Neural Computing and Applications (2024) 36:5399–5412
https://doi.org/10.1007/s00521-023-09352-9

ORIGINAL ARTICLE

CNN-based Indian medicinal leaf type identification and medical use recommendation

S. Praveena[1] · S. M. Pavithra[1] · A. Dalvin Vinoth Kumar[2] · P. Veeresha[1]

Received: 25 July 2023 / Accepted: 28 November 2023 / Published online: 3 January 2024

Abstract

Medicinal leaves are playing a vital role in our everyday life. There are an enormous amount of species present in the world. Identification of each type would be a tedious task. Using image processing technology, we can overcome this problem by providing computer vision with the help of a convolution neural network (CNN). The objective of this research is to find out the best CNN model that helps in classifying the plant leaf species and identifying its category. In this research work, the proposed basic CNN model consisting of four convolution layers uses ten different medicinal leaf species each belonging to two categories providing an accuracy of 96.88%.

Keywords Convolution neural network · Graphical user interface · Gradio · Confusion matrix

1 Introduction

Indian medicinal leaves are proven to be life-saving for many of the living creatures in this world. Many Ayurvedic Indian literature have stated the medicinal value of plants in treating various human diseases [1]. The traditional usage of Indian plants with medicinal properties has been practiced in India for centuries and has continued to date. Some of the high-potential medicinal plant leaves, such as Bael leaves, Betel leaves, Black Nightshade leaves, Clover leaves, Curry leaves, Fenugreek leaves, Lantana leaves, Mint leaves, Neem leaves and Tulsi leaves, are considered for this study. During the growth of the plant due to various factors present in nature and caused by humans, leaves might become prone to certain diseases. Diseased or unhealthy leaves cannot be used for consumption purposes.

In India, there exists an extremely huge number of plant species. Hence it is not possible even for the specialist to identify species correctly [2]. Also, some of the species have similarities, which may take even longer time to differentiate them. Hence, the use of technological development plays a vital role. To make such tasks easy we make use of deep learning models, such as image processing. This not only helps the specialist but also helps laymen and ordinary people to identify plant species within seconds.

Data science is an interdisciplinary field that is used to gain knowledge and obtain insights from the data. Data science uses mathematics and statistics predominantly. Based on the type of data considered, we use various proven techniques to obtain the desired results. One of the main branches of data science is image processing. Image processing was initially used to obtain high-quality images from low-quality images. Nowadays, it is applied to various tasks which include pattern recognition, classification tasks, identification tasks, and much more. Artificial intelligence mimics the human brain. In AI technology, deep learning is one such, which learns features on its own [3]. One of the deep learning neural networks that are used to make computers see is the convolution neural network (CNN). CNN is designed to analyze and process image datasets to perform computer vision tasks, such as image processing. CNN is a mathematical construction that is helpful in feature extraction and it helps in providing deep

✉ P. Veeresha
pundikala.veeresha@christuniversity.in

S. Praveena
praveena.s@maths.christuniversity.in

S. M. Pavithra
pavithra.sm@maths.christuniversity.in

A. Dalvin Vinoth Kumar
dalvin.vinoth@christuniversity.in

[1] Department of Mathematics, CHRIST (Deemed to be University), Bengaluru 560029, India

[2] Department of Statistics and Data Science, CHRIST (Deemed to be University), Bengaluru 560029, India

information about images. The layers that extract features are convolution and pooling. The extracted features are mapped to the output with the help of the fully connected layers [4].

Consumption of green leaves in our diet is proven to give us various health benefits. Digitalization of the identification of medicinal plants would help in medicine manufacturing industries. Regional medicinal plant leaves are unexplored in the image processing area. Thus we are inspired to connect our regionally available medicinally benefiting plant leaves and image processing. Since in the real world, not all the leaves that we see or consume are healthy ones, there might be some leaves that are prone to certain diseases caused by fungus or bacteria, such as leaf spots, leaf blight, rusts, decolorization, leaf wilt, leaf canker. Modeling an image processing model to classify the leaf type and identify whether it is healthy or diseased is the main motivation for this study.

Our study focuses on classifying the medicinal leaves of Indian plants based on images. To achieve this, a convolution neural network (CNN) is used. Medicinal leaves of different plants mentioned above are used for model classification and recommendation systems. The study aims to obtain maximum accuracy of the trained model and best test accuracy. CNN helps to extract features and learns the difficult patterns and helps in the recognition of the target leaf type [5]. The dataset consists of healthy and unhealthy leaves (diseased leaves) of each leaf type. Medicinal leaves were collected from different sources in Bengaluru, Karnataka. The collected dataset is preprocessed and then the basic CNN model is executed on the image dataset that is split as train, validate, and test datasets.

The novelty of our study is the collection of images of the medicinal leaves in the surroundings of Karnataka that are required for our study. Most of the studies are done on existing readily available datasets from online repositories which may be biased or show significant computational resources. Datasets play a vital role in building a model. Hence the dataset is collected from scratch by authors, and an exclusive study is done on the same. We attempt to classify healthy and unhealthy or diseased leaves. Existing research works have implemented high computational models which are time-consuming. Our attempt is to build a basic CNN model that is efficient and easy to compute. The model is then extended to recommend the classified leaves to consume the leaf or to discard it. The usage of GUI as the final output is exclusive in our study as existing research works have focused on accuracy alone.

2 Related works

In Wu et al. [6] have used 1800 leaves of 32 kinds of plants. The proposed model is probabilistic neural network (PNN) which has a training speed faster than a Backpropagation network. PNN has 3 layers. They are input, RBF, and competitive layers. The image is then converted to a vector which is inputed to the PNN. Using PNN, 90% of accuracy is obtained. In Anami et al. [7], have collected plants and tree images from the herbarium and farms of Karnataka. In total 900 images were collected from herbs, shrubs, and tree species, 300 images of each type. Papaya, Neem, Tulsi, Aloe, Garlic, Tengu, Bevu, and Ekki are the types of species collected. The methodology implemented in this paper is SVM and NN classifier. Using the color texture feature, 70%, 65% and 94%, 90% are the minimum, and maximum accuracy obtained using SVM and NN classifiers, respectively. Using edge-based texture feature 80% is the accuracy.

In Gopal et al. [8], used 100 leaf images in each of the 10 plant species for training the model and 50 leaves belonging to different plant species for testing the model. Using optimal methods in image processing, classification, feature extraction, and pattern recognition is performed. The proposed algorithm has the following steps, capturing leaf image, feature extraction, non-green part removal, standardized leaf image, and classification and results. Using leaf edge, boundary-based feature, moment feature, and color feature, the proposed algorithm gave 92% accuracy. Future works are to use leaf venation and texture feature to improve accuracy.

In Kan et al. [9], have collected 12 different medicinal plant leaf images. Using the shape feature and texture feature, and converting a color image to grayscale and then to a binary image, obtaining leaf contour. The methodology they have implemented is SVM classifier. An accuracy of 93.3% is obtained. In Anh et al. [10], have collected images from the environment of Vietnam. They have collected 10279 images of the 10 herbal plant species. They have also other machine learning models, the highest accuracy they obtained is 93% using the lightGBM model. In Dudi et al. [11] have used Flavia dataset. They have used four-layered CNN model along with the ML model and obtained an accuracy of 98%. Training and validation accuracies are matching. The confusion matrix is constructed for the test dataset and performance metrics precision, recall, F1-score, and support are calculated.

In Izwan Asraf Md Zin et al. [12], collected herbal plant images along with leaves which were captured at an herbal nursery in different to increase image varieties. Around 10 images from each class with 2736 × 3648 dimensions are collected for the CNN model. The images were rotated by 10^o and 30^o to increase the number of images dataset, by increasing 160images in each. Used 12 different CNN models along with Alex net, Google net, and Squeeze net and respective accuracies are compared with each other on the basis of convolution layers, pooling layers, and epochs.

In Anchitaalagammai et al. [13] have used 58280 images of Basil, Jamun, Jatropha curcas, kuppaimeni, Pungai, each of species with approximately 10000 images. They have split the dataset into training, testing (50 images each), and validation set(5 images each). CNN model was used. Using the inception V3 model for classifying leaves, 96.97% accuracy is obtained. Increasing the depth of the model sometimes might overtrain the model. Future work of this work is to increase the number of classes. In Malarvizhi et al. [14] have used the Flavia dataset that contains 1907 images of 32 different species. Original images are of 6016×4016 resolution which is resized to 1600×1200 resolution. RGB images are converted to gray-scale images. Noise removal is one of the preprocessing tasks performed. Background-removed images are used for processing. They extract the features using contours. Classification tasks were performed by SVM, k-NN, and Random forest and obtained an accuracy of 88%, 84%, and 90%, respectively

In Rahim Azadnia et al. [15], have collected leaf images of lemon balm, peppermint, bael, stevia, and tulsi. The original image dataset has 750 images. A total of 13500 augmented images were collected and 80% for training and 20% for testing the model. The proposed CNN model consists of four layers. They are Global Average Pooling layer, dense layer, dropout layer, and softmax layer. Using this, 99% accuracy is obtained. CNN model is performed for different image sizes 64×64, 128×128, and 256×256. Using the confusion matrix, individual accuracy and other measures are listed.

In Mirajul Islam et al. [16], have collected 3785 images of five types of spinaches. They have collected datasets from the local vegetable market and spinach fields. They used 4 types of CNN models, namely Inception V3, Xception, VGG19, and VGG16, and obtained an accuracy of 98.68%, 99.47%, 99.26%, 99.79%, respectively. Out of all the images 757 images are tested and 753 images were classified correctly. Results are also established using the confusion matrix and its evaluation metrics. We referred and there in [17–19].

3 Proposed model of the work

The proposed model uses a basic CNN architecture, which does feature extraction during the training process. These extracted features are then fed into the last few layers of CNN which are the fully connected layers. The output is then generated at the softmax layer. The model is ready for prediction, which makes identification and classification tasks. At this stage, the input leaf is identified as which type and then classified into one of the two classes—healthy or diseased. Model evaluation is performed with the help of test images and the results are also summarized in the confusion matrix. The proposed model mainly focuses on the recommendation of the leaf for consuming it or not. This phase of the proposed model is purely dedicated to the recommendation system. This phase can also be considered the model deployment stage. A graphical user interface (GUI) is designed using the Gradio package. Using Gradio for the recommendation system, it creates a web link where users can input leaf images and get their recommendations.

3.1 Data collection

The image data set is prepared by taking the photos of following medicinal plant leaves: Bael Leaf (*Aegle marmelos*), Betel Leaf (*Piper betle*), Black Nightshade Leaf (*Solanum nigrum*), Clover Leaf (*Trifolium*), Curry Leaf (*Murraya koenigii*), Fenugreek Leaf (*Trigonella foenum-graecum*), Lantana Leaf (*Lantana*), Mint Leaf (*Mentha piperita L*), Neem Leaf (*Azadirachta indica*), Tulsi Leaf (*Ocimum tenuiflorum*). All the images are taken in iPhone 11 with high efficiency and default settings at CHRIST (Deemed to be University). Around 1800+ images are collected. The collected images are preprocessed and the images of each leaf type are displayed in Figs. 1, 2.

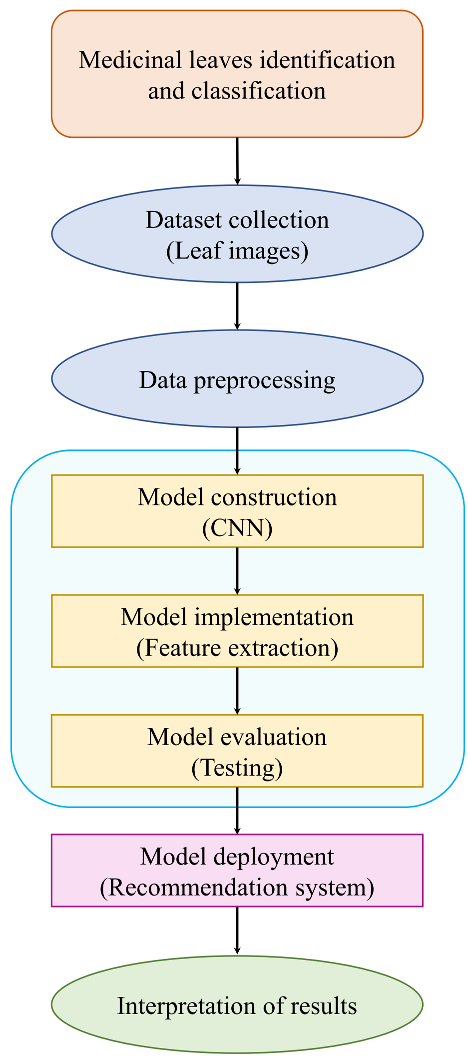


Fig. 1 The steps followed to accomplish the proposed model

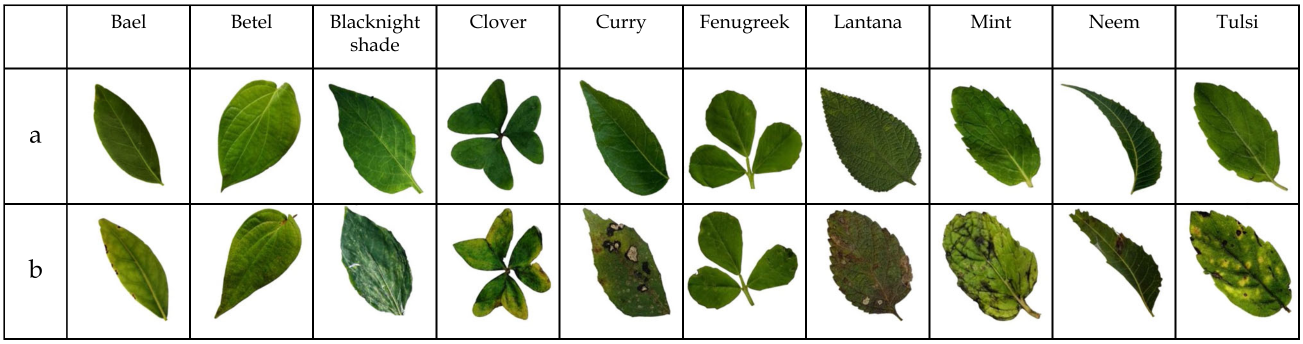


Fig. 2 Figure showing the sample images from each type of leaves, where 'a' indicates healthy leaves and 'b' indicates unhealthy leaves

Bael leaf can be used to treat dysentery, dyspepsia, malabsorption, neurological illnesses, edoema, vomiting, and rheumatism conditions. Unhealthy bael leaf has white floury patches on leaflets or fungal dark spots [20]. Betel leaf can be used to treat bronchitis, constipation, indigestion, congestion, coughs, asthma, sore throats, headaches, and obstructed or insufficient urine passage. An unhealthy betel leaf has a pale yellow color with big black specks in the center or withering and dryness [21]. Black nightshade leaf can be used to treat rheumatoid and gouty joints, skin conditions, TB, nausea, neurological disorders, inflammations, and skin conditions. An unhealthy black nightshade leaf has curled leaves as might be infected by aphids, spider mites or early blight [22]. Clover leaf can be used to treat injuries to the mouth and throat, fever, skin blemishes, menopause issues, cardiovascular, thyroid, diarrhea, and stomach problems. Unhealthy clover leaf has small circular brow-black spots or yellow streaks parallel to leaf veins, yellow-green mottling on leaves [23].

Curry leaf can be used to treat edoema, bruising, piles, diarrhoea, inflammation, itching, fresh cuts and also to maintain a normal hair tone and promote hair development. Unhealthy curry leaf has circular, asymmetrical, yellowish-brown spots or hole [24]. Fenugreek leaf can be used as an antioxidant, decrease blood pressure, Wounds, and sore muscles treatment, anti-cancer agent, anti-ulcer agent, gastro- and hepatoprotective, digestive and appetizer. Unhealthy fenugreek leaf has white, powdery spots, yellow spots or circular sunken lesions with chlorotic halos, and necrotic areas [25]. Lantana leaf can be used as an antifungal, anti-ulcerogenic, hemolytic, antihyperglycemic, wound healing, antimotility, anti filaria, antiinflammatory, antiurolithiatic, anticancer, antiproliferative, antimutagenic, antioxidant, and mosquito-controller. Unhealthy lantana leaf has powdery mildew, white or gray powdery substance, twisted, discolored leave [26]. Mint leaf can be used to treat rheumatism, neuralgia, congestive headache, and toothache. It can be used to prepare sweets, teas, mouthwash, toothpaste, alcoholic liqueurs, jellies, syrups, ice cream, cough drops, chewing gum, confections, soaps, detergents, and insect repellent. Unhealthy mint leaf has yellow, deformed, necrotic patches on leaves or sooty mold released by aphids [27].

Neem leaf can be used to treat cancerous cells, harmful bacteria, and heal dental diseases. Unhealthy neem leaf has twig blight which is caused by fungus [28]. Tulsi leaf can be used to treat diabetes, hypertension, cancer, rheumatoid arthritis, different bacteria, parasites, stomach ulcers, liver damage, infection, and as herbal supplements. Unhealthy tulsi leaf has yellow, discoloration, gray fuzzy growth on the lower surface, brown to black angular patches, and circular to irregular dark back or brown spots on them with light centers [29].

Data pre-processing

Data preprocessing is one of the common methods that is performed to implement the model, where we apply different techniques to raw images to prepare the input data. The purpose is to increase the quality of the information such that the model is able to be analyzed easily. The originally collected raw image dataset consists of approximately 1800 images belonging to 20 classes, 10 classes with each class having healthy and diseased classes. In order to obtain better accuracy of the model certain preprocessing techniques are applied to the raw dataset (Figs. 3, 4). They are listed as follows:

- Categorisation The raw images that are collected together are segregated and categorized by creating different folders, which makes the model take these folder names as the labels for the images and it learns it and identifies when given a test image.
- Cropping Cropping is a process of reducing the height and width of the image without losing its quality. Cropping is performed by focusing on the major portion of the image consisting of the leaf.
- Brightness correction Brightness correction is a process where the brightness of each of the pixels of the image is corrected (here brightness is increased) without losing the

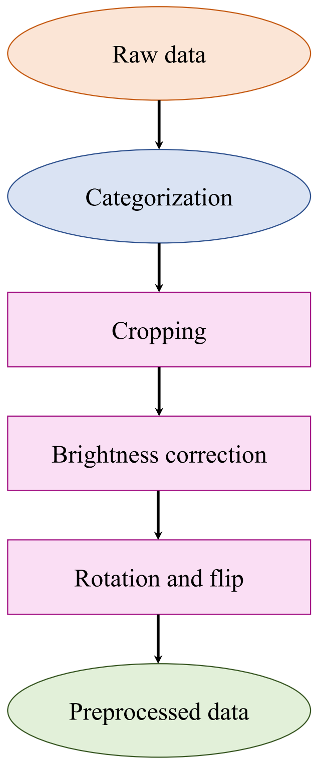


Fig. 3 The steps followed to obtain preprocessed dataset from raw dataset

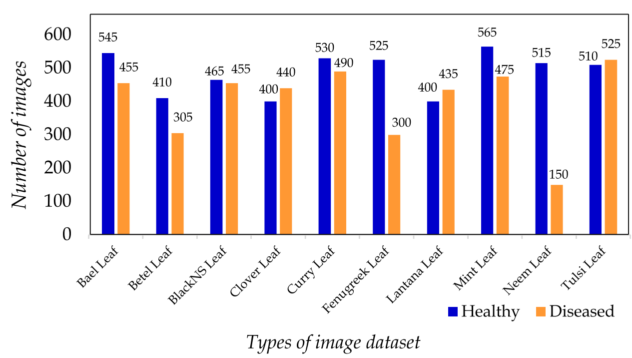


Fig. 4 Graph showing the number of images from each type of leaves present in the dataset

quality of the image. This process made the background and leaf look bright and this helped in a drastic change in the accuracy of the model.

- Geometric transformation Geometric transformation refers to the rotation or flipping of the image. The image is rotated by 90^o to the right and then these rotated images are flipped vertically. The original bright images are flipped horizontally.

After performing all the above preprocessing techniques, the images are combined, respectively, and a total of approximately 9000 images is used for this study. In order to train and test the model, the image dataset is divided in this ratio $8:1:1$. The training dataset consists of 80% of the image dataset. The validation dataset consists of 10% of the image dataset. The remaining 10% of the data remains as never-seen image data for the model, which is the test dataset.

CNN model construction

The proposed model of this paper which is used to classify and recommend medicinal leaves for consumption purposes is the basic CNN model given in Fig. 5 that consists of 4 convolution layers, 4 pooling layers, 2 fully connected layers, and an output layer.

Input layer

The image is the input for the CNN model. Random-sized images are given as input to the model by reshaping it to 227×227 pixels with 3 channels (RGB - red, green, blue) i.e., (227, 227, 3) (Fig. 6).

Activation functions

The activation function used in the CNN model defines the output of the layer given an input or set of inputs. It is used to make models learn the complex patterns of the input data [30]. The activation functions used in this model are ReLU and Softmax. ReLU is given by $f(x) = \max(0, x)$. The positive and negative gradient change of the ReLU function in the x-axis is constant and zero, respectively [31]. The softmax function is defined as $f(x_i) = \frac{e^{x_i}}{\sum_j e^{x_j}}$. The Softmax activation function is used in multi-class classification models, where it returns the probabilities of each class with the target class having the highest probability [32].

Convolution layer

The convolution layer is used mainly to extract features from the input image to give feature map as an output. Linear operation is performed during convolution operation. Nonlinear operation is used as an activation function [33]. Multiple filters undergo this process and produce a number of feature maps, each of these feature maps represents a different characteristic of the input array. The dimension of the output image (feature map) is $n_{\text{out}} \times n_{\text{out}} \times d$ obtained by the formula:

$$n_{\text{out}} = \left(\frac{(n_{\text{in}} + 2p - f)}{S} + 1 \right),$$

where n_{in} is the dimensions of the input image, p is pooling constant, f is number of filters and S is stride.

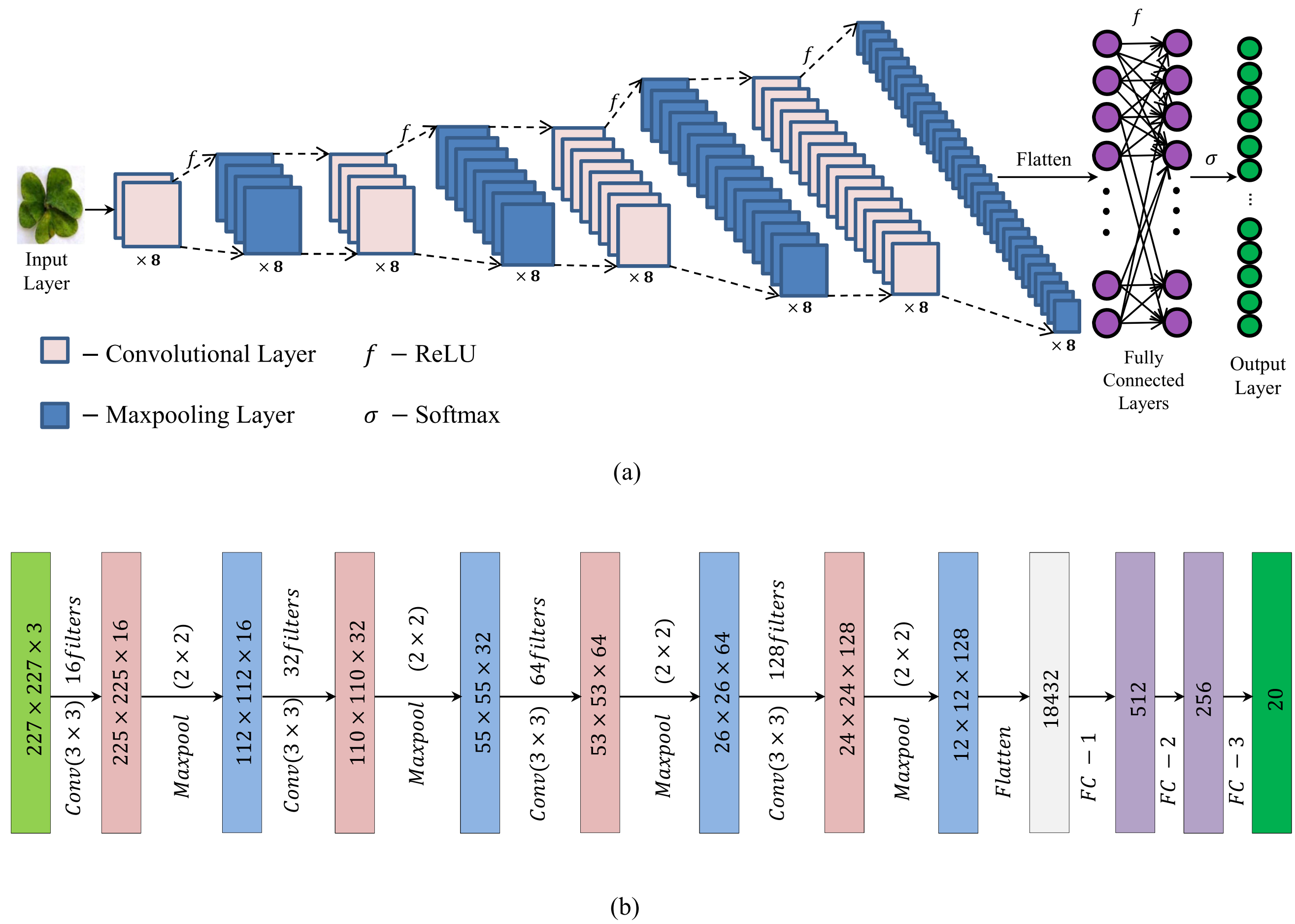


Fig. 5 **a** The block diagram of the proposed model CNN architecture and **b** The dimensions of the blocks present in the block diagram

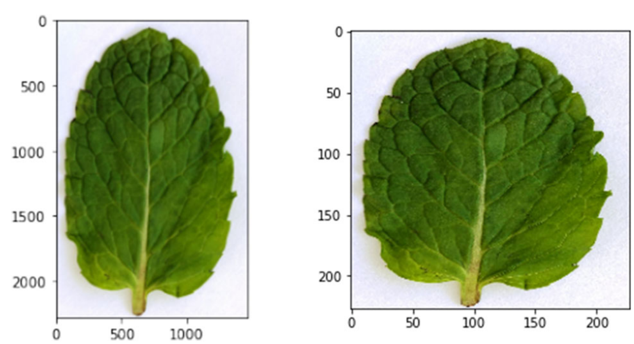


Fig. 6 The original input image with random dimensions and the resized image with dimensions (227 × 227)

Pooling layer

The pooling layer is the layer at which the downsampling of the feature maps takes place. It reduces the dimensionality of the feature maps while keeping depth constant. In the proposed model, the max-pooling operation is used. The dimension of the output image of the pooling layer is $n_{\text{out}} \times n_{\text{out}} \times d$ obtained by the formula:

$$n_{\text{out}} = \left(\frac{(n_{\text{in}} - f)}{S} + 1 \right),$$

where d is the depth of the previous feature map.

Fully connected layers

The last output feature maps are flattened. It gives a one-array of real numbers, which in turn is connected to fully connected layers. These layers are also known as dense layers. FC layers consist of weights and neurons. The connection between neurons of one layer to neurons of another layer takes place. It is used for training models for classification tasks [34].

Output layer

Classification and recommendation are the output of the model. The soft-max activation function is used in the output layer. The immediate output of the model is a (1 × 20) array, whose entries are ranging from 0 to 1, which indicates the probability values of the specific class of leaf to be detected. The indexing position of the entry close to 1 indicates which type of leaf it is classified as in the ground truth labels.

4 Results and discussions

The CNN model learns the patterns based on the features extracted. Convolution layers help in feature extraction. Totally, there are 4 convolution layers used in the proposed model. Each of the convolution layers yields a set of feature maps, which is shown below in Fig. 7. These feature maps are smaller in dimensions when compared to that of the input image. As the number of convolution layers increases, the size of the feature maps decreases. Because of these some information regarding feature extraction could be lost and the image could diminish, hence the execution have been stopped at 4th convolution layer.

The preprocessed dataset is split into training, validating, and testing datasets. The training data along with validating dataset is given as input to the model. The model is constructed using sequential layers of convolutional, max-pooling, and it is fully connected along with the activation functions. The model is made to run for 10 epochs with a learning rate of 0.001. The training, validating accuracy, and loss for every epoch are, respectively, given in Figs. 8 and 9.

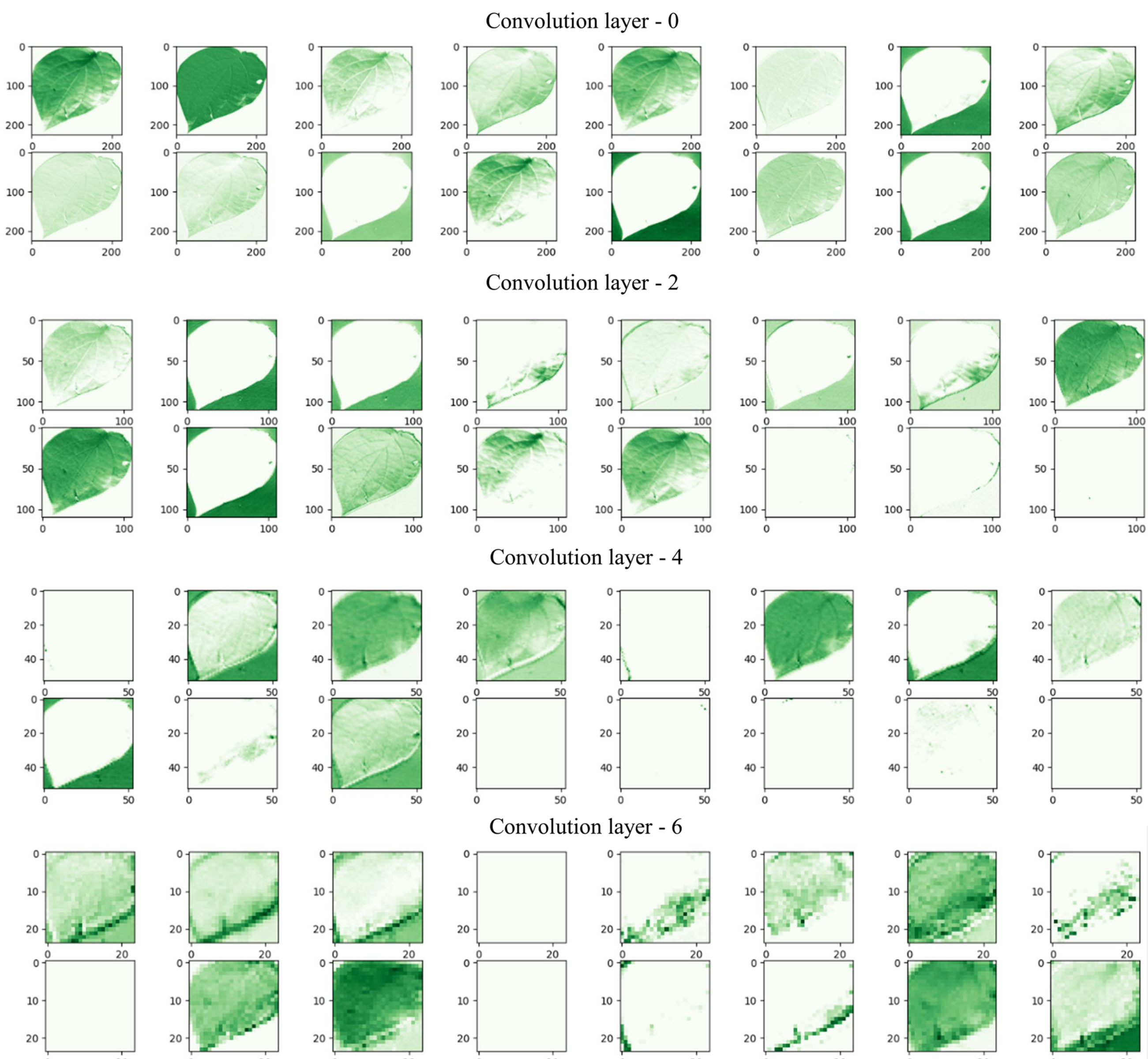


Fig. 7 The different feature maps obtained as an output of the convolutional layers 0, 2, 4 and 6 using different filters

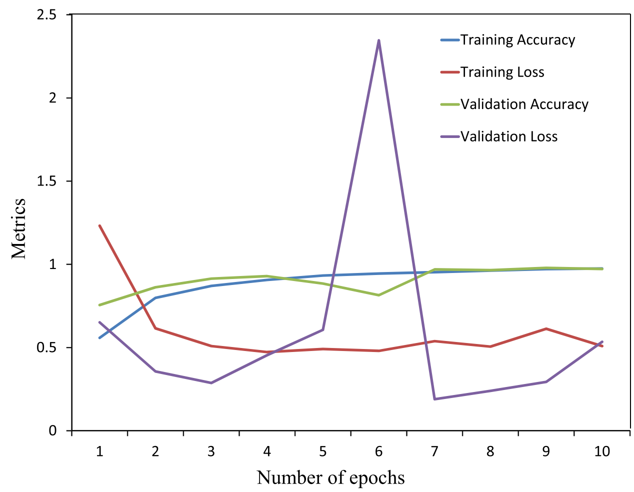


Fig. 8 The graph showing variation of training, validation accuracy and loss for the preprocessed data with respect to 10 number of epochs

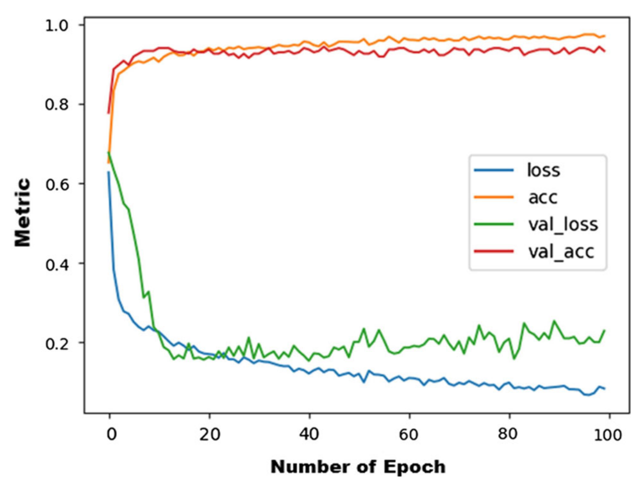


Fig. 9 The graph showing variation of training, validation accuracy and loss for the preprocessed data with respect to 100 number of epochs

The maximum training accuracy obtained by the model is 97.45%. To this trained model, the unseen leaf images present in the testing dataset are made to predict by the model. The model was able to identify 96.88% correctly.

The graph in Fig. 8 gives a comparison between accuracy and loss for both the raw data training model and pre-processed data training model. Once the model is trained, the readings of accuracies and loss are plotted. Raw data accuracy and loss refer to the values from the model that is trained with raw data. It is clearly seen that both losses are decreasing as the number of epochs increases. Also, the training accuracy obtained for pre-processed data is higher than that of raw data.

The validation dataset consisted of 10% of pre-processed data. While training the model along with the validation set, helps the model to validate the model's performance during training. Similar to the previous graph, the validation accuracy and loss are plotted against the number of epochs in Fig. 9. It is observed that accuracy increased and loss decreased, as the number of epochs increased.

Individual test images belonging to different classes are tested and summarized into a matrix format known as a confusion matrix. The confusion matrix is a matrix with dimensions $n \times n$ where n is the number of classes in classification tasks, which helps to visualize the results of a classification problem. It is used to find which classes were classified correctly and incorrectly. Diagonal entries of the below-given confusion matrix represent the correctly predicted leaf types. The row and column headings indicated by [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19] refers to the types of leaves specified in Fig. 2. Column headings and row headings correspond to the Predicted labels and Ground truth labels respectively (Table 1).

Performance metrics comparison

The model is implemented and results are tabulated. With the help of certain tools, the performance of the model is evaluated, which is known as performance metrics or evaluation metrics. In this study, three metrics namely precision, recall and f1 score are evaluated. The obtained results are then compared with that of VGG16 and VGG19. The comparison is visualized in Figs. 10, 11. Let M be the confusion matrix.

Precision

Precision is defined as the ratio of the correctly classified particular class to the classes that are incorrectly predicted as this particular class. It gives the rate of positive prediction.

$$\text{Precision} = \frac{M_{ii}}{\sum_{j=1}^{n} M_{ji}}; \forall i = 1, 2...20.$$

Recall

Recall is defined as the ratio of the correctly classified particular class to the same class incorrectly predicted as other classes. It indicates the actual positive predictions over all the predicted as positive.

$$\text{Recall} = \frac{M_{ii}}{\sum_{j=1}^{n} M_{ij}}; \forall i = 1, 2...20.$$

F1 score

F1 score is defined as the harmonic mean of precision and recall. It is the combination of both precision and recall that

Table 1 Confusion matrix for the test dataset

Ground truth-labels\ predicted labels	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
0	21	28	4	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
1	0	45	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
2	0	0	38	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	31	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	1	0	40	2	0	0	0	0	0	0	2	0	2	0	0	0	0	0
5	0	0	0	0	0	45	0	1	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	36	4	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	44	0	0	0	0	0	0	1	0	0	0	0	0
8	0	0	0	0	0	0	0	0	47	3	0	0	1	0	1	0	0	0	0	1
9	0	0	0	0	0	0	0	0	0	49	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	51	2	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	30	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0	31	0	3	3	0	0	0	3
13	0	0	0	0	0	0	0	0	0	0	0	0	0	44	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	57	0	0	0	0	0
15	0	0	2	0	0	0	0	0	0	0	0	0	0	0	7	40	0	0	0	0
16	0	0	0	0	0	2	0	0	0	1	2	0	0	0	2	0	45	0	0	0
17	0	0	1	0	0	0	0	0	0	0	2	0	0	0	0	0	0	13	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	16	3	0	0	17	15
19	0	0	0	0	0	0	0	0	0	0	1	0	0	0	3	0	0	0	0	49

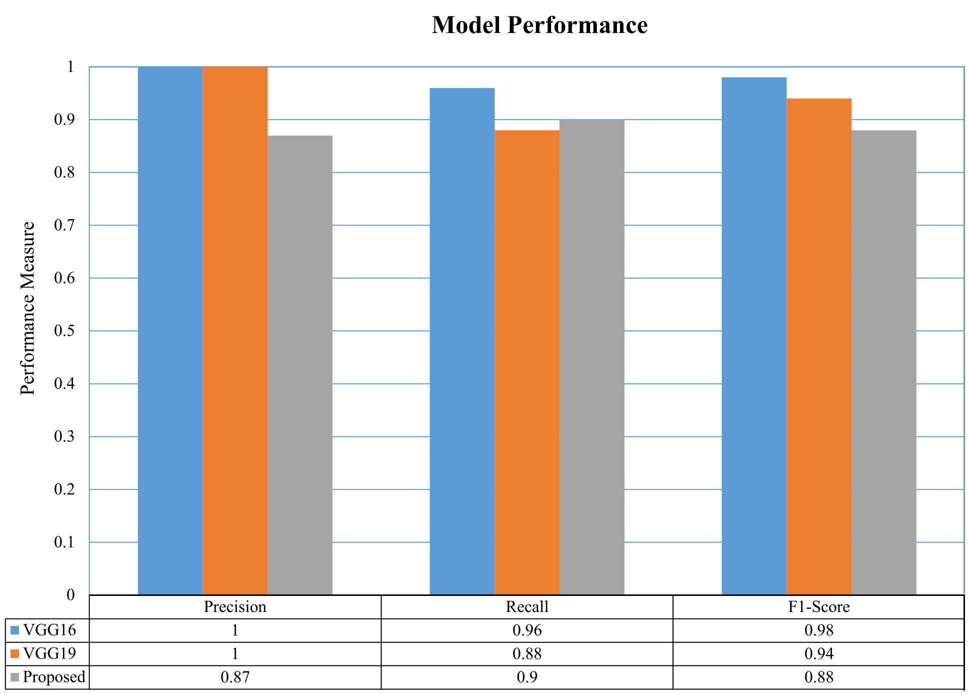


Fig. 10 Comparison of proposed model performance metrics with VGG16 and VGG19

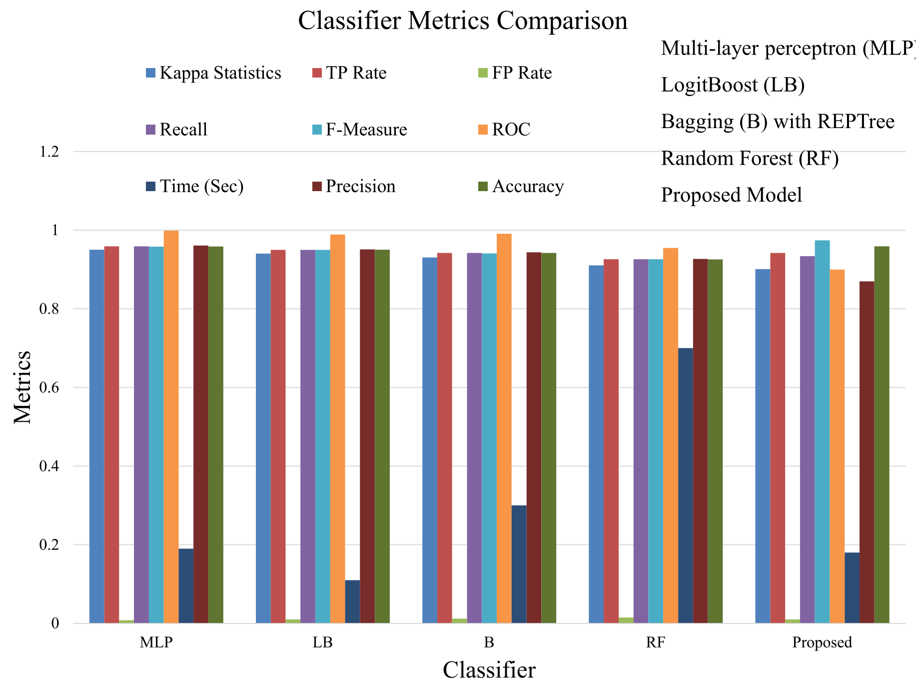


Fig. 11 Comparison of proposed model with classifiers based on parameters

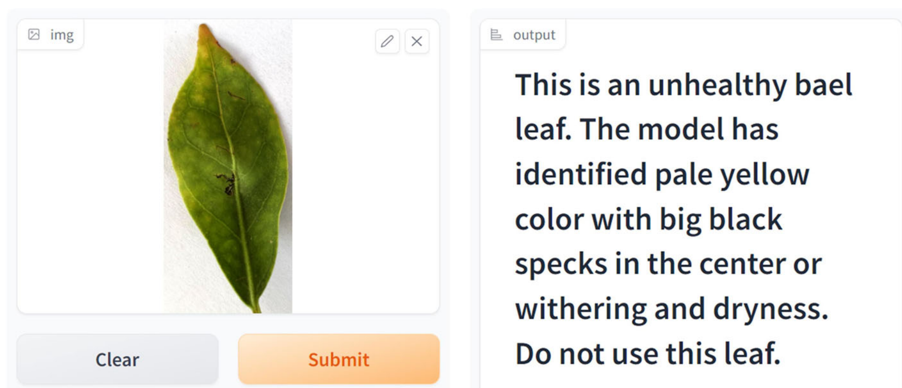


Fig. 12 The recommendation obtained as an output when a leaf image is submitted into the gradio user interface panel

can be used to obtain the overall performance of the model.

$$\text{F1 score} = \frac{2}{\frac{1}{\text{Precision}} + \frac{1}{\text{Recall}}}.$$

Recommendation system

Once the model is trained and tested with respective datasets, it is taken one step ahead that is to the recommendation step. In the recommendation system, we have used a graphical user interface (GUI) as an application of the built CNN model. GUI refers to the interface in which the user interacts directly with the computer. Using gradio, the GUI is implemented. Now, the recommendation for each input leaf image is given in Fig. 12.

5 Conclusion

Indian medicinal plants possess great potential to cure many diseases. We have developed a CNN model which is effective in the identification of such plant leaves and have implemented a recommendation system to suggest the medical use of leaves and whether to consume or not with a related reasonable explanation for the same. They are many image processing models existing but all those require high computational power, but we obtained the results with less computational power. We have collected the dataset from scratch and have built the Basic CNN model that gave 97.45% training accuracy and 96.88% testing accuracy. The predicted class labels of the test dataset leaves are tabulated in the confusion matrix and performance metrics are evaluated and compared with the existing models. Future works may include the study of more Indian medicinal leaves.

Author contributions All the authors have contributed equally to this work.

Data availability statement The data that support the findings of this study are available from the first author upon reasonable request.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

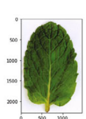
Appendices

Importing all the necessary packages:

```
import tensorflow as tf
import matplotlib.pyplot as plt
import cv2
import os
import numpy as np
from tensorflow.keras.preprocessing.image
import ImageDataGenerator
from tensorflow.keras.preprocessing
import image
from tensorflow.keras.optimizers
import RMSprop
```

Analyzing the original dimensions of the input image:

```
img=image.load_img("leaf.jpeg")
plt.imshow(img)
Out:
```

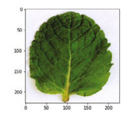


Convert image to n-dimensional array, this is how the model reads the image:

```
cv2.imread("leaf.jpeg")
cv2.imread("leaf.jpeg").shape
Out: (2280, 1461, 3)
```

Reducing the dimensions of the image to the target size $227 \times 227 \times 3$

```
img=image.load_img("leaf.jpeg",target_size=(227,227))
plt.imshow(img)
Out:
```



Used to rescale pixels values range from 0 to 255:

```
train=ImageDataGenerator(rescale=1/255)
validation=ImageDataGenerator(rescale=1/255)
```

Importing training data from the directory (80% of the data):

```
train_dataset=train.flow_from_directory("trainsplit/train",
                                         target_size=(227,227),
                                         batch_size=3,
                                         class_mode="categorical")
Out: Found 7117 images belonging to 20 classes.
```

Importing validating data from the directory (10% of the data):

```
validation_dataset=validation.flow_from_directory("trainsplit/val",
                                         target_size=(227,227),
                                         batch_size=3,
                                         class_mode="categorical")
Out: Found 884 images belonging to 20 classes.
```

Printing the indices for the data:

```
train_dataset.class_indices
train_dataset.classes
Out: array([ 0, 0, 0, ..., 19, 19, 19])
```

Model construction using sequential layers of convolution,maxpooling and fully connected:

```
model=tf.keras.models.Sequential([tf.keras.layers.Conv2D(16,(3,3),
activation="relu",
input_shape=(227,227,3)),
tf.keras.layers.MaxPool2D(2,2),
#
tf.keras.layers.Conv2D(32,(3,3),activation="relu"),
tf.keras.layers.MaxPool2D(2,2),
#
tf.keras.layers.Conv2D(64,(3,3),activation="relu"),
tf.keras.layers.MaxPool2D(2,2),
#
tf.keras.layers.Conv2D(128,(3,3),activation="relu"),
tf.keras.layers.MaxPool2D(2,2),
##
tf.keras.layers.Flatten(),
##
tf.keras.layers.Dense(512,activation="relu"),
##
tf.keras.layers.Dense(256,activation="relu"),
##
tf.keras.layers.Dense(20,activation="softmax")
])
```

Definig the learning rate, metrics:

```
model.compile(loss="categorical_crossentropy",
          optimizer= RMSprop(learning_rate=0.001),
          metrics=['accuracy'])
```

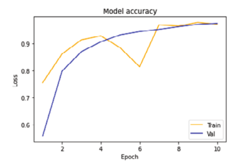
Training the model for 10 epochs:

```
model_fit=model.fit(train_dataset,
                   batch_size=256,
                   epochs=10,
                   validation_data=validation_dataset)

Out:
Epoch 1/10
2380/2380 [==============================] - 4060s 2s/step -
 loss: 1.2414 - accuracy: 0.5558 - val_loss: 0.6066 -
  val_accuracy: 0.7867
Epoch 2/10
2380/2380 [==============================] - 898s 377ms/step -
 loss: 0.5249 - accuracy: 0.8257 - val_loss: 0.3947 -
  val_accuracy: 0.8612
Epoch 3/10
2380/2380 [==============================] - 922s 387ms/step -
 loss: 0.3445 - accuracy: 0.9061 - val_loss: 0.1632 -
  val_accuracy: 0.9492
Epoch 4/10
2380/2380 [==============================] - 965s 405ms/step -
 loss: 0.3141 - accuracy: 0.9385 - val_loss: 0.4461 -
  val_accuracy: 0.9244
Epoch 5/10
2380/2380 [==============================] - 939s 395ms/step -
 loss: 0.3047 - accuracy: 0.9550 - val_loss: 0.6312 -
  val_accuracy: 0.9153
Epoch 6/10
2380/2380 [==============================] - 930s 391ms/step -
 loss: 0.2802 - accuracy: 0.9604 - val_loss: 0.3336 -
  val_accuracy: 0.9639
Epoch 7/10
2380/2380 [==============================] - 936s 393ms/step -
 loss: 0.2534 - accuracy: 0.9651 - val_loss: 0.3939 -
  val_accuracy: 0.9503
Epoch 8/10
2380/2380 [==============================] - 946s 398ms/step -
 loss: 0.2722 - accuracy: 0.9632 - val_loss: 0.3459 -
  val_accuracy: 0.9616
Epoch 9/10
2380/2380 [==============================] - 972s 408ms/step -
 loss: 0.2881 - accuracy: 0.9692 - val_loss: 0.3297 -
  val_accuracy: 0.9605
Epoch 10/10
2380/2380 [==============================] - 940s 395ms/step -
 loss: 0.2211 - accuracy: 0.9745 - val_loss: 0.1468 -
  val_accuracy: 0.9808
```

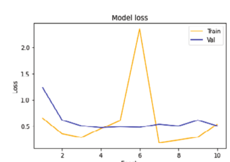
Plotting training accuracy and validation accuracy:

```
plt.plot(model_fit.history['accuracy'])
plt.plot(model_fit.history['val_accuracy'])
plt.title('Model accuracy')
plt.xlabel('Epoch')
plt.ylabel('Accuracy')
plt.legend(['Train','Val'],loc='lower right')
plt.show()
Out:
```



Plotting training loss and validation loss:

```
plt.plot(model_fit.history['loss'])
plt.plot(model_fit.history['val_loss'])
plt.title('Model loss')
plt.xlabel('Epoch')
plt.ylabel('Loss')
plt.legend(['Train','Val'],loc='upper right')
plt.show()
Out:
```



Importing test data from the directory (10% of the remaining data not seen by the model):

```
test=ImageDataGenerator(rescale=1/255)
test_dataset=test.flow_from_directory("trainsplit/test",
   target_size=(227,227))
Out: Found 898 images belonging to 20 classes.
```

Predicting the test dataset:

```
model.evaluate(test_dataset)[1]
Out: 29/29 [==============================]
  - 126s 4s/step - loss: 0.6214 - accuracy: 0.9688
0.9688196182250977
```

Softmax layer output array can be obtained using:

```
X=image.img_to_array(img)
X=np.expand_dims(X,axis=0)
images=np.vstack([X])
val=model.predict(images)
val
Out: 1/1 [==============================] - 5s 5s/step
array([[0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
        0., 0., 1., 0., 0., 0., 0., 0.]], dtype=float32)
```

The above array is then processed to identify the class label to which the leaf belongs to:

```
list_index=[0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,
                16,17,18,19]
x=val
for i in range(20):
   for j in range(20):
       if x[0][list_index[i]]>x[0][list_index[j]]:
           temp=list_index[i]
           list_index[i]=list_index[j]
           list_index[j]=temp
print(list_index)
Out: [14, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12,
         13, 0, 15, 16, 17, 18, 19]
```

References

1. Mukherjee G, Tudu B, Chatterjee A (2021) A convolutional neural network-driven computer vision system toward identification of species and maturity stage of medicinal leaves: case studies with Neem, Tulsi and Kalmegh leaves. Soft Comput 25:14119–14138. https://doi.org/10.1007/s00500-021-06139-9
2. Sladojevic S, Arsenovic M, Anderla A, Culibrk D, Stefanovic D (2016) Deep neural networks based recognition of plant diseases by leaf image classification. Comput Intell Neurosci. https://doi.org/10.1155/2016/3289801
3. Xin M, Wang YX (2019) Research on image classification model based on deep convolution neural network. EURASIP J Image Video Process. https://doi.org/10.1186/s13640-019-0417-8
4. Kschandra R, Srinivas Y, Krishna M (2020) Study on image processing using deep learning techniques. Mater Today Proc. https://doi.org/10.1016/j.matpr.2020.09.536
5. Azadnia R, Al-Amidi MM, Mohammadi H, Cifci MA, Daryab A, Cavallo E (2022) An AI based approach for medicinal plant identification using deep CNN based on global average pooling. Agronomy. https://doi.org/10.3390/agronomy12112723
6. Wu SG, Bao FS, Xu EY, Wang YX, Chang YF, Xiang QL (2007) A leaf recognition algorithm for plant classification using probabilistic neural network. In: 2007 IEEE international symposium on signal processing and information technology, pp 11–16. https://doi.org/10.1109/ISSPIT.2007.4458016.
7. Anami BS, Nandyal SS, Govardhan A (2010) A combined color, texture and edge features based approach for identification and classification of Indian medicinal plants. Int J Comput Appl 6(12):45–51
8. Gopal A, Prudhveeswar RS, Gayatri V (2012) Classification of selected medicinal plants leaf using image processing. In: International conference on machine vision and image processing. https://doi.org/10.1109/MVIP.2012.6428747
9. Kan HX, Jin L, Zhou FL (2017) Classification of medicinal plant leaf image based on multi-feature extraction. Pattern Recognit Image Anal 27:581–587. https://doi.org/10.1134/S105466181703018X
10. Vo AH, Dang HT, Nguyen BT, Pham VH (2019) Vietnamese herbal plant recognition using deep convolutional features. Int J Comput Appl 9(3):363–367
11. Dudi B, Rajesh V (2019) Medicinal plant recognition based on CNN and machine learning. Int J Adv Trends Comput Sci Eng 8(4):999–1003
12. Zin IA, Ibrahim Z, Isa D, Aliman S, Sabri N, Mangshor NN (2020) Herbal plant recognition using deep convolutional neural network. Bull Electr Eng Informat 9(5):2198–2205
13. Anchitaalagammai JV, Shantha Lakshmi Revathy JS, Kavitha S, Murali S (2021) Factors influencing the use of deep learning for medicinal plants recognition. J Phys 2089(1):012055. https://doi.org/10.1088/1742-6596/2089/1/012055
14. Malarvizhi K, Sowmithra M, Gokula Priya D, Kabila B (2021) Machine learning for plant species lassification using leaf vein morphometric. Int J Eng Res Technol https://doi.org/10.17577/IJERTV10IS040273
15. Azadnia R, Al-Amidi MM, Mohammadi H, Cifci MA, Daryab A, Cavallo E (2022) An AI based approach for medicinal plant identification using deep CNN based on global average pooling. Agronomy 12(11):2723. https://doi.org/10.3390/agronomy12112723
16. Islam M, Ria NJ, Ani JF, Masum AKM, Abujar S, Hossain SA (2022) Deep learning based classification system for recognizing local spinach. In: Advances in deep learning, artificial intelligence and robotics (ICDLAIR), https://doi.org/10.1007/978-3-030-85365-5-1
17. Manikanta Sanjay V, Kyalkond SA, Aithal SS, Kumar PS (2022) A novel approach to classification of ayurvedic medicinal plants using neural networks. Int J Eng Res Technol https://doi.org/10.17577/IJERTV11IS010128
18. Suresh V, Krishnan M, Hemavarthini M, Jayanthan K, Gopinath D (2020) Plant disease detection using image processing. Int J Eng Res Technol https://doi.org/10.17577/IJERTV9IS030114
19. Viknesh GS, Arun Yagan (2022) Leaf classification for plant recognition using EfficientNet architecture, Int J Eng Res Technol 10(11):1–5. https://doi.org/10.17577/IJERTV10IS110021
20. Pathirana CK, Madhujith T, Eeswara J (2020) Bael (Aegle marmelos L. Corrêa), a medicinal tree with immense economic potentials. Adv Agric https://doi.org/10.1155/2020/8814018
21. Rai KK, Trivedi RV, Umekar MJ (2019) Review on betel leaf used in various ailments. Int J Pharmacognosy, 6(8):259–267. https://doi.org/10.13040/IJPSR.0975-8232.IJP.6(8).259-67
22. Rani YS, Reddy VJ, Basha SJ, Koshma M, Hanumanthu G, Swaroopa P (2017) A review on Solanum nigrum. World J Pharmacy Pharmaceut Sci, 6:293–303. https://doi.org/10.20959/wjpps201712-10538
23. Akbaribazm M, Khazaei F, Naseri L, Pazhouhi M, Zamanian M, Khazaei M (2021) Pharmacological and therapeutic properties of the Red Clover (Trifolium pratense L.): an overview of the new findings. J Tradit Chinese Med, 41(4): 642. https://doi.org/10.19852/j.cnki.jtcm.20210604.001
24. Balakrishnan R, Vijayraja D, Jo SH, Ganesan P, Su-Kim I, Choi DK (2020) Medicinal profile, phytochemistry, and pharmacological activities of Murraya Koenigii and its primary bioactive compounds. Antioxidants 9(2):101. https://doi.org/10.3390/antiox9020101
25. Wani SA, Kumar P (2018) Fenugreek: a review on its nutraceutical properties and utilization in various food products. J Saudi Soc Agric Sci 17(2):97–106. https://doi.org/10.1016/j.jssas.2016.01.007
26. Kalita S, Kumar G, Karthik L, Rao KVB (2012) A review on medicinal properties of Lantana camara Linn. Res J Pharmacy Technol 5(6):711–715

27. Nayak P, Kumar T, Gupta AK, Joshi NU (2020), Peppermint a medicinal herb and treasure of health: a review. J Pharmacognosy Phytochem, 9(3), 1519–1528. https://doi.org/10.22271/phyto.2020.v9.i3y.11525
28. Subapriya R, Nagini S (2005) Medicinal properties of neem leaves: a review. Curr Med Chem Anti-Cancer Agents 5(2):149–156. https://doi.org/10.2174/1568011053174828
29. Prakash PA, Gupta N (2005) Therapeutic uses of Ocimum sanctum Linn (Tulsi) with a note on eugenol and its pharmacological actions: a short review. Indian J Physiol Pharmacol 49(2):125
30. Wang Y, Li Y, Song Y, Rong X (2020) The influence of the activation function in a convolution neural network model of facial expression recognition. Appl Sci 10(5):1897. https://doi.org/10.3390/app10051897
31. Hao W, Yizhou W, Yaqin L, Zhili S (2020) The role of activation function in CNN. In: International conference on information technology and computer application, pp 429–432. https://doi.org/10.1109/ITCA52113.2020.00096
32. Nwankpa C, Ijomah W, Gachagan A, Marshall S (2018) Activation functions: comparison of trends in practice and research for deep learning. https://doi.org/10.48550/arXiv.1811.03378
33. Yamashita R, Nishio M, Do RKG, Togashi K (2018) Convolutional neural networks: an overview and application in radiology. Insights Imaging 9:611–629. https://doi.org/10.1007/s13244-018-0639-9
34. Basha SHS, Dubey SR, Pulabaigari V, Mukherjee S (2019) Impact of fully connected layers on performance of convolutional neural networks for image classification. Neurocomputing 378:112–119. https://doi.org/10.1016/j.neucom.2019.10.008