Ayurvedic Medicinal Plant Identification System using Embedded Image Processing Techniques

Arnab Das[1], B Siva Sai Kumar[1], S Shiva Shankar Reddy[1], S Naveen Reddy[1], Peeyush K P[1]

[1] Department of Electronics & Communication Engineering,
Amrita School of Engineering, Coimbatore,
Amrita Vishwa Vidyapeetham, India.

Abstract: Medicinal plants seem to be a rich resource for drug development and have a huge impact in ayurvedic medicine preparation. Plants can be classified based on various classification methods such as cell, genetic and serum etc. It's difficult for an individual to explore the various classification methods and it's practically not feasible as it demands good knowledge in plant taxonomy and long-term time investment. Due to the shortage of experienced and qualified taxonomists in identification and classification of medicinal plants, With the help of different image processing algorithms and computer vision, the above difference can be bridged. The main objective is to develop a Deep Learning and ML based model to identify and classify plants based on various features, which is done with thehelp of Gabor filter and to develop a standalone device that clicks a picture and identifies themedicinal plant. And made comparative accuracy analysis between different classifiers and feature extractors.

Keywords: feature extraction, segmentation, classification, CNN, image processing, machine learning, Random forest classifier, Gabor filter, LGBM, GLCM, Shannon entropy, medicinal plants

1 Introductory

Plants are vital to the survival for life & biodiversity on Earth by allowing the air, water flow for all existence. The most significant classes of the flora are medicinal herbs that are used to treat various diseases. The knowledge of medicinal plants that has been passed down through the years ought to be conserved as well as guarded. Vision systems, Specific pattern classification, different methods for imagery analysisare a promise to identifying and then categorizing medicinal uses of flora. One of the most difficult jobs is identifying a medical plant with the needed therapeutic properties. Even if herbal therapy has no negative effects, a patient's life might be lost if a medicinal plant is misidentified. As a result, at this point in time, a completely automated method to accurately recognize medicinal plants is inevitable.

For production of ayurvedic medications, the identification and categorization of medicinal plants is critical. Appropriate classification of medicinal plants benefits horticulturists, botanists, ayurvedic medical therapists, forest service authorities, as well as those engaged in the making of ayurvedic pharmaceuticals. However, a critical shortage of skilled taxonomists exists in this field. There is a growing segment of the population who prefers ayurvedic medicine over other medications. Taxonomists employ leaf, flower, trunk, and branch characteristics to classify based on their respective features. The leaves are the greatest choice for plant categorization.

2 Literature Survey

Deep learning relates to neural network structure, according to the findings of the research [1], it can learn characteristics from huge data automatically and train and categorize plant leaf samples through the use of a neural network model using a back propagation method. Leaf form, color, and texture are the most important factors in plant categorization. Leaf texture veins have more consistent properties than color and form.

A plant leaf identification system based on morphological characteristics and adaptive boosting methods is provided in this study [2]. They employed k-NN, decision trees, and multilayer perceptrons, among other categorization algorithms. The accuracy rate has been improved with adaboost approach compared to conventional approach.

Ayurleaf [3], a CNN-based methodology for classifying medicinal plants, has been presented. Neural network with deep learning based on Alexnet is used for the employment to extract the features of the flora. To classify the data, SVM classifiers & SoftMax were utilized.

The author compared well-known models VGG16 and VGG19 and presented a complete analysis of deep learning-based Convolution Neural Networks. They created [4] their own dataset, which includes 64 medicinal plant species and 1000 samples from each.

For recognizing contoured borders of overlapped leaves from a complicated backdrop, the working procedures outlined in this study [8] are used. The proposed approach divides the primary picture into leaf regions by segmenting the specified contour areas. The model attained a segmentation rate of 90.46 percent overall, with segmentation rates of 95.34 percent for single leaves and 86.73 percent for overlapping leaves, respectively, according to the experimental data.

The CNN [9] is a form of neural network that is extensively employed in image processing. Convolution, Max-pool, and fully linked layers combine to form CNN. The author uses CNN to construct a hardware device that can distinguish between plastic and non-plastic with a 97.8% accuracy.

According to CNN, a machine [10] has been created that can determine if a plant leaf is healthy or unhealthy. For processing, the author employed a Linux-based machine with a CUDA-enabled GPU, Darknet for image processing and recognition, and Python to interface the program with the darknet.

The key processes in identifying the leaf are feature extraction and classification. The author [11] utilized histogram and texture to extract features and a Support Vector Machine to classify them. They employed five distinct classes, each including four different leaf sizes. Following the categorization, the output forecasts which class the leaf corresponds to.

The database is the most significant element of image processing in general. The author [12] analyses three distinct methods of feature extractions, including Wavelet transformations, Gray Level Co-occurrence matrix (GLCM), and Gray Level Difference method (GLDM), using varied training and validation percentages. The classifier utilized is KNN. Finally, they came to the conclusion that Wavelet transformations had the highest accuracy.

3 Methodology

Feature extraction is the most important aspect of classification/identification. Image processing, picture feature extraction, and image classification are the three phases that make up the recognition process.

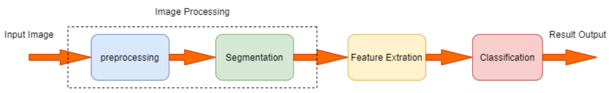


Fig. 1. Classification Process

Image preprocessing and segmentation techniques are the two key processes in the image recognition process. Image restoration and transformation are two of the most used image preprocessing techniques. Its major goal is to reduce picture interference and noise, increase image usable information, and improve object detection. The purpose of segmentation is to divide a recognized image into subregion.

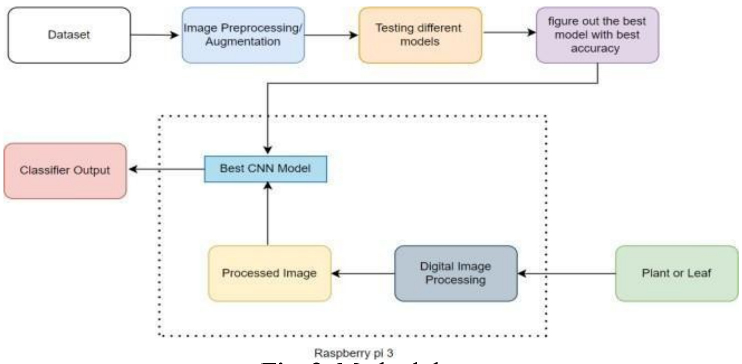


Fig. 2. Methodology

3.1 Data Source

When it comes to object categorization, the dataset is crucial. We have collected a segmented leaves dataset from Kaggle which consists of leaves of 30 classes of which each class has an average of 60 images of size 1600x1200.
Data augmentation is a strategy for extending the quantity of data available by generating new data from current data. It allows us to improve the dataset's size and variety without having to acquire fresh information.
The following data enhancements were performed: Rotation of the images, Horizontal flip. For data augmentation, Gaussian noise is injected into the picture. The process of blurring a picture entail averaging nearby pixels. This blurs the image and lowers the amount of information. The dataset size grows larger, and the model's training improves. To enhance the precision of a deep learning-based model, we applied a data augmentation technique. The training and testing dataset are randomly split from the dataset. The train set is liable for 80% of the data source, whilst the testing set is liable for 20%.

3.2 Deep Learning Model (CNN)

A CNN sequential model is implemented by considering VGG16 and Alexnet as reference. The input layer is the initial layer, and it defines the size of the input pictures. The convolution layer uses 32 (3 x 3) filters with a stride size of one. The output is then thresholded by a ReLU layer, which is accompanied by the max – pooling layer by a 2x2 filter. The previous layer precisely halves the exact size of its return. A secondary convolution layer follows this max pooling layer in which a 32 kernel is performed with a dimension of 3x3. ReLU layered will be followed, accompanied with the layer of max-pool with 2x2 of size of the filter by a stride valueof 1. Now the next two layers are back-to-back convolution layers with following configurations. Both of them employ 3 x 3 kernels by a stride value of 1, with 64 & 250 kernels in each. Following these layers, a ReLU layer is added. The next layer is a convolution layer having 128 (3x3) kernels, after that a ReLU activation, an averagingpool layer by a value of (2,2) size. The next levels follow a similar pattern, with 64 (3x3) filters, after that a ReLU activation function, then averaging pool layers by a value of (2,2) size. Last Convolution layer is of size 256 (2x2) and followed by activation of the ReLU function and max pool layers with a value of (2x2) size. Final value of the last max-pool layers is transferred to fully-connected layers which have 6400-neurons. And it is again processed through another fully connected layer of size 30 as our dataset consists of 30 classes. Final classification of the model is done with the help of SoftMax which is used for multi-classification purposes.

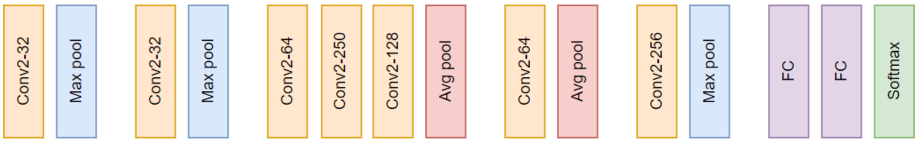


Fig. 3. Implemented CNN model

3.3 Machine Learning

Model Gabor Filter Bank

Gabor is just a convolutional filter that comprises a gaussian and sinusoidal term together. The weights are given by the gaussian component, whereas the directionality is given by the sine component.

Gabor is used to provide texture and edge features. Gabor kernel actually mimics the visual cortex meaning, the way we recognize textures with our eyes can be simulated using Gabor kernel.

$$g(x, y; \lambda, \theta, \psi, \sigma, \gamma) = \exp\left(-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}\right) \exp\left(i\left(2\pi\frac{x'}{\lambda} + \psi\right)\right)$$

λ – Wavelength of the sine component
Θ - Orientation of filter
Ψ- Phase offset
σ- Std. dev. Of the gaussian envelop
γ – Spatial aspect ratio

Where,
$X' = x\cos\theta + y\sin\theta$ and $Y' = -x\sin\theta + y\cos\theta$

For example, if we change the parameters like lambda, theta, phase offset of the standard deviation of the gaussian, we can generate an ideally infinite number of filters.

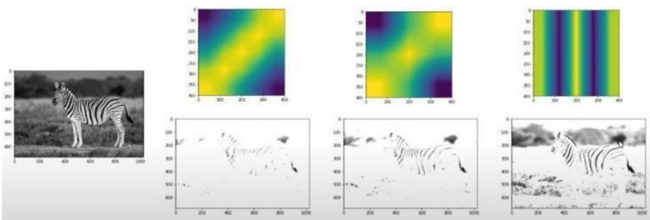


Fig. 4. Different orientations of Gabor filter

In the above figure we can observe that by changing the parameters of the filter it generates filters with various orientations through which we can extract features from the original image. Usually one filter is not enough, so we generate nearly 12 Gabor filters, convoluted with the image and then fed to our ML model.

GLCM – Gray Level Co-occurrence Matrices

Features like aspect ratio, roundness, perimeter, area, length, breadth [2] give better results, but in most of the cases, these features are not enough or efficient enough to classify the leaves. There are chances that 2 different leaves might have the same length and breadth. But to an extent it's sure that the texture of 2 different kinds of leaves will be different. The textures determine the patterns that appear repeatedly and are regarded as local changes in picture intensity. In this paper, a statistical approach like the use of GLCM is being suggested [11]. GLCM uses second order statistics for features that can be used to infer the degree of correlation between pairs of pixels.

Uses pairs of pixels where the user can define the distance and angle between the pixels. It is recommended to extract GLCM for multiple distances and angles between pixels. A GLCM matrix will be formed for the computation of

- Contrast
- Dissimilarity
- Homogeneity
- ASM
- Energy
- Correlation

Random Forest Classifier (RFC)

Random Forest Classifiers and SVMs are the mostly used traditional ML models. As the name says, Random Forest means it has a number of decision trees. Random Forest Classifier randomly picks the attributes or features that are generated previously, the Gabor filter banks among a selection or sub selection of all the attributes or features. RFC uses Gini impurity to pick a node to split that yields maximum information gain. This is used to take the decision of the tree.

A Random Forest is made up of a huge set of discrete trees that work together as a unit. Each tree offers a forecast, and the model's prediction is the group with the most votes.

We have chosen this classifier since it helps to minimize overfitting (works great on training data but not on new data). Bootstrap / bagging in Random Forest allows each tree to randomly sample from the dataset. Feature randomness is introduced by only allowing a subset of available features at each node.

In our model we have used 40 n_estimators (decision trees) and the image size which is given to our model is 256x256 which is resized from our original image which is of size 1600x1200. We have used all the features i.e., Gabor filter banks, original pixel values and Sobel filter to train our model. We have also used bootstrapping of samples technique to minimize overfitting.

LGBM – Light Gradient Boosting Machine

In this paper, we have also suggested the use of Light BGM as a classifier which works great with the GLCM feature extractor. It’s a fast, distributed, high-performance gradient boosting framework based on a decision tree algorithm. It splits the tree leaf wise, unlike other models do it tree wise or level wise.

3.4 Image Segmentation

Because leaves are most typically observed in groups with natural backgrounds, leaf identification and segmentation is a challenging task. Because of their similar colors, the edges of leaves are difficult to see from the photograph. It's also more difficult to separate each individual leaf, especially overlapping ones, because leaves are almost identical in color, texture, and shape.
Leaf segmentation entails two steps: (1) separating the main leaf region from the natural backdrop, and (2) extracting individual leaves and covered or overlapped (i.e., objects on object) leaves from the picture. A contour selection-based leaf segmentation technique is used for segmentation.

Input Image
The source pic was taken in blue, green, red colors form instead of Red, Green, Blue since we used the Open cv to generate the model.

Preserve Edges
We must first determine the contours or outlines of each leaf before segmenting the connected regions inside them, as our goal is to identify the contour or edges of each leaf. Internal texture is smoothed and boundary edges are preserved using the Edge Preserving Filter at this step.

Stylize Leaf Boundary
To make every item's edge, shape, or contour sleek and crisp at the same time, we employ a Stylization Filter. In comparison to traditional outline sharpening filters, it uses the Normalized Conv (NC) filter to increase accuracy and speed. Internal texture has no effect on it as it hardens all boundary edges over time.

Multiple-Channel to Solitary-Channel
The data in BGR format (3 channels) is then processed to a gray level (1 channel).In comparison to BGR images, only one third of the picture data has to be processed in grayscale images, which greatly minimizes the amount of computation and memory usage.

Expand Boundary Edges
This process enlarges the outline borders of leaves by expanding the areas of bright regions. This processing phase also aids in the separation of overlapping edges. This is done using a Sobel filter.

Smooth Edges
When the dilatation procedure expands the bright borders, this may enlarge remaining portions of the leaf's internal texture edges. To solve the issue, we perform a smoothing operation after dilatation.

Separate Background and Foreground
We use a single intensity to differentiate the pixels in the foreground first from pixels in the backdrop.

Contour Detection
By recognizing the contours of the connected regions, we are able to identify them. The OpenCV Libraries findContours() method is used to detect contours.

4 Results

4.1 Deep Learning (CNN) Results

For the implemented CNN based model which is constructed taking VGG16 and Alexnet as reference consists of 15 layers with various functions. Validation and training accuracy of the model are around 77% and 78% respectively.

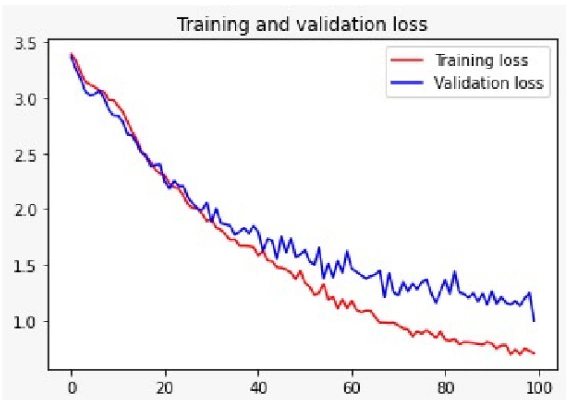


Fig. 5. Loss in testing & training.

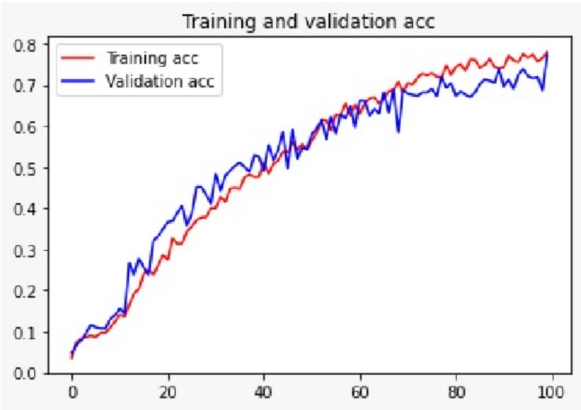


Fig.6. Accuracy in testing & training

We can see in the graphs of training & testing accuracy as well as the loss of implemented sequential models. When an image is provided as a predictions input to a classifier, we have achieved an accuracy of 75%.

4.2 Machine Learning Model Results

For the RFC based ML model due to memory constraints we have tested our model with only 8 classes which consist of data from following classes.

S.no	Class
1	Basella Alba (Basale)
2	Carissa Carandas (Karanda)
3	Ficus Religiosa (Peepal Tree)
4	Jasminum (Jasmine)
5	Mangifera Indica (Mango)
6	Mentha (Mint)
7	Moringa Oleifera (Drumstick)
8	Psidium Guajava (Guava)

TABLE I. Opted classes for model implementation

In this paper we have tried to compare different feature extractors and different classifiers with accuracy as our major metric. In the testing phase around 25 images of each leaf class were used to calculate the accuracy of our model. As said above, 8 leaf classes are used to compare different models and feature extractors. Gabor filter bank with Random Forest classifier achieves the highest accuracy of 96%. Next LBGM classifier with GLCM and Shannon entropy as feature extractor were used and achieved an accuracy of 87.5%. Random forest classifier with GLCM and Shannon entropy extractors achieves an accuracy of 86.5% max. Table II gives the details of accuracy after changing various parameters of theclassifiers and feature extractors.

Classifier	Gabor, pixel and Sobel	GLCM & Shannon entropy	Accuracy	
RF	YES		**n_estimator= 40**	
			Max feature	**Accuracy**
			0.2	0.96
			0.3	0.96
			0.5	0.96
			0.6	0.955
		YES		
			Max_feature	**Accuracy**
			0.3	0.934
			0.4	0.923
			0.8	0.934
			n_estimator= 50	
			Maxfeature	**Accuracy**
			0.3	0.945
			0.4	0.923
			0.8	0.934
			n_estimator= 70	
			Max_feature	**Accuracy**
			0.3	0.956
			0.4	0.934
			0.9	0.923
			n_estimator= 90	
			Max_feature	**Accuracy**
			0.3	0.945
			0.4	0.934
			0.6	0.923
			n_estimator= 110	
			Max_feature	**Accuracy**
			0.6	0.945
			0.8	0.934
			0.9	0.934
			n_estimator= 140	
			Max_feature	**Accuracy**
			0.3	0.945
			0.4	0.934
			0.6	0.945
LGBM		YES	**lr**	**Accuracy**
			0.04	0.913
			0.06	0.934
			0.08	0.945
			0.2	0.956
			0.4	0.956

TABLE II. Accuracy comparison of different Classifiers & featureextractors

We have used Gabor filter banks to extract the features of leaves which is the most widely used filter and which depends on various parameters like theta, sigma, lambda and gamma etc. It generates filters of various orientations which makes it easier for models to train.

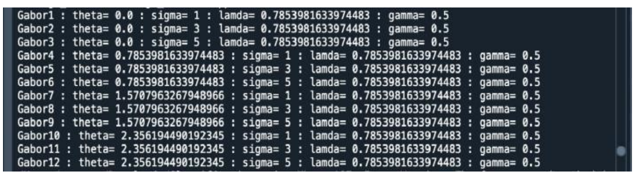


Fig. 8. Gabor filter parameters

The accuracy of each class we trained is shown below respectively and overall accuracy is around 96.5%.

Class	Class Name	Accuracy
1	Basella Alba (Basale)	0.96
2	Carissa Carandas (Karanda)	0.92
3	Ficus Religiosa (Peepal Tree)	0.96
4	Jasminum (Jasmine)	0.96
5	Mangifera Indica (Mango)	1.0
6	Mentha (Mint)	0.96
7	Moringa Oleifera (Drumstick)	0.96
8	Psidium Guajava (Guava)	1.0
Overall Accuracy		0.965

Table III. implemented classes and model accuracy

Confusion matrix is plotted below which describes the validation accuracy of each class from Table I.

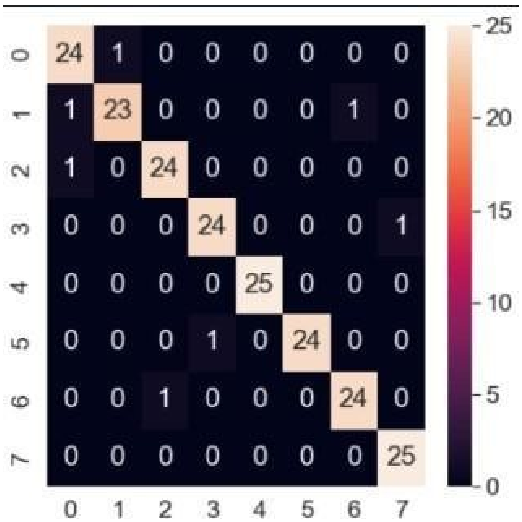
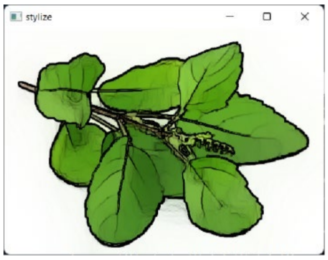
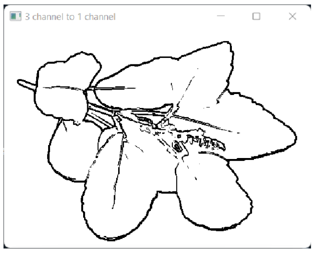
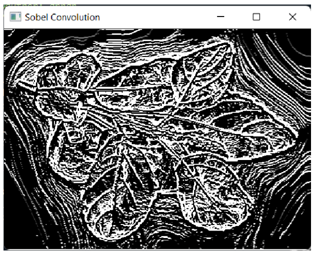


Fig. 9. Confusion matrix of implemented model

4.3 Segmentation Results

When an image consisting of multiple leaf images of the same class is given as input our task is to segregate or find contours from the image and give it as an input to implemented models to classify it.

Stage	Output
Input image	
Edge-Preserving Filter	
Stylization Filter	
Multi-Channel to Single Channel conversion	
Sobel Filter	

Smoothing Edges	
Contour Detection	

TABLE IV. Segmentation Results

5 References

1. Jiang Huixian, “The Analysis of Plants Image Recognition Based on Deep Learning and Artificial Neural Network”, IEEE(Data Mining for Internet of Things),DOI:10.1109/ACCESS.2020.2986946,13 April 2020

2. Munish Kumar, Surbhi Gupta, Xiao-Zhi Gao, AmitojSingh,“Plant Species Recognition Using Morphological Features and Adaptive Boosting Methodology”, IEEE Access (Volume: 7), DOI: 10.1109/ACCESS.2019.2952176,07 November 2019.

3. M.R. Dileep, P.N. Pournami, “AyurLeaf: A Deep Learning Approach for Classification of Medicinal Plants”, TENCON 2019 - 2019 IEEE Region 10 Conference(TENCON),Kochi,India,DOI: 10.1109/TENCON.2019.8929394, 17-20 Oct. 2019.

4. Anu Paulson, S Ravishankar, “AI Based Indigenous Medicinal Plant Identification”, 2020 Advanced Computing and Communication Technologies for High Performance Applications (ACCTHPA), Cochin, India, DOI:10.1109/ACCTHPA49271.2020.9213224, 2-4 July 2020.

5. LiwenGao, XiaohuaLin, “Fully automatic segmentation method for medicinal plant leaf images in complex background”, ELSEVIER, Computers and Electronics in Agriculture Volume 164, September 2019,104924

6. MadsDyrmann, HenrikKarstoft, Henrik SkovMidtiby, “Plant species classification using deep convolutional neural network”, ELSEVIER, Biosystems Engineering Volume 151, November 2016, Pages 72-80.

7. C. Amuthalingeswaran, M. Sivakumar, P. Renuga, S. Alexpandi, J. Elamathi, S. Santhana Hari, “Identification of Medicinal Plant's and Their Usage by Using Deep Learning”, 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI),Tirunelveli,India,DOI: 10.1109/ICOEI.2019.8862765, 23-25 April 2019.

8. Rafflesia Khan, Rameswar Debnath, “ Segmentation of Single and Overlapping Leaves by extracting Appropriate Contours”,DOI: 10.5121/csit.2019.91323.

9. Tarun, K., K. Sreelakshmi, and K. P. Peeyush. 2019. "Segregation of Plastic and Non-Plastic Waste using Convolutional Neural Network.".

doi:10.1088/1757-899X/561/1/012113.

10. P. S. A. Kesava and K. P. Peeyush, "Autonomous Robot to Detect Diseased Leaves in Plants using Convolutional Neural Networks," 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI), 2019, pp. 806-809, doi: 10.1109/ICOEI.2019.8862737.
11. D. Venkataraman and N. Mangayarkarasi, "Support vector machine based classification of medicinal plants using leaf features," 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI), 2017, pp. 793-798, doi: 10.1109/ICACCI.2017.8125939.
12. B. R. Pushpa, N. Megha and K. B. Amaljith, "Comparision and Classification of Medicinal Plant Leaf Based on Texture Feature," 2020 International Conference for Emerging Technology (INCET), 2020, pp. 1-5, doi: 10.1109/INCET49848.2020.9154155.