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 are rich resource for drug development and has 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 shortage of experienced and qualified taxonomists in identification and classification of medicinal plants, this gap can be bridged with the help of various image processing techniques and computer vision. 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 the help of Gabor filter and to develop a standalone device that clicks a picture and identifies the medicinal plant.

Keywords: feature extraction, segmentation, classification, CNN, image processing, machine learning, rain forest classifier, Gabor filter

1 Introduction

Plants play an essential role in the preservation of life and biodiversity on Earth by allowing the flow of air and water for all living things. One of the most significant classes of plants is medicinal herbs, that are used to treat various diseases. The knowledge of medicinal plants that has been passed down through the years must be conserved and protected. Computer vision, pattern classification, and image analysis technologies are a promise for identifying and categorizing medicinal plants. 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. A deep neural network based on Alexnet is employed for feature extraction. 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.

3 Methodology

The major part of the classification/identification deals with feature extraction. Image recognition process consist of three stages mainly (i) image processing (ii) feature extraction (iii) classification of image.

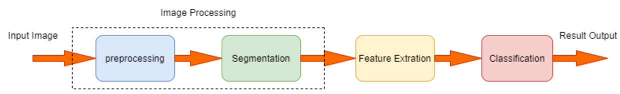


Fig. 1. Classification Process

Image recognition is sub divided into image preprocessing and image segmentation. Image preprocessing main purpose is to remove the noise and enhance the required information using various filters for example gaussian filter is used to reduce the noise in the image but it introduces the blur based on the requirement we should opt the filter and different techniques. Image segmentation deals with extracting the necessary sub-regions from the image which is done using (i) edge-based segmentation (ii) region-based segmentation and (iii) threshold-based segmentation.

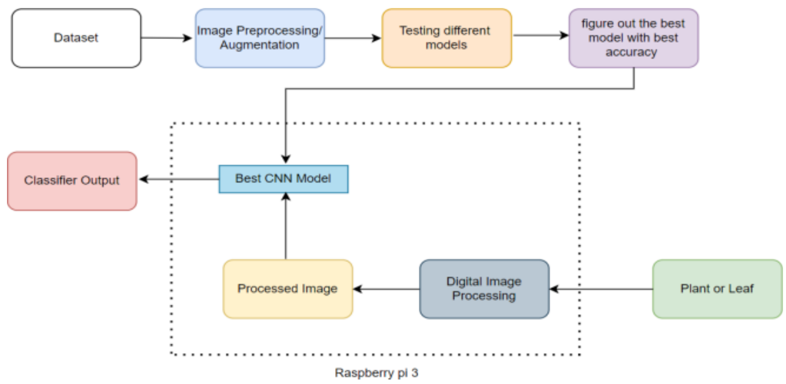


Fig. 2. Methodology

3.1 Dataset

Dataset plays a major role in classification of objects. 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 accounts for 80% of dataset, whereas the testing set accounts 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) filter with such a stride size of one. The output is then thresholded by a ReLU layer, which is accompanied by a max - pooling with a 2x2 filter size. This layer precisely halves the exact size of its output. A secondary convolution layer follows this max pooling layer, which performs on 32 kernels with a 3x3 dimension. A ReLU layer follows, accompanied by a max - pooling with a filter size of 2x2 and a stride of 1. Next two layers are two convolution layers that are back-to-back with following configurations. Both employ 3x3 kernel with such a stride of 1, with 64 and 250 kernels in each. Following these layers, a ReLU layer is added. The next layer is a convolution layer having 128 (3x3) kernels, followed by a ReLU activation and an averaging pooling layer with size (2,2). The next levels follow a similar pattern, with 64 (3x3) filters and a ReLU activation function, then an averaging pooling layer with size (2,2). Last Convolution layer is of size 256 (2x2) and followed by ReLU activation function and max pooling layer of size (2x2). The output of last max pooling layer is sent to the 6400-neuron first fully connected layer. And which 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.

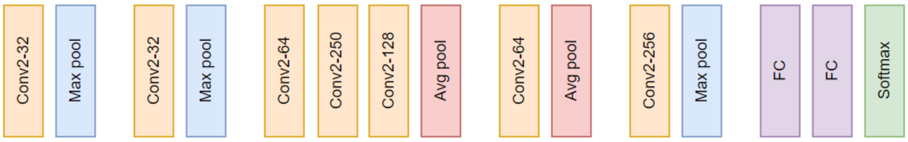


Fig. 3. Implemented CNN model

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.

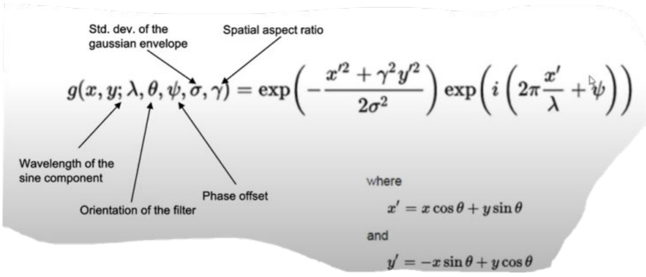


Fig. 4. Gabor filter expression & parameters

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.

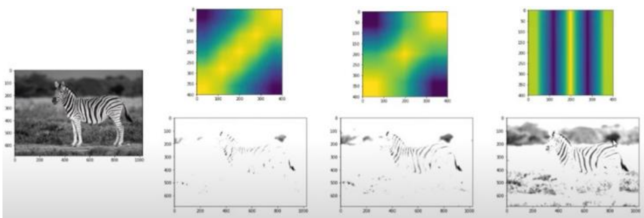


Fig. 5. Different orientations of Gabor filter

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 30 Gabor filters, convoluted with the image and then fed to our ML model.

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 200 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.

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

Because we utilize the OpenCV library to create the model, the input picture is taken in BGR color format rather than RGB.

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.

Multi-Channel to Single-Channel
The next step is to convert the BGR format (3 channel) image to a grayscale image (1 channel). In comparison to BGR images, we only need to process one-third of the image data 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.

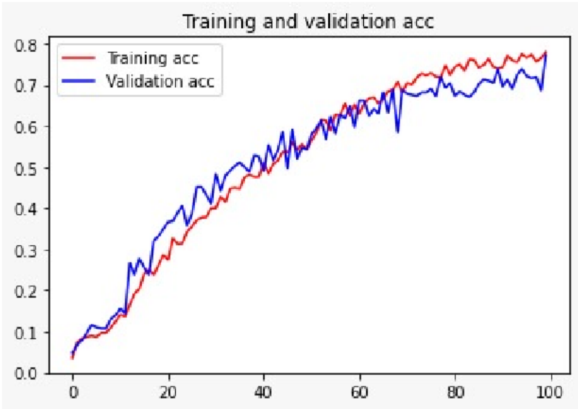


Fig. 6. Training & validation accuracy

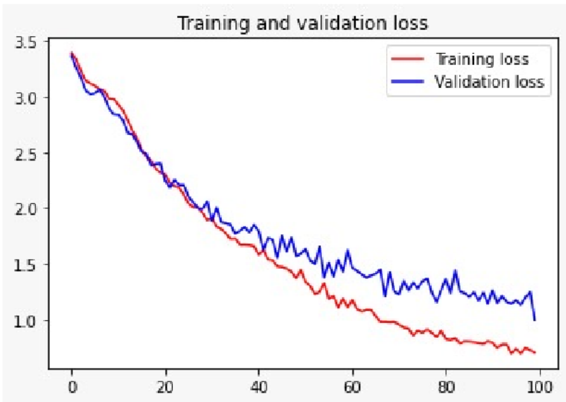


Fig.7. Training & Validation loss

Above we can see the graphs of training and validation accuracy, loss of the implemented sequential model. When an image is given as input to a model to predict we have achieved an accuracy of 75%.

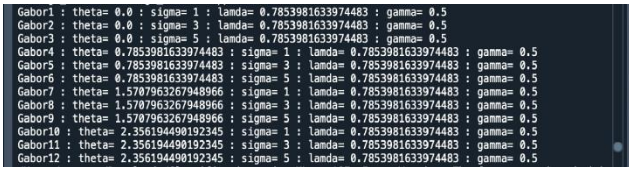
4.2 Machine Learning Model Results

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

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.



```
Gabor1 : theta= 0.0 : sigma= 1 : lamda= 0.7853981633974483 : gamma= 0.5
Gabor2 : theta= 0.0 : sigma= 3 : lamda= 0.7853981633974483 : gamma= 0.5
Gabor3 : theta= 0.0 : sigma= 5 : lamda= 0.7853981633974483 : gamma= 0.5
Gabor4 : theta= 0.7853981633974483 : sigma= 1 : lamda= 0.7853981633974483 : gamma= 0.5
Gabor5 : theta= 0.7853981633974483 : sigma= 3 : lamda= 0.7853981633974483 : gamma= 0.5
Gabor6 : theta= 0.7853981633974483 : sigma= 5 : lamda= 0.7853981633974483 : gamma= 0.5
Gabor7 : theta= 1.5707963267948966 : sigma= 1 : lamda= 0.7853981633974483 : gamma= 0.5
Gabor8 : theta= 1.5707963267948966 : sigma= 3 : lamda= 0.7853981633974483 : gamma= 0.5
Gabor9 : theta= 1.5707963267948966 : sigma= 5 : lamda= 0.7853981633974483 : gamma= 0.5
Gabor10 : theta= 2.356194490192345 : sigma= 1 : lamda= 0.7853981633974483 : gamma= 0.5
Gabor11 : theta= 2.356194490192345 : sigma= 3 : lamda= 0.7853981633974483 : gamma= 0.5
Gabor12 : theta= 2.356194490192345 : sigma= 5 : lamda= 0.7853981633974483 : gamma= 0.5
```

Fig. 9. 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 II. implemented classes and model accuracy

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

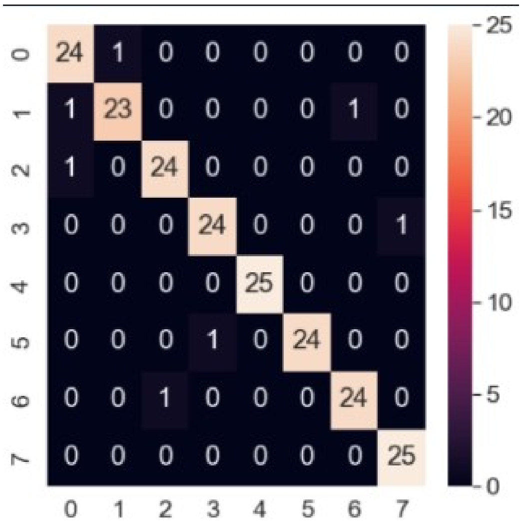
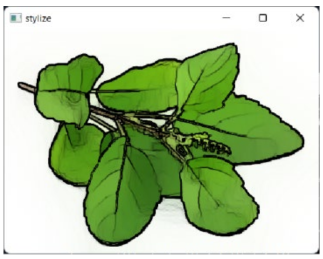


Fig. 10. Confusion matrix of implemented model

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	

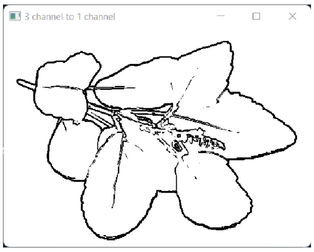
Multi-Channel to Single Channel conversion	
Sobel Filter	
Smoothing Edges	
Contour Detection	

TABLE III. 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.