AI Based Indigenous Medicinal Plant Identification

Anu Paulson
Department of Computer Science and Engineering
Vidya Academy of Science and Technology
Thrissur, India
anupaulsonp@gmail.com

Ravishankar S
Department of Computer Science and Engineering
Vidya Academy of Science and Technology
Thrissur, India
ravishankar.s@vidyaacademy.ac.in

***Abstract*—In preserving the physical and psychological state of persons, ayurvedic medicines have an important role. The research aims to identify indigenous ayurvedic medicinal plant species using deep learning techniques. The social relevance of the proposal is so high as it would solve the problems of a wide range of stakeholders like physicians, pharmacy, government, and public. The identification of rare plant species may lead to a significant impact on the research associated with medical and other related areas. Another application can be the identification of plant species in forest and remote areas, where access to humans is limited. In such cases, the image of a particular plant species may be captured using drones and further analyzed. Currently, a lot of research work has been going on in the area of plant species identification using machine learning algorithms. The performance of Convolutional Neural Network (CNN), and pretrained models VGG16, and VGG19 has been compared for leaf identification problem. The dataset proposed in this research work contains indigenous medicinal plants of Kerala. The dataset consists of leaf images of 64 medicinal plants. CNN obtained a classification accuracy of 95.79%. VGG16 and VGG19 achieve an accuracy of 97.8% and 97.6% respectively, outperforms basic CNN.**

***Index Terms*—Deep Learning, Ayurveda, Medicinal Plants, Plant Classification, Dataset, Convolutional Neural Network, Transfer Learning, VGGNet, VGG16, VGG19**

I. Introduction

Ayurveda is an ancient medical science that works hand in hand with the medicinal plants. There is a huge problem that most people cannot recognize these medicinal plants and thus are not able to take advantage of herbal power to cure diseases. Experts are needed to recognize ayurvedic medicines and sometimes they're also in dilemma about the species of plants. There are various advancements in technology. Deep Learning and Machine Learning helps machines to do the tasks which humans do but with much ease. There are several Machine Learning applications available that recognize humans, objects, trees, animals, etc. But the necessity for a recognizer is increasing in the field of Ayurveda. Because the experts are very less and the use is increasing day by day. Thus we propose a system that recognizes ayurvedic plants just by a single image of its leaf. This can have a large impact on laymen in day to day use, within the field of education and also among researchers. Also, it will unite everyone to take this field of Ayurveda ahead.

In India, there is a wide range of people using Ayurvedic Medicines. The majority prefer these herbal medicines mainly because there are fewer or no side effects of these medicines. Ayurveda cannot sustain without Medicinal plants. Many medicinal plants have been decreased in the last few years and we need an easy and quick way to recognize these medicinal plants. But the biggest problem is the difficulty in finding ingredients (medicinal plants). Also, the identification of a simple medicinal plant requires an expert to be present around. Plants can be classified with the help of shapes, colors, texture, and structure of their leaf, bark flowers seeding and morph. But it is very difficult to recognize plants with their two-dimension image. Thus leaf has been one of the most important features while studying Plants property. Leaf classification is believed to be the most important step in studying plant diversification. An efficient plant recognition system will help the medical field as well as Botanic researches and Ayurvedic studies. This research proposes an efficient method of classification of ayurvedic plants. This solution will fill the gap between the knowledge possessed by the Ayurvedic practitioner and other people [1]. Plant classification is considered a challenging problem because of the variety and the similarity of plants in nature.

In the past few years, the convolutional neural network (CNN) has obtained huge success in computer vision tasks with remarkable accuracy [2]. This paper aims to analyze the performance of basic CNN, VGG16, and VGG19 for leaf recognition using the proposed dataset of the leaves.

II. Related Works

For large-scale image classification, Karen Simonyan and Andrew Zisserman [3] evaluated very deep convolutional networks (up to 19 weight layers). The representation depth is useful for classification accuracy. The performance on the ILSVRC dataset may be attained employing a conventional ConvNet architecture with considerably increased depth. The major contribution of this paper is a thorough evaluation of networks of increasing depth using an architecture with very small (3x3) convolution filters. It shows that a notable development on the prior-art configurations is often attained by pushing the depth to 16-19 weight layers.

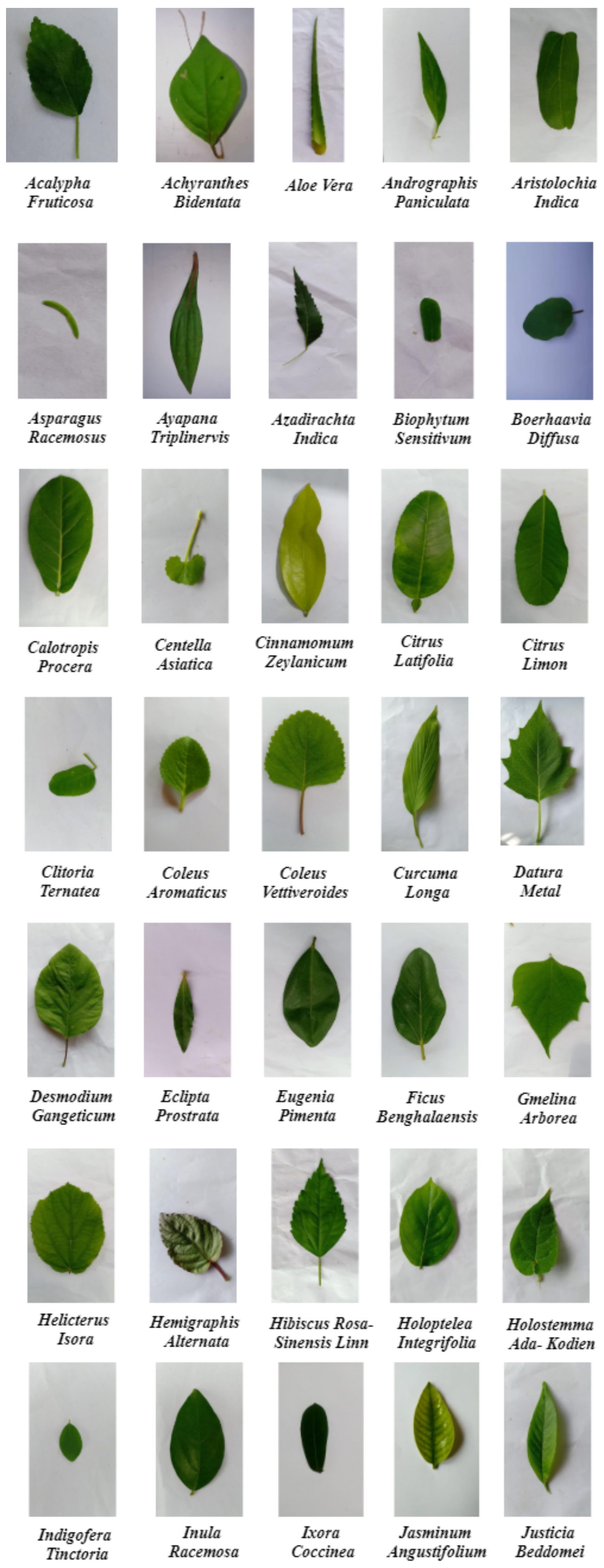
978-1-7281-6453-3/20/$31.00 ©2020 IEEE

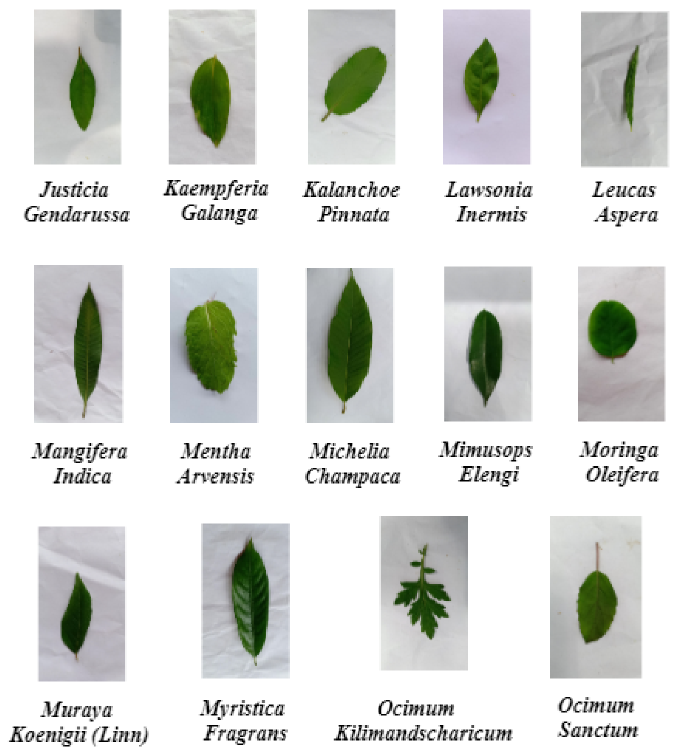
In [5], the performance of basic CNN, AlexNet, and GoogLeNet is compared, concerning the accuracy and elapsed time. Each model has its layers of convolution and computational complexity. GoogLeNet attains a better accuracy rate compared to basic CNN and Alexnet, which are 100%. But, GoogLeNet has the longest processing time because of the number of layers and deeper learning process. GoogLeNet can be utilized for when there is no issue with longer processing time.

In [6], a Convolutional Neural Network (CNN) model called AyurLeaf is proposed to classify medicinal plants. AyurLeaf dataset contains leaf images of 40 medicinal plants. For feature extraction, a deep neural network inspired by Alexnet is utilized. Softmax and SVM classifiers are used to perform classification. With 3-fold and 5-fold cross-validation, the model achieved a accuracy of 96.76%.

III. Data

The proposed dataset is created by capturing images from Vaidyaratnam Ayurveda College, Thaikkattussery. 64 species of medicinal plants with 1000 samples from each medicinal plant species were collected. Thus, the dataset consists of 64000 leaf images. Each plant species has its different leaf characteristics such as elongation, color, shape, and size. The images were acquired using a mobile device. The width and height for all images with the resolution of 96 dpi were 1920 pixels and 1080 pixels respectively. The figures given below show the image samples of 64 medicinal plants.





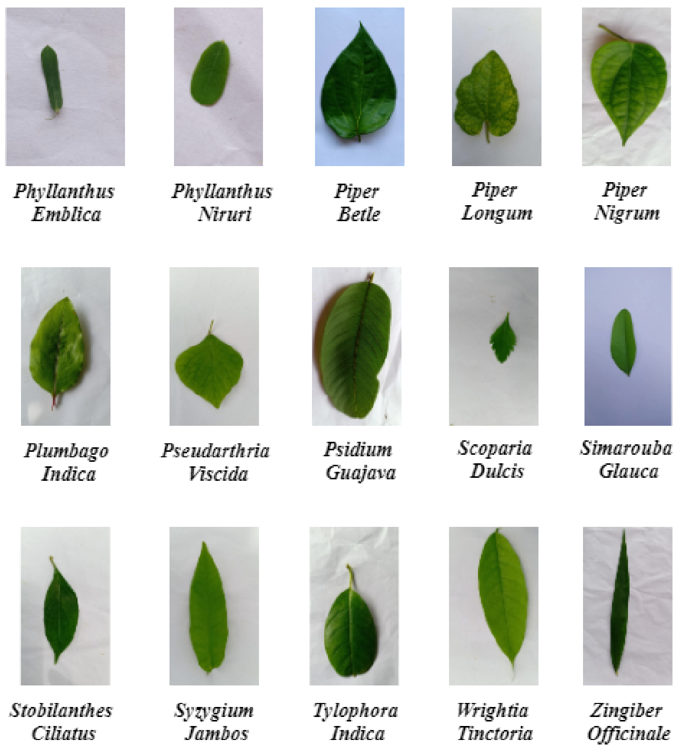


TABLE I
AYURVEDIC MEDICINAL PLANTS AND THEIR SCIENTIFIC NAMES

No:	Name	Scientific Name
1.	Red Flowered Leadwort	Plumbago Indica
2.	Malabarnut	Justicia Beddomei
3.	Indian Elecampane	Inula Racemosa
4.	Cemetery Plant	Hemigraphis Alternata
5.	White Datura	Datura Metal
6.	Willow-Leaf Justicia	Justicia Gendarussa
7.	Indian Acalypha	Acalypha Fruticosa
8.	Indian Long Pepper	Piper Longum
9.	Pala Indigo Plant	Wrightia Tinctoria
10.	Indian Borage	Coleus Aromaticus
11.	Butterfly Pea	Clitoria Ternatea
12.	Persian Lime	Citrus Latifolia
13.	Gigantic Swallow Wort	Calotropis Procera
14.	Champak	Michelia Champaca
15.	Sal Leaved Desmodium	Desmodium Gangeticum
16.	Vettiver	Coleus Vettiveroides
17.	Indian Pennywort	Centella Asiatica
18.	Common Turmeric	Curcuma Longa
19.	Gale of the Wind, Stonebreaker	Phyllanthus Niruri
20.	Banyan Tree	Ficus Benghalaensis
21.	Rose-Apple	Syzygium Jambos
22.	Jungle-Flame Ixora	Ixora Coccinea
23.	Indian Aloe	Aloe Vera
24.	Indian Indigo	Indigofera Tinctoria
25.	Hogweed	Boerhaavia Diffusa
26.	Creat, Kariyat	Andrographis Paniculata
27.	Salaparni	Pseudarthria Viscida
28.	Ginger	Zingiber Officinale
29.	Guava	Psidium Guajava
30.	Thumbe	Leucas Aspera
31.	Indian Elm	Holoptelea Integrifolia
32.	Holostemma	Holostemma Ada- Kodien
33.	Curry Leaf	Muraya Koenigii (Linn)
34.	Emetic Swallow-Wory	Tylophora Indica
35.	Indian Birthwort	Aristolochia Indica
36.	Air Plant	Kalanchoe Pinnata
37.	Allspice	Eugenia Pimenta
38.	Sweet Broom	Scoparia Dulcis
39.	Lesser Kurinji	Stobilanthes Ciliatus
40.	Lemon	Citrus Limon
41.	East Indian Screw Tree	Helicterus Isora
42.	Horse-Radish Tree	Moringa Oleifera
43.	Holy Basil	Ocimum Sanctum
44.	Camphor Basil	Ocimum Kilimandscharicum
45.	Cinnamon	Cinnamomum Zeylanicum
46.	Bullet-Wood Tree	Mimusops Elengi
47.	Coomb Teak	Gmelina Arborea
48.	Emblic Myrobalan	Phyllanthus Emblica
49.	East Indian Galingale	Kaempferia Galanga
50.	Mango Tree	Mangifera Indica
51.	Wild Asparagus	Asparagus Racemosus
52.	Shoe Flower	Hibiscus Rosa Sinensis Linn
53.	Black Pepper	Piper Nigrum
54.	Nutmeg	Myristica Fragrans
55.	Paradise Tree	Simarouba Glauca
56.	Egyptian privet	Lawsonia Inermis
57.	Little Tree Plant	Biophytum Sensitivum
58.	Neem	Azadirachta Indica
59.	Ayapana	Ayapana Triplinervis
60.	Small prickly chaff flower	Achyranthes Bidentata
61.	Wild jasmine	Jasminum Angustifolium
62.	Mint	Mentha Arvensis
63.	Betel	Piper Betle
64.	False Daisy	Eclipta Prostrata

IV. EXPERIMENTAL SETTINGS

Google Colaboratory is used to perform experiments carried out in this work. Colab is a hosted Jupyter notebook service that allows you to write and execute Python in your browser, with zero configuration required, free access to computing resources including GPUs and easy sharing.

Convolutional neural networks (ConvNets or CNNs) can perform image recognition and image classifications. CNN used in the areas of object detection, face recognition, etc. CNN image classification takes an input image, process it, and classify it to different categories. The models used are CNN, VGG16, and VGG19.

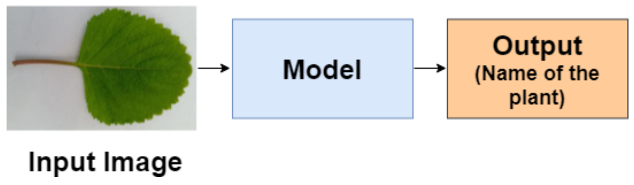


Fig. 1. Level 0

Level 0 gives a basic overview of the complete system or process being analyzed or modeled. In the proposed system input image is fed into the model. Then the name of the plant will be obtained as output, as shown in Fig 1. Deep learning CNN models will be used for training and testing purposes. Each input image passed through a series of convolution layers, filters (Kernals), Pooling layers, fully connected layers

(FC), and will apply Softmax function to classify an object with the probabilistic values between 0 and 1.

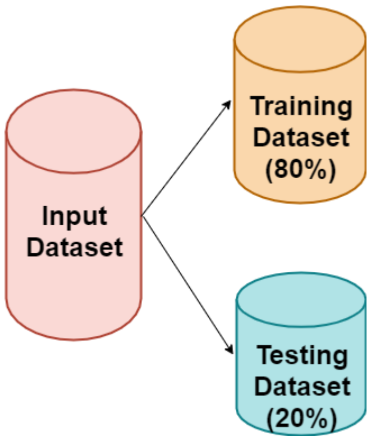


Fig. 2. Level 1

Level 1 depicts basic modules within the system and the flow of data between various modules. Here, the input dataset is split into a training dataset (80%) and a testing dataset (20%), as shown in Fig 2. The model learns on the training dataset. The test dataset used to test the model's prediction and accuracy of the model.

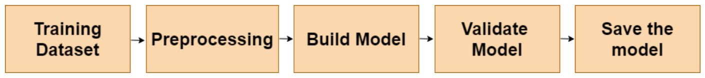


Fig. 3. Level 2

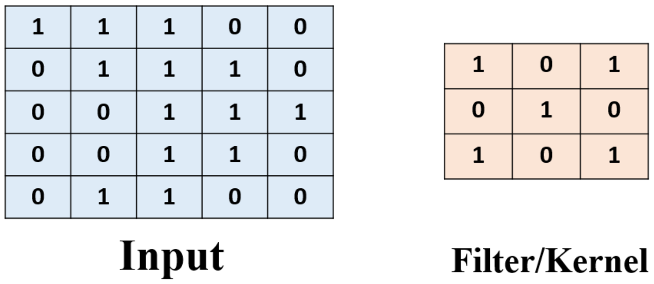
Level 2 offers a more detailed look at the processes. First, the leaf images were collected, and a training dataset created. The dataset was then pre-processed. In this step, the images will be resized into 224x224 resolution to reduce the computational time and then labeled. Then the CNN models with 3x3 order convolution layers, 2x2 order max-pooling layers, fully connected layers (FC), and a softmax classification layer are built. Then the CNN models are validated and saved. Fig 3 shows Level 2.

V. Convolutional Neural Networks

CNN based deep learning methods are comparatively helpful for image classification tasks as they can learn high-level features effectively [7]. To learn the features of an image, CNN uses more than one layer of the perceptron. So CNN is another type of multilayer perceptron. [4]. The CNN architecture used in this work shown in Fig 4.

1) Convolution Layer

The first block of CNN is that the convolutional layer. Primary purpose of convolution in CNN is to extract features from the input image. Convolution is a mathematical operation to combine two sets of data. The convolution is applied to the input data, utilizing a convolution filter to generate a feature map.



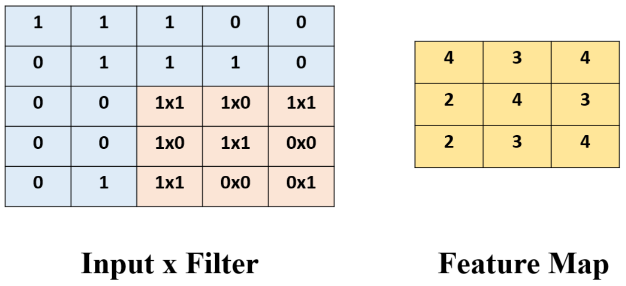
Input

1	1	1	0	0
0	1	1	1	0
0	0	1	1	1
0	0	1	1	0
0	1	1	0	0

Filter/Kernel

1	0	1
0	1	0
1	0	1

In the figure given above, the input to the convolution layer is on the left side, for instance, the input image. On the right side is the 3x3 convolution filter, also called a kernel, often called so because of the filter shape.



Input x Filter

1	1	1	0	0
0	1	1	1	0
0	0	1x1	1x0	1x1
0	0	1x0	1x1	0x0
0	1	1x1	0x0	0x1

Feature Map

4	3	4
2	4	3
2	3	4

Fig. 5. Convolution Operation

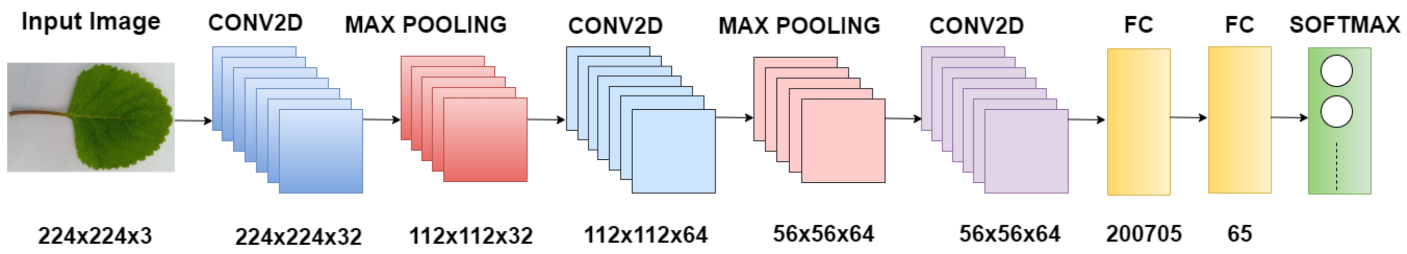


Fig. 4. Architecture of Convolutional Neural Network

The convolution operation performed by sliding this filter over input. Then perform elementwise matrix multiplication at every location. Add the result, and that sum goes into the feature map. In Fig 5, the area where convolution operation takes place is marked as red, termed a receptive field. This field is also 3x3 because of the dimensions of the filter.

- Stride
  In convolution, the number of steps moving denotes stride. By default, it is one. When stride=1, then move the filters to 1 pixel at a time.
- Padding
  Padding helps to preserve the original dimensions of the input. Here zeros are added to outside of the input. The number of zero layers depends upon the size of the kernel.

$$OutputSize = (\frac{n+2p-f}{s}+1)x(\frac{n+2p-f}{s}+1) \quad (1)$$

-Where n - input size, p - pad, f - filter size, and s - stride.

ReLU Function

The activation function comes after the convolutional layer. The Rectified Linear Unit is the activation function used here. The function returns that value, for any positive input, and 0 otherwise. That is,

$$R(z) = max(0, z) \quad (2)$$

2) **Pooling Layer**

Pooling performed to reduce dimensionality. It reduces the number of parameters, that minimizes the training time, and combats overfitting. Pooling layers downsample on each feature map independently while keeping important information. This layer also sums up the information obtained from the previous layer.

Here max pooling is used, which takes the max value in the pooling window by sliding a window over its input. Pooling does not have any parameters. But it has two hyperparameters, that is, Filter(F) and Stride(S).

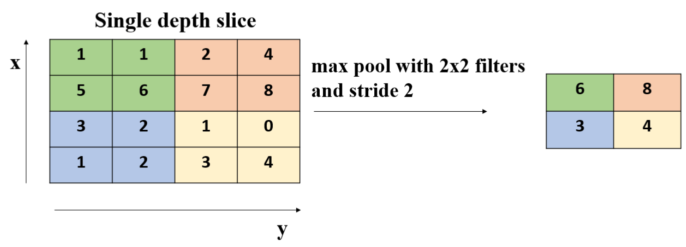


Fig. 6. Max Pooling

Fig 6 shows the max pooling. Each window denoted by a different color. Pooling mostly is done employing 2x2 windows, stride 2, and no padding. And convolution completed using 3x3 windows, stride 1, with padding.

3) **Fully Connected Layer(FC)**

After the convolution and pooling layers, add fully connected layers. This layer contains weights, biases, and neurons. It will connect neurons in one layer to neurons in another layer. It can classify images between different categories by training. Since a fully connected layer expects a 1D vector of numbers, the output of the final pooling layer flattened to a vector, and that becomes the input to the fully connected layer.

In this layer, the number of weights is n*m, with n inputs, and m outputs. This layer has (n+1)*m parameters with a bias for each output node.

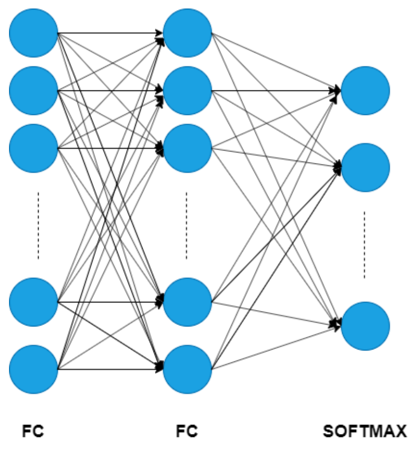


Fig. 7. Fully Connected Layer

4) **Softmax Layer**

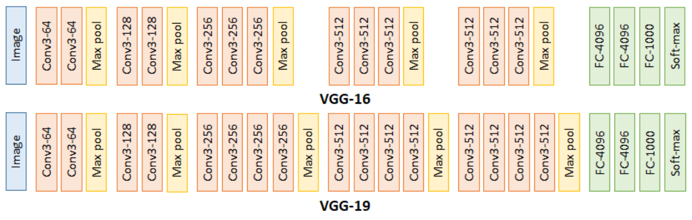
The final layer is the softmax layer. Softmax is for multi-classification. The softmax classification layer also mentioned as the softmax function that yields the predicted probability of each group and fully connected to the final fully connected layer. Softmax functions are multi-class sigmoids. They used in determining the probability of multiple classes at once.

VI. VGGNet

VGGNet is the runner-up at the ILSVRC (ImageNet Large Scale Visual Recognition Challenge) 2014 competition and established by Simonyan and Zisserman. VGGNet contains up to 19 weight layers. It is very appealing due to its very uniform and effective architecture for applications involving image classification. The VGGNet is pre-trained using ImageNet, a large public image repository. During training, the input is a 224 x 224 RGB image. It contains only very small 3x3 convolutions. The convolution stride and padding fixed to 1 pixel. The activation function used is the ReLU function. Max-pooling performed with 2 x 2 pixel window and stride 2. The width of convolution layers (the number of channels) is small, ranging from 64 in the first layer after that, increased by a factor of two, after each max-pooling layer, until it reaches 512. It is trained on four GPUs for 2-3 weeks.

1) **VGG16**

VGG16 is a convolutional neural network architecture with 16 layers. Its layers comprise of Convolutional layers, Max Pooling layers, Fully connected layers, and a softmax layer. The number of parameters in VGG16 is 138 million. There are 13 convolutional layers, 5 Max Pooling layers, and 3 Fully Connected layers that sum up to 21 layers but only 16 weight layers. In the end, we have Fully connected layers (FC) with 4096 units, and a softmax output one of 1000 classes. VGG16 network trained with over 14 million images and 1000 classes.



2) **VGG19**

VGG19 trained on over a million images from the ImageNet database. The depth of the configuration is 19 weight layers and may classify images into 1000 object classes. The number of parameters in VGG19 is 144 million. There are 16 convolutional layers and three Fully Connected layers, which sum up to 19 weight layers. In the end, we have two FC layers that have 4096 units each, and the third with 1000 units that perform 1000-way classification (one for every class). The softmax layer is the final layer.

VII. Results And Discussion

A comparative study between pre-trained CNN models, namely VGG16 and VGG19, and basic CNN, was demonstrated. Leaves of 64 species of plants used for this experiment. It consists of 64000 leaf images with 1000 samples per species. Here, 80% of the data used for training, and the other 20% used for testing. 10% is used for validation. The number of epochs set to 25. The batch size set to 16. In the end, test accuracy is recorded.

As discussed earlier, first we will build a basic convolutional neural network from scratch, train it on the training image dataset, and evaluate the model. The architecture of CNN comprises of three convolutional layers with kernel size 3x3 and activation function as ReLU. This layer coupled with 2D max-pooling of size 2x2. The first 3 convolutional layers act as feature extraction. The output of the layer is input to the dense fully connected layer where the final prediction is done [8]. Finally, we will leverage the pre-trained models VGG16 and VGG19 which is already trained on a huge dataset with a diverse range of categories to extract features and classify images.

The training loss values, training accuracy values, validation loss values, and validation accuracy values at successive epochs of CNN, VGG16, and VGG19 are shown in Fig 8, 9, and 10 respectively. The progress of accuracy and loss functions, during the training process and validation process, are shown in the graphs. The blue line shows the training loss. The orange line shows training accuracy. The green line shows validation loss, and the red line shows validation accuracy.

By referring to TABLE II, the experimental results show that VGG16 achieved the highest classification accuracy, which is 97.8%. VGG19 achieved 97.6% accuracy, and basic CNN achieved 95.79% accuracy. From the results obtained, it shows that VGG16 outperformed VGG19 and basic CNN.

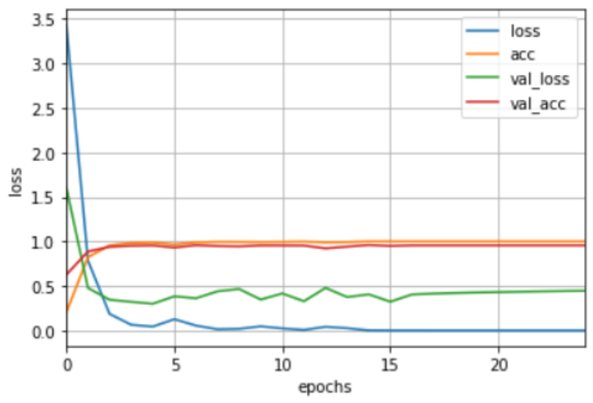


Fig. 8. Convolutional Neural Network

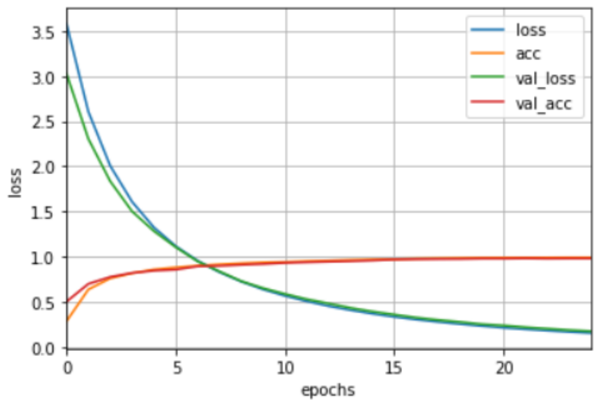


Fig. 9. VGG16

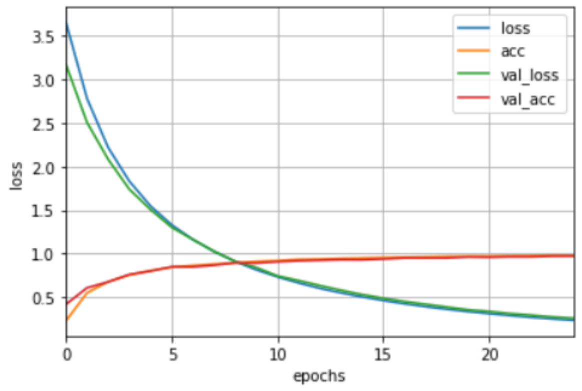


Fig. 10. VGG19

VIII. Conclusion

In this paper, the comparison between basic CNN, VGG16, and VGG19 has been performed concerning accuracy. The

TABLE II
ACCURACY OF DIFFERENT MODELS

Model	**Accuracy**
CNN	95.79
VGG16	97.8
VGG19	97.6

proposed dataset is employed for training and testing purposes. CNN has achieved an accuracy of 95.79%. VGG16 and VGG19 achieve an accuracy of 97.8% and 97.6% respectively, outperforms basic CNN. This work helps to identify indigenous medicinal plants.

As future work, we can perform image classification with larger datasets. Object recognition from videos also recommended. That is when a video is given as input, it can identify various medicinal plants in the video.

References

[1] Vishal Batvia, Drashti Patel, and Dr. Avani R. Vasant, "A Survey on Ayurvedic Medicine Classification using Tensor flow," International Journal of Computer Trends and Technology, Volume 53, Number 2, November 2017.

[2] Ming Liang and Xiaolin Hu, "Recurrent Convolutional Neural Network for Object Recognition," IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3367-3375, 2015.

[3] Karen Simonyan, and Andrew Zisserman, "Very deep convolutional networks for large-scale image recognition.," arXiv preprint arXiv:1409.1556, September 2014.

[4] Jing Wei Tan, Siow-Wee Chang, Sameem Abdul-Kareem, Hwa Jen Yap, and Kien-Thai Yong,"Deep Learning for Plant Species Classification using Leaf Vein Morphometric," IEEE/ACM Transactions on Computational Biology and Bioinformatics, Volume 17, Issue 1, June 2018.

[5] Nurbaity Sabri, Zalilah Abdul Aziz, Zaidah Ibrahim, Muhammad Akmal Rasydan Bin Mohd Rosni, and Abdul Hafiz bin Abd Ghapul,"Comparing Convolution Neural Network Models for Leaf Recognition," International Journal of Engineering Technology, 7 (3.15) 141-144, 2018.

[6] M.R. Dileep and P.N. Pournami,"Ayurleaf: A deep learning approach for classification of medicinal plants," IEEE Region 10 Conference (TENCON), 2019.

[7] Prakruti Bhatt, Sanat Sarangi, Anshul Shivhare, Dineshkumar Singh, and Srinivasu Pappula,"Identification of Diseases in Corn Leaves using Convolutional Neural Networks and Boosting," In Proceedings of the 8th International Conference on Pattern Recognition Applications and Methods (ICPRAM 2019), pages 894-899.

[8] Srikanth Tammina,"Transfer learning using VGG-16 with Deep Convolutional Neural Network for Classifying Images," International Journal of Scientific and Research Publications, Volume 9, Issue 10, October 2019.