller values without regarding values units. Every kind of leaf has 1607 and 2056 piper and other plant images and has been divided into a training set of 80% and a test set of 20%.

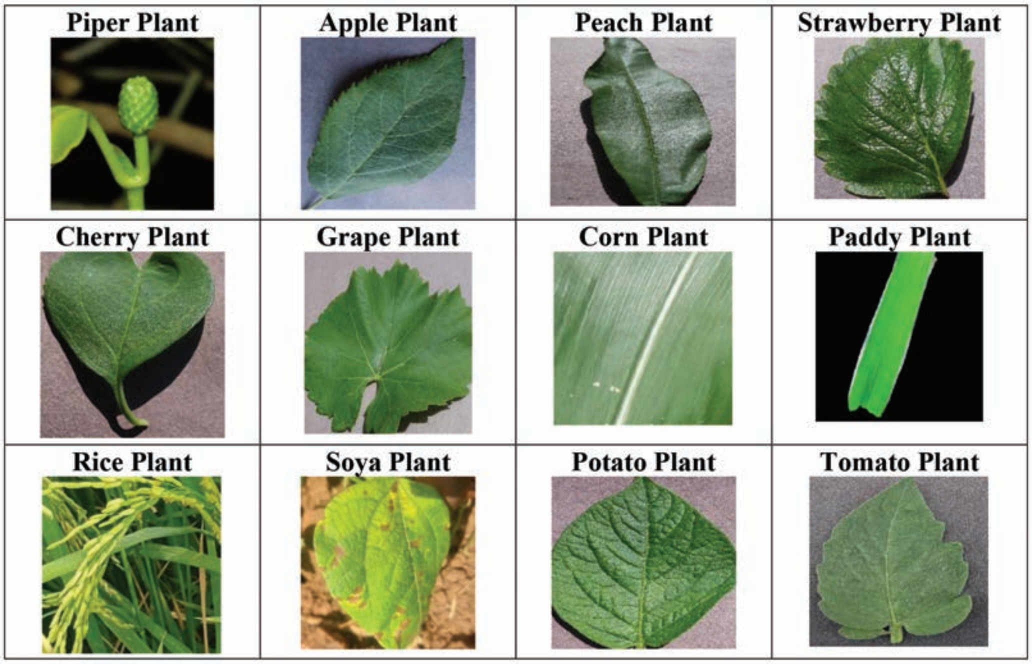


**Figure 2:** *All plants (All plants (Piper and other plant type leaves).*

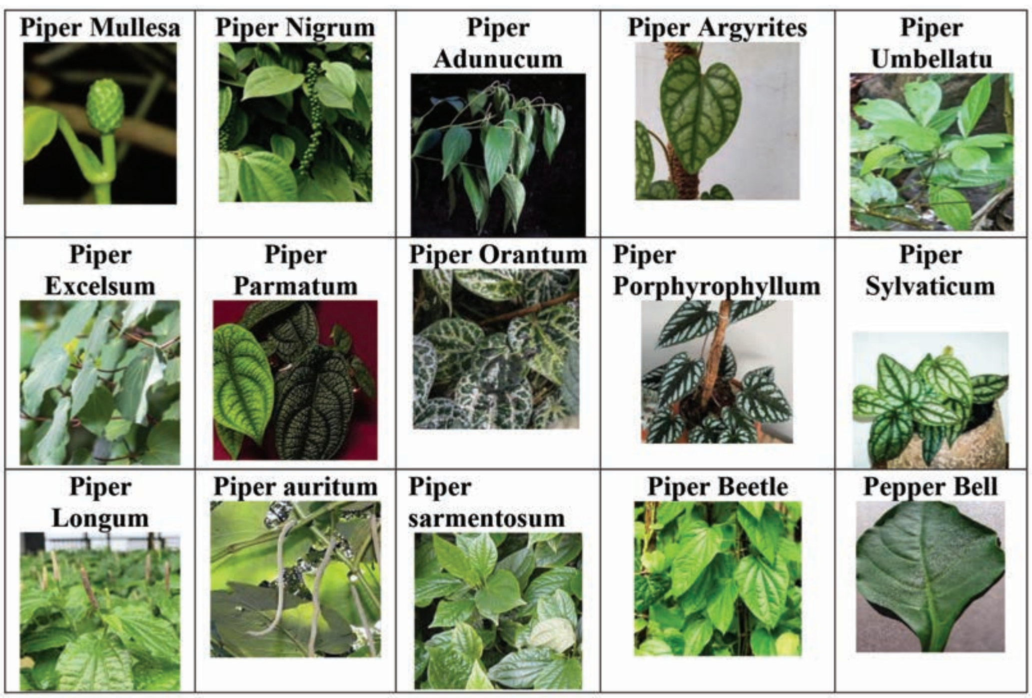


**Figure 3:** *Specific Piper plant leaves.*

## 3.2 Feature Extraction using Deep CNN:

The feature extraction of piper and other plants is performed by using the Deep CNN (Deep Convolutional Neural Networks) model which is inspired by visual system structure, in specific tops at classification and object recognition. Moreover, CNN can reserve the spatial locality and kernel size relation input. Deep neural network architecture shows greater non-linear nature. Subsequently, deep CNN is highly suited to manage the non-linear spectral-spatial evaluation and high dimensional hyperspectral image difficulties. The deep CNN architecture is established shown in Figure 4 to retrieve the hyperspectral images' spectral-spatial features because of the imbalance among the massive parameters and limited labeled samples. And the Piper-256 and Normal plant -10 extracted feature on the basis of the size, depth, and colour.

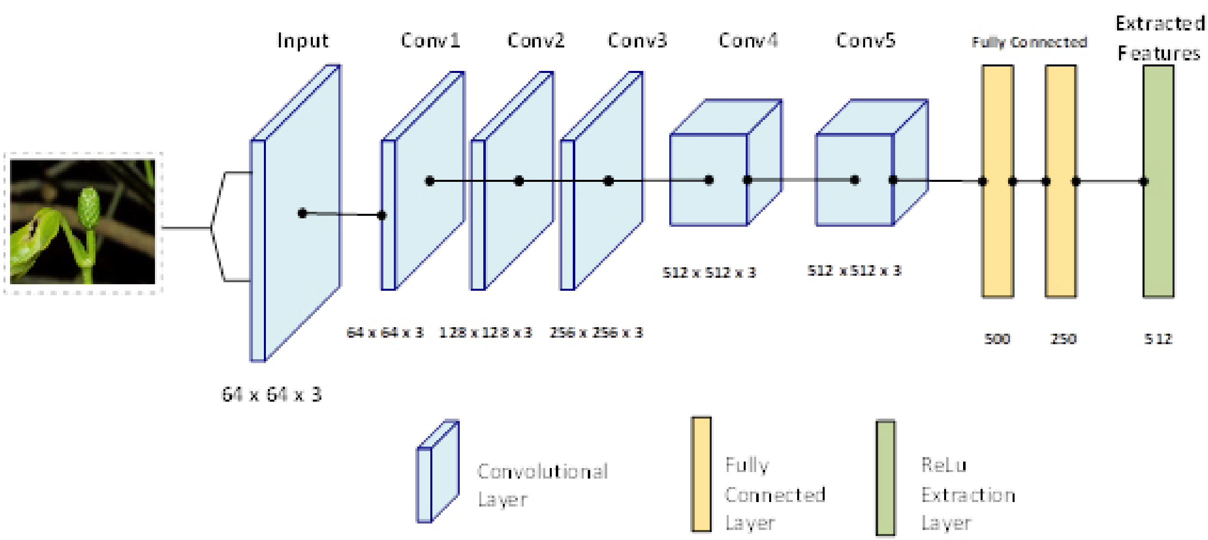


**Figure 4:** *Deep CNN Architecture.*

### 3.2.1 Convolutional Layer

Statistically, the CNN convolutional layer is defined as,

$$\mathrm{e}^1 = \mathrm{g}(\mathrm{F}^1 * \mathrm{e}^0 + \mathrm{bias}^1) \tag{1}$$

From Eq.(1), the convolutional operator is *, input hyperspectral image cube determines as $\mathrm{e}^0$, $\mathrm{F}^1$ and $\mathrm{bias}^1$ are convolutional layer filters and bias and rectified linear unit (ReLU) activation function denoted as g(.), convolutional layer output feature maps and pooling layer input denotes as $\mathrm{e}^1$. For classification, the hyperspectral images provide a spatial and spectral information wealth. The classification performance improved by the spatial and spectral information is broadly utilized. The pixel with the kernel size of 9x9xB has been selected in which several hyperspectral image bands are B and the kernel size is 9x9. Better accuracy is attained through choosing a greater kernel size as input. But it requires higher training to train the network with the input of higher kernel sizes.

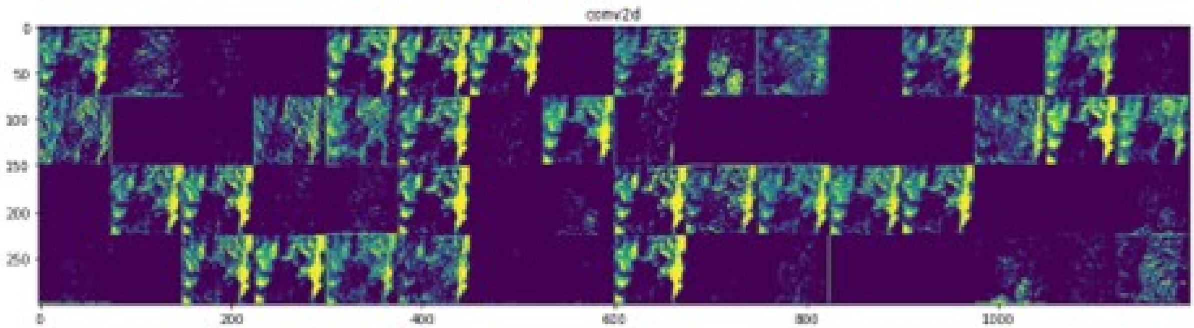


**Figure 5:** *Convolutional layer- feature maps of one plant image (sample).*

To extract the local features, multiple filters are used by the convolutional filter to assure the diversity of the features. There exists a strong correlation between the hyperspectral image of various bands and limited labeled samples, and for convolutional layers, the number of output bands set as bias$^1$ is smaller than *bias* for several parameters decrease. $F^1$ is bias$^1$ of 3x3 filters with one convolutional layer stride. In contrast to that, from the spatial filter selection experience, the convolutional layer learns the better filter automatically for extracting the features of spectral-spatial used for classification. Figure.5 shows the output feature maps of the convolutional layer of every plant type learned with 1000 units.

### 3.2.2 Pooling Layer

The computational complexity is reduced by the pooling layer through convolutional layer features generalization. The feature map shrinks through the pooling operation and however, it resulted in more abstract and robust. The proposed deep CNN pooling layer is defined as,

$$e^2 = g(\mathrm{down}(e^1)) \quad (2)$$

From Eq. (2), the max-pooling function denotes as down(.), the ReLU activation function is g(.) and the pooling layer's output feature maps are shown ase$^2$. During the training process, from the convolutional layer, the function of max pooling slides with two over whole features maps' stride. The following Figure 6 shows the operation of max pooling with two strides. Pooling shows more advantages to CNN and in specific pooling focuses on preventing the overfitting problem by focusing on pooling window local data by minimizing the data dimensionality.

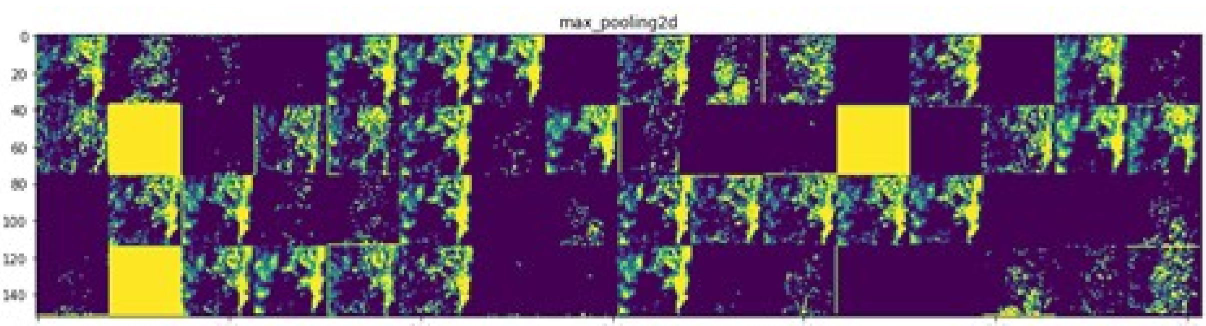


**Figure 6:** *Max-pooling layer- feature maps of one plant image (sample).*

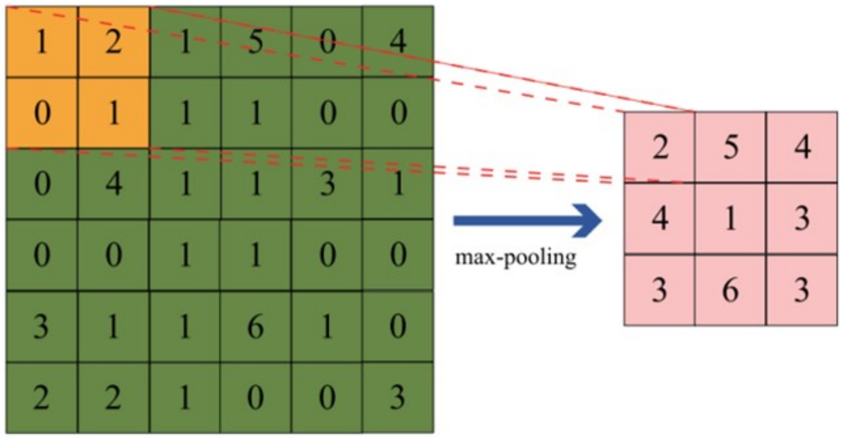


**Figure 7:** *Max pooling operation procedure.*

The calculations are cut down by the reduction in data dimensionality. Pooling exhibited invariance comprised of scaling, translation, and rotation since scalings or certain dislocations makes no variations after accurate pooling. The maximum outcomes are associated with non-overlapping max-pooling of every pooling region and compressed in terms of height and width over the input image. Figure 7 shows the output feature maps of the max-pooling layer of every plant type learned with 1000 units.

### 3.2.3 Fully Connected Layer

The feature maps are compressed and passed to three fully connected layers after the pooling layer. In conventional neural networks, the fully connected layers are utilized to extract more abstract and deep features for classification. The fully connected layers are denoted as,

$$\mathrm{e}^{\mathrm{l}+1} = \mathrm{g}(\mathrm{F}^{\mathrm{l}}\mathrm{e}^{\mathrm{l}} + \mathrm{bias}^{\mathrm{l}}), \mathrm{l} = 2, 3, 4 \tag{3}$$

From Eq. (3), fully connected layer l input is $\mathrm{e}^{l}$, $\mathrm{F}^{l}$, and $\mathrm{bias}^{l}$ are considered as fully connected layer weights and bias correspondingly. ReLU activation function is g(.).$\mathrm{e}^{l+1}$is considered as fully connected layer output. Deep CNNs is ended up with a fully connected layer in almost all cases. This layer is extending the feature extraction and classification ideas in artificial intelligence-based approaches. For achieving the spatial transformation, full inputs connection with neurons are directed to mixed and re-weight all high order features.

## 3.3 Classification using Hyperparameter Tuned Random Forest:

Generally, a decision tree establishes models which are the same as the actual tree. The proposed algorithm comprised the selected data into smaller subsets and added the tree branches simultaneously. The result is a tree comprised of decision nodes and leaf nodes. The leaf nodes hold the result values based on piper plant extraction i.e. target value and the decision node contains two or more branches determining the every feature value to be tested like piper plant types. Several classifier decision trees remove the single decision tree failure risks which rightly predict the target value. The random forest makes the result be around several trees for exhibiting the outcomes. The random forest margin function is expressed in Eq. (4), Eq. (5) shows the generalization error and Eq. (7) shows the prediction confidence. Here $e_1(x), e_2(x), ..., e_k(x)$ are the decision trees or ensemble of classifiers and from vectors, X, Y, the training data is obtained.

The random forest margin function is,

$$\mathrm{mg(A,B) = av_k I(e_k(A) = B) - max_j \neq Bav_k I(e_k(A) = j)} \tag{4}$$

In Eq. (4), I(.) is the indicator function.

Generalization error is,

$$\mathrm{PRE^* = PR_{A,B}(mg(A,B) < 0)} \tag{5}$$

From Eq. (5), over A and space Y the probability is determined. In the random forests, $e_k(A) = e(A, \Theta_k)$, hence for all trees sequences, the number of decision trees or classifiers increases. The probability PRE* meets Eq.(6) using the tree structure and strong law of large numbers,

$$\mathrm{PR_{A,B}(PR_\Theta(e(A,\Theta) = B) - max_{j \neq Y} R_\Theta(e(A,\Theta) = j)} \tag{6}$$

The corrective mechanism is denoted as $(x_1, y_1)$ ,...,$(x_m, y_m)$ ,where $x_i \in A, y_i \in B = \{-1, +1\}$. For t = 1,...,T, initialize $D_I(i) = 1/m$ . After a weak learner training, random forest case is using the distribution $\mathrm{D}_t$.

Get the hypothesis, $e_t : A \rightarrow \{-1,+1\}$

With the error $e_t = PR_{r_{i\sim D_t}}[e_t \ x_i \ \neq y_i]$

After choosing $\alpha_t = \frac{1}{2}\ln\ln\left(\frac{1-e_t}{e_t}\right)$

Update $D_{t+1}(i) = \frac{D_t(i)}{z_t} \ A\{e^{-\alpha t} \text{ if } e_t \ x_i \ = y_i e^{\alpha t} \text{ if } e_t \ x_i \ \neq y_i$

$= \frac{D_t \exp(-\alpha_t y_i e_t(x_i))}{Z_t}$

Normalization factor is $Z_t$ and final hypothesis is obtained as,

$H(x) = \mathrm{sign}(\Sigma_{t=1}^{T} \alpha_t e_t(x))$ (7)

The dependent variable is class whereas the independent variables are plant images. For the proposed optimal performance, a grid search over selected parameters grid is conducted to gain a set of better-performing parameters. The following Table 1 is shown the hyper-parameters returned using the grid search approach for all plants and the piper plants dataset utilized in the study.

***Table 1:*** Optimal hyper parameters

| | |
|---|---|
| **For All plants** | |
| Parameters | Best Parameter |
| max_depth : [80, 90, 100, 110] | 80 |
| max_features: [2, 3] | 3 |
| min_samples_leaf: [3, 4, 5] | 3 |
| min_samples_split: [8, 10, 12] | 8 |
| n_estimators: [100, 200, 300, 600] | 300 |
| **For piper plants** | |
| Parameters | Best Parameter |
| max_depth : [80, 90, 100, 110] | 100 |
| max_features: [2, 3,5] | 5 |
| min_samples_leaf: [2,3, 4, 5] | 2 |
| min_samples_split: [2,4,6,8, 10, 12] | 2 |
| n_estimators: [100, 200, 300, 600] | 500 |

Thus random forest is one type of classifier which is strong enough to outliers and noise due to the randomness it generates. Two randomness types are provided, the first is in terms of data and the next is in terms of features. The proposed hyperparameter tuned random forest algorithm is shown below,

**Hyperparameter Tuned Random Forest:** Algorithm

*Input: NumT = Number of Trees,NumD = Training Data,TF = Total Features,sf = Subset of Features , Tuning Parameters using Grid Search CV*
*Output: Plant class label for the input data*

1. *Set the Hyperparameters and Initialize the values*
2. *For each tree in Forest NumT and Params:*
   a) *Select a bootstrap sample S of size NumD from training data.*
   b) *Create the tree TFb by recursively repeating the following steps for each internal node of the tree.*
      (1) *Choose f at random from the sf.*
      (2) *Select the best among sf.*
      (3) *Split the node*
      (4) *Perform the Grid Search Operation and tune every Parameter*
3. *Once NumT Trees are created,a Test instance will be passed to each tree and the class label will be assigned based on the majority of votes based on H(x).*

As the random forest is the decision trees combination it handles several numbers of hyperparameters and they are,

- Number of trees for generating the decision forest
- Number of features for random selection
- Every tree's depth

All the mentioned hyperparameters are needed to be manually set which can be time-consuming and doesn't assure better results for the manual set parameter. Every hyperparameter possesses its significance and impact on the prediction of output. An initial hyperparameter is the number of trees in a random forest, the accuracy of the model increases linearly with the number of trees. If the size of the forest is larger the accuracy is better, however, the accuracy is not modified at a specific level even though there exists an increase in the number of trees. In the classification process, the number of trees plays a major role. The random forest does not perform on all features but as a replacement, there are two feature values and it can provide better accuracy related to other feature values it deserves to be trying random forest with selecting random features selection of other values. In a random forest, tree depth is considered a highly critical hyperparameter and if smaller values are selected then the model will lead to underfitting.

# 4 Result and Discussion

The following section illustrates the results of piper plant classification using hyperparameter tuned Random forest classifier based on Deep CNN feature extraction.

## 4.1 Dataset description

The dataset used in the study are all plant-type leaves and piper plant leaves types. The normal plant type's dataset is shown in Table 2 and the specific piper plant and its type dataset are shown in Table 3. The dataset shown in the Table 2 were fetched and scrapped from sources that include Google images and Kaggle resources. Since the dataset employed are used for the scientific piper plants, it data were not fetched from the sub-classes. Common images were collected from the Kaggle leaf dataset. Dataset associated for the corn were collected from https://www.kaggle.com/datasets/smaranjitghose/corn-or-maize-leaf-disease-dataset. Dataset for the remaining plants were collected from, https://www.kaggle.com/datasets/abdallahalidev/plant village-dataset. For the normal plant dataset, the plant classes considered are piper plant, apple, peach, strawberry, and others with an image count of a total of 1607.

***Table 2:*** Normal plant dataset

| Class | Plant Classes | Image Count |
|---|---|---|
| 1 | Piper Plant | 172 |
| 2 | Apple Plant | 260 |
| 3 | Peach Plant | 180 |
| 4 | Strawberry Plant | 200 |
| 5 | Cherry Plant | 160 |
| 6 | Grape Plant | 130 |
| 7 | Corn Plant | 151 |
| 8 | Paddy Plant | 150 |
| 9 | Rice Plant | 209 |
| 10 | Soya Plant | 160 |
| 11 | Potato Plant | 152 |
| 12 | Tomato Plant | 132 |

***Table 3:*** Piper plant dataset

| Class | **Piper Classes** | **Image Count** |
|---|---|---|
| 1 | Piper Mullesa | 102 |
| 2 | Piper Nigrum | 100 |
| 3 | Piper Adunucum | 100 |
| 4 | Piper Argyrites | 101 |
| 5 | Piper umbellatum | 103 |
| 6 | Piper Excelsum | 107 |
| 7 | Piper Parmatum | 103 |
| 8 | Piper Orantum | 102 |
| 9 | Piper porphyrophyllum | 102 |
| 10 | Piper sylvaticum | 102 |
| 11 | Piper longum | 100 |
| 12 | Piper auritum | 102 |
| 13 | Pepper Bell | 180 |
| 14 | Piper sarmentosum | 103 |
| 15 | Piper betel | 100 |

Further for the piper plant dataset, several types of piper plants are considered such as Piper Mullesa, Piper Nigrum, Piper Adunucum, Piper Argyrites, and others with an image count of 2056.

Correlation among the features of both all plants and piper plants dataset shows crucial information about features and degree of impact over target value. The Pearson correlation heat map among both the dataset features is shown below Figure 8 shows a relatively stronger positive correlation among the plant's class type whether the plant is related to apple or tomato or piper or others and if it is piper then it is related to Piper Mullesa or Piper Nigrum or others.

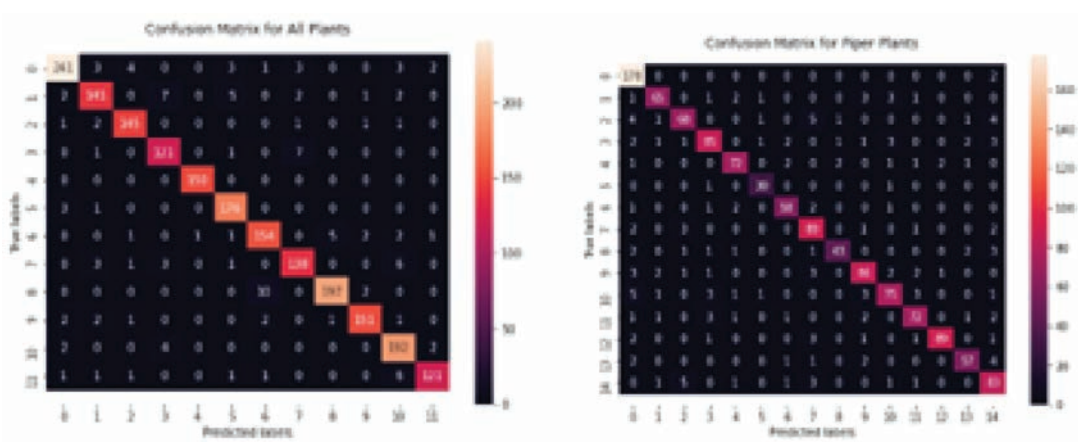


**Figure 8:** *Confusion matrix for all plants and piper plants.*

## 4.2 Performance analysis

The following section depicted the proposed Hyperparameter tuned Random forest classification of all plants and piper plants to performance metrics such as accuracy, precision, recall, and F1-score.

### 4.2.1 For All Plants Classification Results

The following Table 4 shows the performance analysis of all plant types classification based on every plant leaf for precision, recall, and F1score. The paddy plant classification shows the better result with 0.99 as precision and 1 value for recall and f1 score, followed by rice plant with 0.97 as precision and 0.94 and 0.96 as recall and f1-score respectively compared with other plant types.

***Table 4:*** All Plant Performance Analysis

| S.No | Plant Types | Image count | Precision | Recall | F1 Score |
|---|---|---|---|---|---|
| 1 | Piper Plant | 172 | 0.92 | 0.92 | 0.92 |
| 2 | Apple Plant | 260 | 0.96 | 0.93 | 0.94 |
| 3 | Peach Plant | 180 | 0.94 | 0.98 | 0.96 |
| 4 | Strawberry Plant | 200 | 0.9 | 0.96 | 0.93 |
| 5 | Cherry Plant | 160 | 0.96 | 0.94 | 0.95 |
| 6 | Grape Plant | 130 | 0.9 | 0.93 | 0.91 |
| 7 | Corn Plant | 151 | 0.95 | 0.96 | 0.95 |
| 8 | Paddy Plant | 150 | 0.99 | 1 | 1 |
| 9 | Rice Plant | 209 | 0.97 | 0.94 | 0.96 |
| 10 | Soya Plant | 160 | 0.92 | 0.88 | 0.9 |
| 11 | Potato Plant | 152 | 0.91 | 0.91 | 0.91 |
| 12 | Tomato Plant | 132 | 0.96 | 0.92 | 0.94 |

The overall performance of the proposed model with all plant types is shown below in Table 5 and Figure 9. The accuracy value shows 0.94, precision, recall, and f1-score as 0.94 respectively.

***Table 5:*** Overall Performance

| | |
|---|---|
| Accuracy | 0.94 |
| Precision | 0.94 |
| Recall | 0.94 |
| F1 -Score | 0.94 |
| TNR | 0.99449 |
| TPR | 0.93917 |

***Comparative Analysis***

The proposed algorithm hyperparameter tuned random forest classifier compared with other models like NB, SVM, and LR models for all plants type classification shown in below Figure.10.


**Figure 9:** *Performance analysis of all plants types classification.*

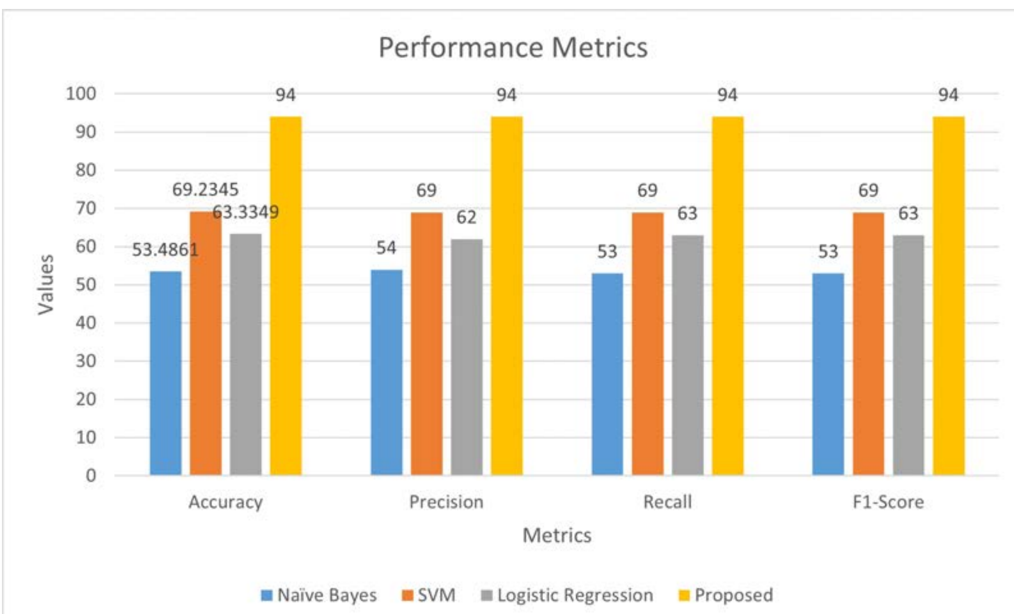


**Figure 10:** *Comparative analysis of proposed algorithm with other machine learning approaches.*

Figure 10 shows that the proposed algorithm shows 94% accuracy, recall, precision, and f1-score compared with SVM at nearly 69% followed by NB at 53% and LR at 63% respectively. Hence the proposed hyper-parameter tuned random forest classifier for all plants type classification shows better results.

***Table 6:*** Test analysis of all plants type

| Test Analysis | Range | K Values |
|---|---|---|
| Landis & Koch | Almost Perfect | 0.93 |
| Fleiss | Excellent | 0.95 |
| Cicchetti | Excellent | 0.96 |
| Cramer | Very Strong | 0.89 |
| Matthews | Very Strong | 1 |
| Scott PI | Perfect Agreement. | 0.93379 |

Table 6 shows the Mathews co-efficient test analysis shows a very strong range of 1 –k value compared with other test analysis co-efficient as Landis & Koch as 0.93, Fleiss as 0.95, Cicchetti as 0.96, Cramer co-efficient as 0.89 and Scott PI as 0.93 value respectively.

***Table 7:*** Piper plant performance analysis

| Class | Piper Classes | Plant count | Precision | Recall | F1 - Score |
|---|---|---|---|---|---|
| 1 | Piper Mullesa | 102 | 0.88 | 0.83 | 0.85 |
| 2 | Piper Nigrum | 100 | 0.9 | 0.84 | 0.87 |
| 3 | Piper Adunucum | 100 | 0.86 | 0.8 | 0.83 |
| 4 | Piper Argyrites | 101 | 0.88 | 0.83 | 0.85 |
| 5 | Piper umbellatum | 103 | 0.9 | 0.88 | 0.89 |
| 6 | Piper Excelsum | 107 | 0.88 | 0.94 | 0.91 |
| 7 | Piper Parmatum | 103 | 0.89 | 0.89 | 0.89 |
| 8 | Piper Orantum | 102 | 0.83 | 0.91 | 0.87 |
| 9 | Piper porphyrophyllum | 102 | 0.91 | 0.8 | 0.85 |
| 10 | Piper sylvaticum | 102 | 0.87 | 0.85 | 0.86 |
| 11 | Piper longum | 100 | 0.86 | 0.81 | 0.83 |
| 12 | Piper auritum | 102 | 0.88 | 0.86 | 0.87 |
| 13 | Piper Bell | 180 | 0.87 | 0.99 | 0.93 |
| 14 | Piper sarmentosum | 103 | 0.89 | 0.85 | 0.87 |
| 15 | Piper betel | 100 | 0.79 | 0.86 | 0.83 |

### 4.2.2 For piper plants classification results

The following Table 7 shows the performance analysis of piper plant types classification based on every piper plant leaf to precision, recall, and f1score. The Piper porphyrophylum plant classification shows the better result as 0.91 precision, 0.8, and 0.85 as recall and f1-score values respectively. Followed by the piper nigrum and piper Umbellatum plant classification with 0.9 as precision and 0.84, 0.88(piper nigrum, piper Umbellatum) and 0.87, 0.89 (piper nigrum, piper Umbellatum) value for recall and f1 score respectively compared with other plant types.

The overall performance of the proposed model with piper plant types is shown in Table 8 and Figure 11. The accuracy, precision, and recall values show 0.88 and f1-score as 0.87 respectively.

***Table 8:*** Overall Performance

| | |
|---|---|
| Accuracy | 0.88 |
| Precision | 0.88 |
| Recall | 0.88 |
| F1 -Score | 0.87 |
| TNR | 0.99097 |
| TPR | 0.86782 |


**Figure 11:** *Performance analysis of piper plants types classification.*

***Comparative Analysis***

The proposed algorithm hyperparameter tuned random forest classifier compared with other models like NB, SVM, and LR models for piper plants type classification shown in below Figure.12.

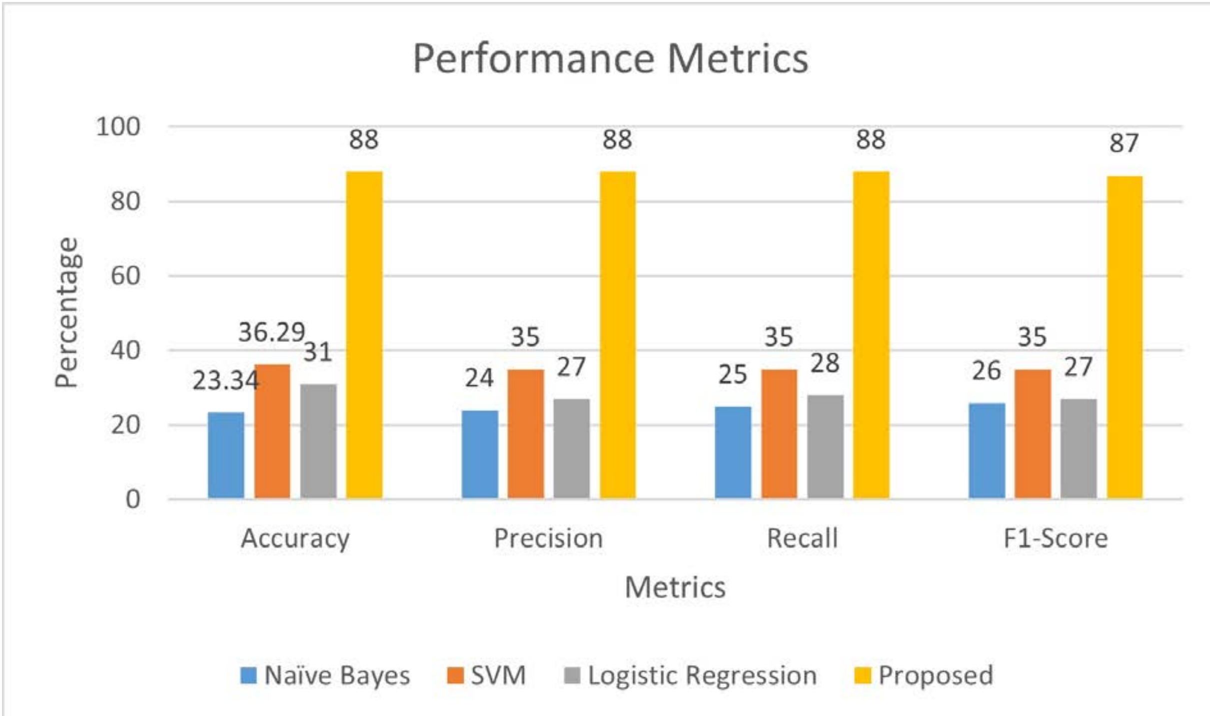


**Figure 12:** *Comparative analysis of proposed algorithm with other machine learning approaches.*

Figure 12 shows that the proposed algorithm shows 88% accuracy, recall, precision, and f1-score compared with SVM at nearly 36.29% followed by NB at 23.34% and LR at 31% respectively. Hence the proposed hyper-parameter tuned random forest classifier for piper plants type classification shows better results.

***Table 9:*** Test Analysis of All plants type

| Test Analysis | Range | K Values |
|---|---|---|
| Landis & Koch | Almost Perfect | 0.94 |
| Fleiss | Excellent | 0.93 |
| Cicchetti | Excellent | 0.94 |
| Cramer | Very Strong | 0.87 |
| Matthews | Strong | 0.85 |
| Scott PI | Perfect Agreement | 0.86483 |

Table 9, shows the Cramer co-efficient test analysis shows a very strong range of 0.87 –k value compared with other test analysis co-efficient of Landis & Koch as 0.94, Fleiss as 0.93, Cicchetti as 0.94, Mathews co-efficient as 0.85 and Scott PI as 0.86 value respectively.

During the process of feature selection, the parameters selected by certain considered methods have been listed in Table 10.

***Table 10:*** Best parameter selection by hyper-parameter tuned -RF

| Parameters | All Plants Parameter | Piper Plants Parameter |
|---|---|---|
| max_depth : [80, 90, 100, 110] | 80 | 90 |
| max_features: [2, 3] | 3 | 2 |
| min_samples_leaf: [3, 4, 5] | 3 | 4 |
| min_samples_split: [8, 10, 12] | 8 | 8 |
| n_estimators: [100, 200, 300, 600] | 300 | 600 |

Hyper parameter tuned RF method has extracted certain parameters from all plant datasets like max_depth: [80, 90, 100, 110] for with the best parameter value of 80 and n_estimators: [100, 200, 300, 600] with the best parameter value of 300. And from the piper plant dataset, RF method has selected the best parameter as 90 from max_depth parameters and selected 600 as the best parameter from the n_estimators parameter.

Similarly, as shown in Table 11, hyper-parameter tuned NB feature selection algorithm selected certain features from all plant datasets that include sample_weight: [0,1,1,1,5,2] with the best parameter value of 1. And selected parameter var_smoothing-2e-9 w with the best parameter value of 2.00E-09. And selected 0.1 as the best parameter from the sample_weight parameter and 2.00E-09 as the best parameter from var_smoothing-=2e-9.

***Table 11:*** Best parameter selection by hyper-parameter tuned -NB

| Parameters | All Plants Parameter | Piper Plants Parameter |
|---|---|---|
| sample_weight : [0.1,1,1.5,2] | 1 | 0.1 |
| var_smoothing=2e-9 | 2.00E-09 | 2.00E-09 |

The hyper-parameter tuned-SVM selected certain parameters from the all plant dataset that include the best parameters as rbf from the kernel and 0.01 as the best parameter from Gamma. Similarly, from the piper plant dataset, the SVM algorithm selected 100 as the best parameter from the C parameter and 0.001 as the best parameter from the gamma parameter (see Table 12).

***Table 12:*** Best parameter selection by hyper-parameter tuned -SVM

| Parameters | All Plants Parameter | Piper Plants Parameter |
|---|---|---|
| kernel : [‘linear’, ‘poly’, ‘rbf’, ‘sigmoid’, ‘precomputed’] | rbf | rbf |
| C': [0.1, 1, 10, 100, 1000] | 10 | 100 |
| gamma': [1, 0.1, 0.01, 0.001, 0.0001] | 0.01 | 0.001 |

*Table 13:* Best parameter selection by hyper-parameter tuned -LR

| Parameters | All Plants Parameter | Piper Plants Parameter |
|---|---|---|
| solvers = ['newton-cg', 'lbfgs', 'liblinear'] | lbfgs | lbfgs |
| penalty = ['l2'] | l2 | l2 |
| c_values = [100, 10, 1.0, 0.1, 0.01] | 10 | 10 |

From Table 13, the hyper-parameter tuned LR selected lbfgs as the best parameter from the solver's parameter and selected l2 as the best parameter from the penalty parameter. Similarly, from the piper plant dataset, LR selected l2 as the best parameter from the penalty parameter and selected 10 as the best parameter from c_values parameter.

Comparative analysis to assure the effectiveness of the techniques of the present study with the all plants dataset was attained. In relevance to this, a comparative Table 14 and its diagrammatical representation has been depicted below.

*Table 14:* Hyper Parameter Tuned Result for All plants

| Algorithm | Accuracy | Precision | Recall | F1-Score |
|---|---|---|---|---|
| NB | 91.25 | 90.54 | 90.21 | 90.34 |
| SVM | 85.62 | 84.25 | 84.17 | 84.63 |
| LR | 87.25 | 86.93 | 86.28 | 86.64 |
| **Random Forest** | **94** | **94** | **94** | **94** |

By inferring Table 14 and Figure 13, it is clear that RF has better results when compared with the other methods such as SVM, LR, and NB. The comparison was made on the basis of certain metrics like precision, accuracy, recall and F1-score. Random forest yielded the value of 94 for all the considered metrics which is obviously greater than the other methods. These better results are due to the tuning operations performed with the deep hidden layers, stride adjustments, kernel adjustments and hyper-parameter tunings done in ML models with the corresponding parameters. This has assisted in obtaining better results.

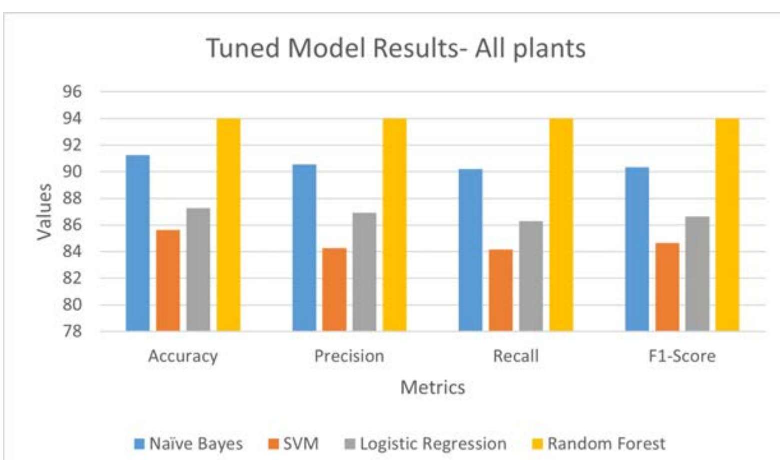


**Figure 13:** *Comparison of Tuned model results of- All plant dataset.*

A comparative analysis was made on the dataset of the piper plant in order to evaluate the effectiveness of the present study. For a clear depiction, Table 15 and its graphical representation have been illustrated below (see Figure 14)

***Table 15:*** Hyper Parameter Tuned Result for Piper plants

| Algorithm | Accuracy | Precision | Recall | F1-Score |
|---|---|---|---|---|
| Naïve Bayes | 85.76 | 84.36 | 84.13 | 84.3 |
| SVM | 84.25 | 83.96 | 83.54 | 83.61 |
| Logistic Regression | 82.65 | 81.59 | 81.23 | 81.45 |
| **Random Forest** | **88** | **88** | **88** | **87** |

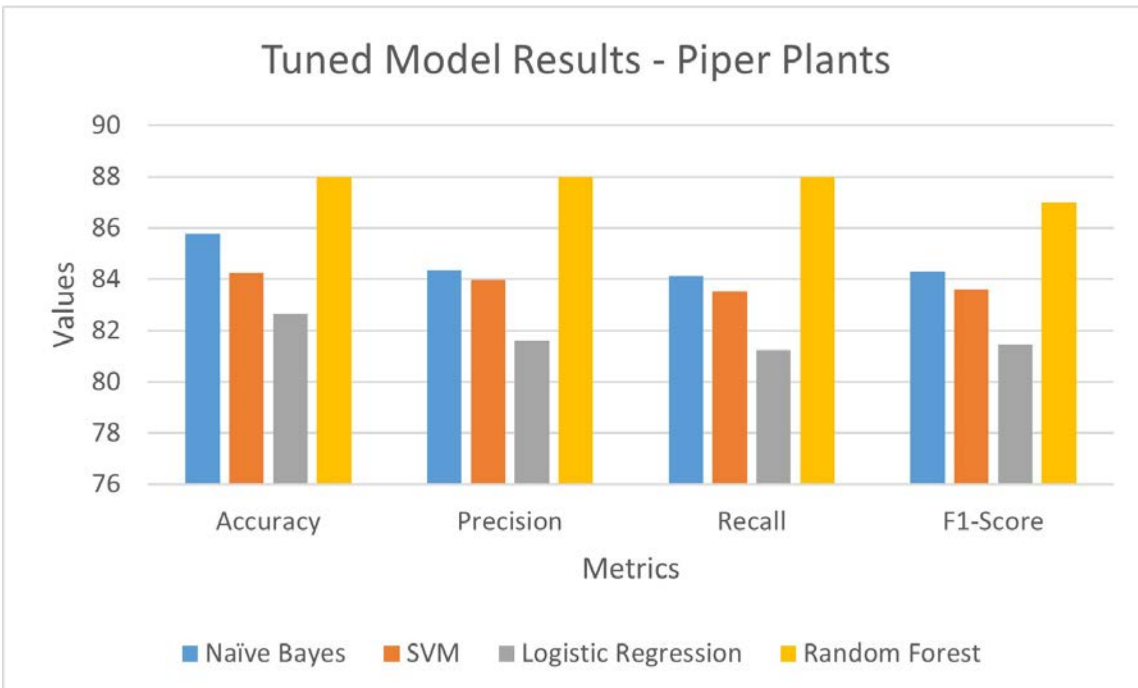


**Figure 14:** *Comparison of Tuned model results of Piper plant dataset.*

From the above Table 15 and Figure 14, it is obvious that the selected algorithm RF has the optimal results when compared with the other existing methods (SVM, LR and NB). The comparison was performed on the basis of certain metrics like precision, accuracy, recall and F1-score. RF method has secured 88 as the value for precision, accuracy and recall and secured the value 87 for F1-score.

ReLU function is used to avoid issues like vanishing gradient during the process of feature extraction. Also an attempted to employ softmax. Yet, the result of this function was not potential enough when compared with ReLU.

Similarly, a comparative analysis based on the activation function has been attempted for both all plants and the piper plant dataset in order to prove the efficiency of the present study. To accomplish this, Table 15 represents the outcomes, and its graphical representation has been illustrated.

Table 16 has listed the results of the comparative analysis of the considered activation functions.

*Table 16:* Softmax vs Relu

| Activation Function | Piper Plant | All Plant |
|---|---|---|
| Relu | 94 | 88 |
| Softmax | 90.32 | 81.52 |

From Table 16 and graph (Figure 15), it is more obvious that the Relu function employed by the present study has higher accuracy with the value of 94 for all plants dataset and 88 as the accuracy value for the piper plant dataset when compared with the softmax activation function.

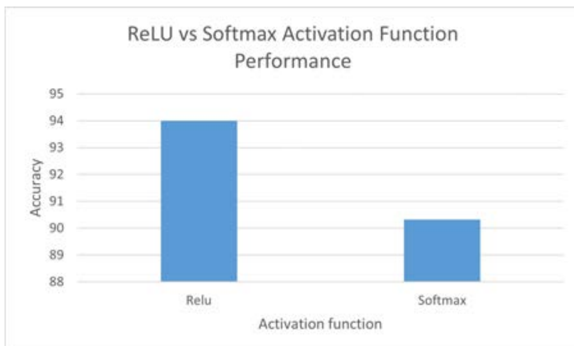


**Figure 15:** *Comparative results of the activation functions.*

## 4.3 Screenshots

The screenshots in the section enumerated the classification of appropriate plant species through the proposed framework. The image is chosen from the dataset, on the GUI screen. The interface outputs in the model show the Classification process through the Hyper-parameter Tuned Random-forest method. The method implies if the plant belongs to a non-piper plant or belongs to a piper plant classes were determined in the outcomes.


**Figure 16:** *Deep CNN feature extraction and HPT Random Forest Classification.*

Figure.16 depicts the identification of non-piper plants from the uploaded sample image. The sample image was selected from the present all plants dataset. The input images undergo the Deep CNN method for feature extraction and HPT Random-forest method for classification, to generate the classified results. The categorized plant name is depicted in the above figure, with a prediction accuracy rate of 70.80 percent. Apple plant is determined as the classified plant name result showing 70.8 % of accuracy in prediction.


**Figure 17:** *Deep CNN feature extraction and HPT Random Forest Classification.*

The Same task proceeded with another image sample. The image is uploaded after the selection from the non-piper or piper plant dataset. Figure.17 describes the classified plant name with another image sample. After the implementation classification algorithm, the image sample was predicted in belonging to the Piper-plant class category. The Piper betel is the identified outcome in piper plant species, belonging to the Piper Plant class category. The name of the piper plant species is determined with a prediction accuracy rate of 77.40 %.

# 5 Conclusion

This study proposed a hyperparameter tuned random forest classifier for all types of plants and piper plants classification. The deep CNN is utilized for feature extraction purposes for effective extraction of all plants and piper plants separately. Further for piper plants and all types of plants, the classification is performed and the respected plant type is identified. The proposed algorithm shows effective results as 0.94 accuracies for all plants and 0.88 for piper plants respectively. Hence the specific piper plants are classified by using the proposed hyperparameter tuned random forest classifier accurately. In the future, the piper plants classification can be performed by using various other deep learning approaches to improve the accuracy

**Funding:** There is no funding provided to prepare the manuscript.

**Conflicts of Interest:** The authors declare that there is no conflict of interest regarding the publication of this paper.

**Authors Contribution:** All authors have approved the manuscript and agree with its submission.



# References

[1] Alajrami, M. A., & Abu-Naser, S. S. (2020). Type of tomato classification using deep learning. *International Journal of Academic Pedagogical Research (IJAPR),* 3(12).

[2] Alimboyong, C. R., Hernandez, A. A., & Medina, R. P. (2018). Classification of plant seedling images using deep learning. Paper presented at the TENCON 2018-2018 IEEE Region 10 Conference, 1839-1844, IEEE.

[3] Anubha Pearline, S., Sathiesh Kumar, V., & Harini, S. (2019). A study on plant recognition using conventional image processing and deep learning approaches. *Journal of Intelligent & Fuzzy Systems,* 36(3), 1997-2004.

[4] Barré, P., Stöver, B. C., Müller, K. F., & Steinhage, V. (2017). LeafNet: A computer vision system for automatic plant species identification. *Ecological Informatics,* 40, 50-56.

[5] Chaki, J., Parekh, R., & Bhattacharya, S. (2020). Plant leaf classification using multiple descriptors: A hierarchical approach. *Journal of King Saud University-Computer and Information Sciences,* 32(10), 1158-1172.

[6] Chau, A. L., Hernandez, R. R., Mora, V. T., Canales, J. C., Mazahua, L. R., & Lamont, F. G. (2017). Detection of compound leaves for plant identification. *IEEE Latin America Transactions,* 15(11), 2185-2190.

[7] Chaudhury, A., & Barron, J. L. (2018). Plant species identification from occluded leaf images. *IEEE/ACM transactions on computational biology and bioinformatics,* 17(3), 1042-1055.

[8] Chen, J., Zhang, D., Nanehkaran, Y. A., & Li, D. (2020). Detection of rice plant diseases based on deep transfer learning.*Journal of the Science of Food and Agriculture,* 100(7), 3246-3256.

[9] Ferentinos, K. P. (2018). Deep learning models for plant disease detection and diagnosis. *Computers and Electronics in Agriculture,* 145, 311-318.

[10] Ghazi, M. M., Yanikoglu, B., & Aptoula, E. (2017). Plant identification using deep neural networks via optimization of transfer learning parameters. *Neurocomputing,* 235, 228-235.

[11] Goyal, N., & Kumar, N. (2021). On solving leaf classification using linear regression. *Multimedia Tools and Applications,* 80(3), 4533-4551.

[12] Hang, J., Zhang, D., Chen, P., Zhang, J., & Wang, B. (2019). Classification of plant leaf diseases based on improved convolutional neural network. *Sensors,* 19(19), 4161.

[13] Hu, J., Chen, Z., Yang, M., Zhang, R., & Cui, Y. (2018). A multiscale fusion convolutional neural network for plant leaf recognition. *IEEE Signal Processing Letters,* 25(6), 853-857.

[14] Huixian, J. (2020). The analysis of plants image recognition based on deep learning and artificial neural network. *IEEE Access,* 8, 68828-68841.

[15] Kala, J. R., & Viriri, S. (2018). Plant specie classification using sinuosity coefficients of leaves. *Image Analysis & Stereology,* 37(2), 119-126.

[16] Kolivand, H., Fern, B. M., Saba, T., Rahim, M. S. M.,& Rehman, A. (2019). A new leaf venation detection technique for plant species classification. *Arabian Journal for Science and Engineering,* 44(4), 3315-3327.

[17] Kumar, M., Gupta, S., Gao, X.-Z., & Singh, A. (2019). Plant species recognition using morphological features and adaptive boosting methodology. *IEEE Access,* 7, 163912-163918.

[18] Lee, S. H., Chan, C. S., Mayo, S. J., & Remagnino, P. (2017). How deep learning extracts and learns leaf features for plant classification. *Pattern Recognition,* 71, 1-13.

[19] Mostajer Kheirkhah, F., & Asghari, H. (2019). Plant leaf classification using GIST texture features. *IET Computer Vision,* 13(4), 369-375.

[20] Pravin, A., & Deepa, C. (2021). A Identification of Piper Plant Species Based on Deep Learning Networks. *Turkish Journal of Computer and Mathematics Education* (TURCOMAT), 12(10), 6740–6749.

[21] Raj, A. P. S. S., & Vajravelu, S. K. (2019). DDLA: dual deep learning architecture for classification of plant species. *IET Image Processing,* 13(12), 2176-2182.

[22] Rajesh, K., & Bhaskari, D. L. (2021). Optimum Dataset Size for Ayurvedic Plant Leaf Recognition Using Convolution Neural Networks. Data Management, Analytics and Innovation, (pp. 249-261): Springer, Singapore.

[23] Selvam, L., & Kavitha, P. (2020). Classification of ladies finger plant leaf using deep learning. *Journal of Ambient Intelligence and Humanized Computing,* 1-9.

[24] Sun, Y., Liu, Y., Wang, G., & Zhang, H. (2017). Deep learning for plant identification in natural environment. *Computational intelligence and neuroscience,* 2017.

[25] Wäldchen, J., Rzanny, M., Seeland, M., & Mäder, P. (2018). Automated plant species identification—Trends and future directions. *PLoS computational biology,* 14(4), e1005993.

[26] Wang, B., & Wang, D. (2019). Plant leaves classification: A few-shot learning method based on siamese network. *IEEE Access,* 7, 151754-151763.

[27] Wei Tan, J., Chang, S.-W., Abdul-Kareem, S., Yap, H. J., & Yong, K.-T. (2018). Deep learning for plant species classification using leaf vein morphometric. *IEEE/ACM transactions on computational biology and bioinformatics,* 17(1), 82-90.

[28] Yang, K., Zhong, W., & Li, F. (2020). Leaf segmentation and classification with a complicated background using deep learning. *Agronomy,* 10(11), 1721.

[29] Zhang, S., Wang, H., & Huang, W. (2017). Two-stage plant species recognition by local mean clustering and Weighted sparse representation classification.*Cluster Computing,* 20(2), 1517-1525.

[30] Zhang, S., Zhang, C., Wang, Z., & Kong, W. (2018). Combining sparse representation and singular value decomposition for plant recognition. *Applied Soft Computing,* 67, 164-171.

## About the Authors

**A. Pravin** was born in Nagercoil, Kanyakumari of Tamil Nadu is a graduate from Bishop Ambrose College, Coimbatore in the year (2010). He completed his post-graduation degree in Master of Science in Information Technology and Master of Philosophy in Computer Science from Sri Ramakrishna College of Arts and Science (Formerly SNR Sons College), Coimbatore. He started his professional career in 2012 as an Assistant Professor in Bishop Ambrose College, Coimbatore. Then in 2015 he worked as a teacher in Computer Science at Belfield Matric Hr Sec School, Nagercoil. He worked Ruben College of Arts and Science, Thadikkarankonam as Head of the Department in Computer Science till 2019 and A.Pravin Joined Elim Meditech Pvt Ltd as Software Programmer. His areas of interest are Networking, Data Mining and Web Technology.

**Dr. C. Deepa** born in Elappully, Palakkad, India is a graduate in Bachelor of Science in Physics from University of Calicut, Kerala and has completed her Post-graduation degree in Master of Computer Application and Master of Philosophy in Computer Science and Ph.D in Computer Science from Bharathiar University, Coimbatore, Tamilnadu, India. At present Dr. C Deepa, is Head, Department of Computer Science (Artificial Intelligence & Data Science), Sri Ramakrishna College of Arts and Science, (Formerly SNR SONs College) Coimbatore, Tamilnadu, India. She has got 20+ years of experience in teaching and research and is a research supervisor for both M.Phil and Ph.D. Her specific research interests include Data Mining featuring Web Content Mining, Network and Communication and Image Processing. She had authored a book and has published over 20 refereed articles in International Journals and Conferences. She is a member for Board of Studies and reviewer for International Journals. Formerly she was the Head of the Department for Computer Science and Application at Nehru College of Arts and Science, Tamilnadu, India where she has organized and headed many activities for the Department and College. Dr C Deepa was an invited speaker for WGC-2018 held at Singapore and is a recipient of Best Women Researcher Award from IOSRD and I2OR, Best Research Paper Award from SNR SONs College, Best Paper Presentation award in ICACCS-2013.