Wireless Personal Communications (2024) 136:2275–2298
https://doi.org/10.1007/s11277-024-11374-y

Classification of Different Plant Species Using Deep Learning and Machine Learning Algorithms

Siddharth Singh Chouhan[1] · Uday Pratap Singh[2,3] · Utkarsh Sharma[4] · Sanjeev Jain[5]

Accepted: 14 June 2024 / Published online: 2 July 2024

Abstract

In the present situation, a lot of research has been directed towards the potency of plants. These natural resources contain characteristics valuable in combat against a number of diseases. But due to lack of familiarity of these plants among human beings, an appropriate advantage of their significance cannot be drawn away. Plants also shares the certain similar characteristics of leaves like color, texture, shape or size, making them hard to classify them among others. So, to eradicate this problem, a deep learning model has been used for the purpose for classification of different plants species captured in real-time using internet of things practice. Six different plants namely *Ashwagandha*, *Black Pepper*, *Garlic*, *Ginger*, *Basil*, and *Turmeric* has been selected for this purpose. Our proposed convolutional neural network (CNN) model achieved higher performance with an accuracy of 99% when compared with other benchmark deep learning models. Also, to analyze the performance of deep learning versus machine learning models like logistic regression, decision tree, random forest, Gaussian naïve Bayes, support vector machine results were evaluated and when compared CNN outperforms against all machine learning models. The future study will be directed towards the automated plant growth estimation.

Keywords Computer vision · Convolution neural network · Deep learning · Image classification

✉ Uday Pratap Singh
usinghiitg@gmail.com

[1] School of Computing Science and Engineering, VIT Bhopal University, Kothrikalan, Sehore, Madhya Pradesh 466114, India

[2] School of Mathematics, Shri Mata Vaishno Devi University, Katra, J&K 182320, India

[3] Department of Mathematics, Central University of Jammu, Samba, J&K 181143, India

[4] Department of Computer Science and Engineering, JUET, Guna, Madhya Pradesh 473226, India

[5] Department of Computer Science and Information Technology, Central University of Jammu, Samba, J&K 181143, India

1 Introduction

In the recent times, comparatively a number of research has been focused upon the identification and classification of the plant species that are useful for human beings' health. These plant species are good in their number of characteristics like medicinal usages, food supply, timber or furniture production and many more. Ayurveda, a well-known ancient culture used for the treatment of number of diseases significantly considers almost all plants having medicinal values. Some of them includes Moringa, Neem, Basil, Ashwagandha, Ginger, Garlic, Turmeric, Black cumin, Citrus, Red Bell Peppers, Broccoli, Spinach, Almond, Sunflower, Papaya, Orange, Kiwi, Passley, Strawberry, etc. But due to inter-class and intra-class variations within the families of plant including color, occlusion, background, anatomy sometimes it becomes harder for a normal human being to identify such plants [1–4]. It requires a subject matter expert to identify them and because of this some most important species lack in production. Computer vision (CV) methods along with deep learning (DL) models have been widely used in a number of applications as disease classification, species identification, weed and pest classification, fruit detection, and many more. This amalgamation achieves higher accuracy and are very popular in recent years [4–6].

Deep learning models differ from conventional machine algorithms in which the high-level features are extracted automatically rather than manually. Deep models are also capable enough to overcome the limitations associated with other algorithms with the help of parallel architectures and working with large datasets [8–10]. Gradually, a number of deep learning models are introduced with an increase in the number of layers. Some of these models include AlexNet, GoogLeNet Inception V3, Inception V4, VGG net, Microsoft ResNet, DenseNets. However, the models also suffer from the problem of vanishing gradients, degradation, and internal covariate shift. But these limitations can be overcome by Optimization strategies, layer-wise training, including skip connections, batch Normalization, initialization strategies, transfer learning [11–15]. Advancement in Computer vision and Deep learning approaches has facilitated a way to extend the research and development opportunities in the classification of specific plant images [16–19]. So, the proposed is focused upon the classification of six different plant species that are helpful in boosting immunity. These species are Ashwagandha, Black Pepper, Garlic, Ginger, Basil, and Turmeric. In the proposed work, the amalgamation of computer vision and deep learning approaches along with internet of things methodology has been adopted for the identification of the plant species. Proposed work diagram is given by Fig. 1.

Some of the motivations of the presented work are as follows:

1) To offer a plants classification approach helpful to the community in selecting, understanding, and growing plants helpful in human health resistance.
2) Amalgamation of computer vision and deep learning approaches for the purpose of plants classification using leaves images.
3) Real time collection of leaves images for six different plants namely well-known in Ayurveda.
4) Evaluation and comparison with different deep learning models and machine learning algorithms.

The organization of the manuscript is as follows: The introduction in Sect. 1 is followed by the related studies given by Sect. 2. Section 3, includes a detailed explanation of the materials and methods used. Section 4, illustrates the experimental results, followed by the

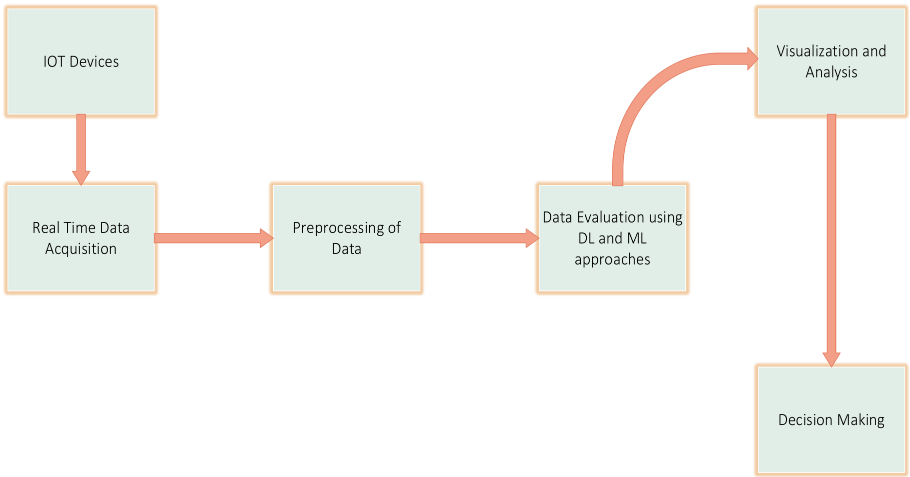


Fig. 1 Proposed work

discussion presented in Sect. 5. Lastly, a conclusion with possible future enhancement is given by Sect. 6, followed by references.

2 Related Work

A number of studies have been directed towards a better understanding of plant anatomy. This includes plant phenotyping, plant classification, plant disease detection, fruit detection, fruit disease detection, weed identification, pest detection, crop yield prediction, plant canopy estimation, and many more. For this study, our focus is directed towards the survey of related works that practice the concepts of computer vision and deep learning with respect to plant pathology. The various studies are given in Table 1.

3 Materials and Methods

The motivation behind this article is to present an approach, which could be appreciated to identify the different plant species that can be useful in enhancing the resistance power of human being to fight against corona virus. Six different plants were selected for this purpose. Then using the state-of-the-art computer vision methods and deep learning models the plant species were classified. The data collection process and the detailed description of deep learning model used in our work is given by following sections.

3.1 Data Collection

The sample leaf images for the different plants were collected physically from different locations with the help of raspberry pi board shown in Fig. 2. These images were then preprocessed for maintaining uniformity among the database. All the leaf images were

Table 1 Literature survey

Author with year of publication	Application goals	Model used	Plant	Dataset details	Accuracy obtained/Analysis
Sinan Uguz and Nese Uysal in 2020 [3]	Leaf disease classification	VGG16 and VGG19	Olive	3400 leaf sample	95%
Sanaz Rasti et al. in 2020 [7]	Crop growth estimation	ConvNets	Wheat and Barley	138,000 images	95.9%
Ramar Ahila Priyadharshini et al., in 2019 [13]	Leaf disease classification	LeNet	Maize	3852 images	97.89%
Arturo Aquino et al., in 2020 [18]	Identification of fruit	AlexNet, VGG19, InceptionV3, ResNet-50, and Inception-ResNetV2	Olive	5, 34, 114 images	98.22%
Arthur Z. da Costa et al., in 2020 [19]	External defect on tomato	Deep residual neural network (ResNet)	Tomato	43,843	91.7%
Leihong Wu et al., in 2019 [20]	Automatic identification of 15 storage product beetle species	Convolutional neural network	Images of elytra fragments	6900 images	83.8%
Sakshi Srivastava et al., in 2020 [21]	Disease Detection	VGG-16, VGG-19 and Inception v3	Sugarcane	240 images	90.2%
Aydin Kaya et al., in 2019 [22]	Plant species classification	Deep Neural Networks	Multiple	57,781 images	92.65%, 99.11%, 70.79, and 98.77%
M. P. Vaishnnave et al., in 2020 [23]	Classification of diseases	Deep convolutional neural network	Groundnut	1800 images	99.88%
Everton Castelao Tetila in 2019 [24]	Classification of leaf diseases	Inception-v3, Resnet-50, VGG-19, and Xception	Soybean	300 images	99.04%
K. Thenmozhi and U. Srinivasulu Reddy in 2019 [25]	Pest classification	AlexNet, ResNet, GoogLeNet and VGGNet	Multiple	NBAIR dataset	96.75%, 97.47%, and 95.97%
Basavaraj S. Anami et al., in 2020 [26]	Recognition and classification of yield	VGG-16 CNN	Paddy crop	30,000 images	92.89%
Jose G.M. Esgario et al., in 2020 [27]	Biotic stress classification and severity estimation	AlexNet, GoogLeNet, VGG16, ResNet50 and MobileNetV2)	Coffee	6092 images	95.24% and 86.51%

Table 1 (continued)

Author with year of publication	Application goals	Model used	Plant	Dataset details	Accuracy obtained/Analysis
Tuan-Tang Le et al. in 2019 [28]	Classification of some horticultural crop	Mask R-CNN	Banana	2621 samples	96.5%
Solemane Coulibaly et al. in 2019 [29]	Identification of disease	VGG16	Pearl millet	711 images	95.0%
Hanwen Kang and Chao Chen in 2020 [30]	Fruit detector	RCNN	Apple	800 images	85.3%
Hubert Cecotti et al. in 2020 [31]	Fruit detection	Multiple CNN architectures	Grapes	1104 images	99.0%
Qiufeng Wu et al. in 2019 [32]	Leaf disease identification	ResNet	Soybean	1470 images	94.29%
Shuxiang Fan et al. in 2020 [33]	Disease detection	CNN	Apple	300 samples	96.5%
Aditya Khamparia et al. in 2019 [34]	Disease prediction and classification	Convolutional neural network (CNN)	Multiple crops	900 images	97.50% and 100%
Xiaolong Zhu et al., in 2018 [35]	Identification of plant species	RCNN	Multiple	900 images	99%
Guillermo L. Grinblat et al. in 2016 [36]	Plant identification	Deep convolutional neural network	White bean, red bean and soybean	866 images	96.9%
Mostafa Mehdipour Ghazi et al. in 2017 [37]	Plant identification	GoogLeNet, AlexNet, and VGGNet	Multiple	91,758 images	80.0%
A. Kumar and S. Sachar in 2024 [38]	Study on Segmentation and classification of plants	CNN, Mask-RCNN, RNN,	Multiple	Multiple	DL for generation of synthetic leaf images
S. B. Jadhav and S. B. Patil in 2024 [39]	Identification and classification of plant species	Local binary histogram pattern of gradient (LBHPG)	Multiple	PlantVillage	94.58%
M. Manaouch et al., in 2023 [40]	Predicting reforestation areas	Multiple ML algorithms	Quercus ilex	Self-acquired data	98%

Table 1 (continued)

Author with year of publication	Application goals	Model used	Plant	Dataset details	Accuracy obtained/Analysis
T. Meenakshi in 2023 [41]	Disease detection in medicinal plant species	GLCM + LR	Multiple	Multiple	NA
Singh et al. in 2023 [42]	Detection of leaf disease	GLCM + RF	Paddy	Multiple	99.10%
J. G. Thanikkal et al., in 2023 [43]	Medicinal plant identification	SDAMPI	Multiple	Multiple	96%
M. Tholkapiyan et al., in 2023 [44]	Disease identification and classifcation	ML + DL	Rice	NA	ML and DL methods were presented
H. M Hama et al., in 2024 [45]	Houseplant leaf classification	ResNet-50	House plant leaves	2500 images	99%
K. Kayaalp in 2024 [46]	Classification of sherry species	DenseNet169	Cherry	3570 images	99.57%
T. Varma et al., in 2024 [47]	Leaf disease detection	InceptionV3	Mango	NA	99.87%

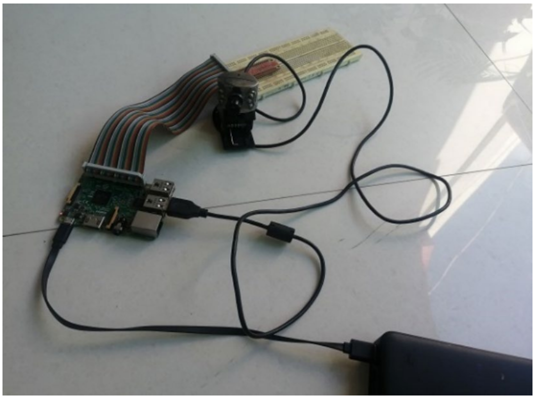


Fig. 2 Setup used for image acquisition

classified among six different classes and stored respectively. The labeling has been done accordingly ranging from A0 to A5 for the plants as *Ashwagandha* (P0), *Black Pepper* (P1), *Garlic* (P2), *Ginger* (P3), *Basil* (P4), and *Turmeric* (P5). A total of 1265 images were acquired for this work (the data will be available at https://data.mendeley.com/datasets/hb74ynkjcn/4).

3.2 Data Augmentation

Training a deep learning models with an abundant amount of data, increases the probability of a model to be well generalized and therefore reducing the chance of over fitting. A very diverse property of a deep learning model is that, it can classify an object in different orientations, sizes, illuminations, or shearing's, this methodology allows academicians to work even with a small dataset. Image augmentation is the process of producing numerous images from the existing one, therefore expanding the existing small dataset to a large one [48]. An image can be transformed in number of manners including flipping, rotating, zooming in or out, adjustment in contrast or brightness among others. Augmentation mostly is a pre-processing step, but it can be applied in real time as well. Some basic augmentation techniques are (a) Rotating, (b) Flipping, (c) Shearing, (d) Cropping, (e) Zoom in or out, and (f) Change in brightness or contrast. There are number of image augmentation libraries available with the tools like python, *scikit-image*, *augmentor*, *albumentations*, *torchvision*, *imgaug*, and *openCV* are some of them. In this manuscript python library *imgaug* has been used for the following transformation [48]:

a. Rotation
b. Adding noise
c. Cropping
d. Shearing
e. Flipping
f. Brightness adjustment
g. Scaling

Figure 3 shows the augmented images. Because of this process our dataset size was increased from 1265 images to 30,360 images (1265*24).

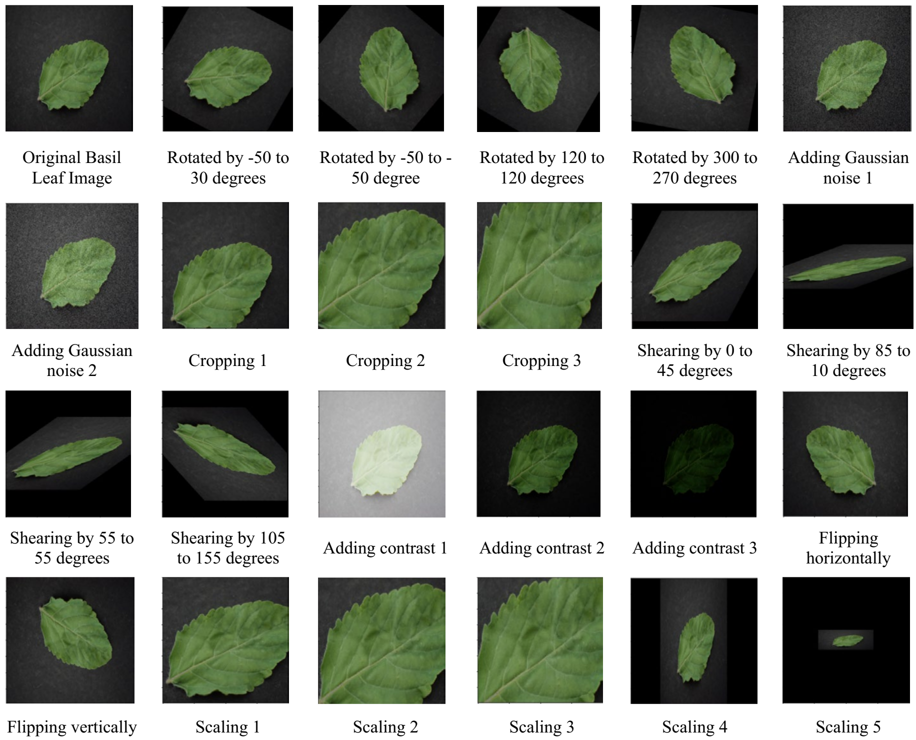


Fig. 3 Augmented images

3.3 Model Description

The CNN model which we use for the classification purpose has the following specifications. Normally, a CNN comprises four different layers that help with extracting data from a picture e.g., fully associated layer, ReLU layer, Pooling layer, and convolution layer [17]. In the model proposed, the input to the CNN is an image of a medicinal plant leaf, with those images it makes some disfigured pictures, and then categorize them into different classes. CNN perceives pictures as a network of numbers related to separate pixels with help of kernels. The convolution layer uses few filters for features evaluation that play out the convolution activity. These features of the filters will look at two little bits of greater pictures in the event that it matches, at that point pictures will be categorized accurately. There are four stages that will be needed in this layer, the feature filter is needed to be lined up in the image first and afterward duplicate each picture pixel by the related feature pixel [12, 17]. After that include the qualities and figure aggregate that will be isolated by the complete number of pixels in the component. The last watched value is set at the focal point of the separated picture. Likewise, include channel moves all through the picture, wherever on the picture, and rehash the same past steps. For each filter of the feature, this cycle is rehashed to get the convolution yield (maps for features are extricated).

The rectified linear unit layer is the other layer which is an actuation function that will actuate a node, if the inputs over the specific amount are under zero the yield is likewise zero. At the point when the information transcends the specific edge, it has a straight relationship with the variable which is dependent. That same function eliminates all the negative qualities and gets converted to zero from the convolution [49–51]. It is applied to all the image feature pairs to produce the yield (corrected component map). Another sort of actuation work is sigmoid, spoken to as S-shape and it is predominantly utilized where output is anticipated as the probability, implies yield exits between the scope of 0 and 1 [17].

Last is the pooling layer, which diminishes the cardinality of the feature maps or compresses the pictures. It applies on images that are filtered subsequent to going through the initial layer of activation. A few stages are generally remembered for Pooling. At first, select the size of a window (3×3 or 2×2), and this will move over the image which was filtered, and then most extreme values will be taken from every window. The last feature map which is pooled is made flat and taken care of to a completely connected layer where the final classification occurs and predicts the last yield. The Network with completely connected nature utilizes activation function as softmax (appeared in Eq.), it is a somewhat regression of logistic nature that follows a distribution of probability and standardizes the value of the input into a vector of qualities [49, 52]. The yield esteems are between the reach [0, 1]. It is otherwise called the most extreme entropy classifier.

To conquer the issues of overfitting the model uses the dropout layers. In the field of machine learning, overfitting is reduced by introducing loss functions with a penalty, similarly, a dropout layer is used in CNN for regulation in which arbitrarily dropping of neurons is performed [52–54]. This implies that their commitment to the initiation of downstream neurons is transiently taken out on the forward pass and any weight refreshes are not applied to the neuron on the retrogressive pass. Figure 4, shows the CNN models used for classification [17] purpose, followed by the layer's configuration details and the input–output combinations defined separately.

3.3.1 Convolution Layers

Suppose the input of the convolution layers has dimension $W \times H \times D$, then the convolutional layers are set parallel feature map by strides the kernels of different size over an input and projecting pixels as features maps. If kernel of the size $K \times K \times D$ and assume that there is stride S, which representing the sliding portion and P is zero padding parameter used for control the dimension and feature maps using kernels, then the size of output says $W_1 \times H_1 \times D_1$ of such convolutional layer will be:

$$W_1 = \frac{W - K + 2P}{S} + 1$$

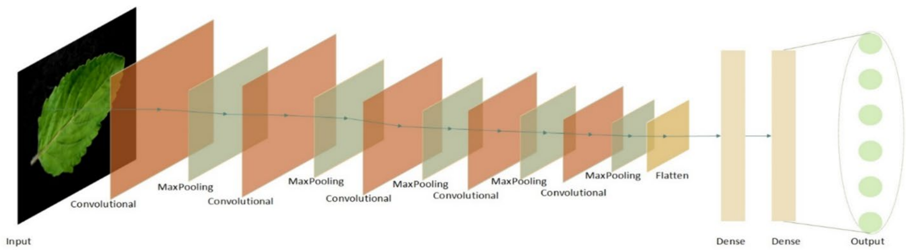


Fig. 4 CNN model used for the classification of immunity boosting plant images [14]

$$H_1 = \frac{H - K + 2P}{S} + 1$$

$$D_1 = D$$

3.3.2 Activation Function

It defines the output of neuron based on given set of the inputs. Weighted sum of linear net input value is passed through an activation function for non-linear transformation. We can use some common activation function as.

a. Sigmoid: $(x) = \frac{1}{1+e^{-x}}$; and $\frac{df}{dx} = f(x)(1 - f(x))$.
b. Tanh: $(x) = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}}$; and $\frac{df}{dx} = 1 - (f(x))^2$.
c. ReLU: $(x) = \begin{cases} x : x \geq 0 \\ 0 : x < 0 \end{cases}$; $f'(x) = \begin{cases} 1 : x \geq 0 \\ 0 : x < 0 \end{cases}$
d. Softmax: $\left(x_i\right) = \frac{e^{x_i}}{\sum_{i=1}^{n} e^{x_j}}$; $\frac{\delta f\left(x_i\right)}{\delta x_j} = f\left(x_i\right)\left(\delta_{ji} - f\left(x_i\right)\right)$.

3.3.3 Pooling Layer

If the kernel size is K, the pooling layers are down sampling layer combined output of the layer to a single neuron. Let D_n as number of kernel windows and S as Stride develop pooling layers, then the size of output $W_1 \times H_1 \times D_1$, dimension of the pooling layer will be for input of size $W \times H \times D$:

$$W_1 = \frac{W - K}{S} + 1$$

$$H_1 = \frac{H - K}{S} + 1$$

$$D_1 = D_n$$

3.3.4 Fully Connected Layer

After the poling layers, pixels of the pooling layers is structured to single column vector. The vectorized and concatenated data points are fed into dense layers, known as fully connected layers.

3.3.5 Backpropagation

Backward propagation of errors which use gradient descent to compute the gradient of the loss function with respect to the parameters like weight and bias. The diagram is given by Fig. 5.

3.3.5.1 Process of Backpropagation in Fully Connected Layer In the Backpropagation such as W^{L+1}, b^{L+1}, W^l, b^l, $K^{p,q}$, and $b^{p,q}$ are required in order to minimize the loss func-

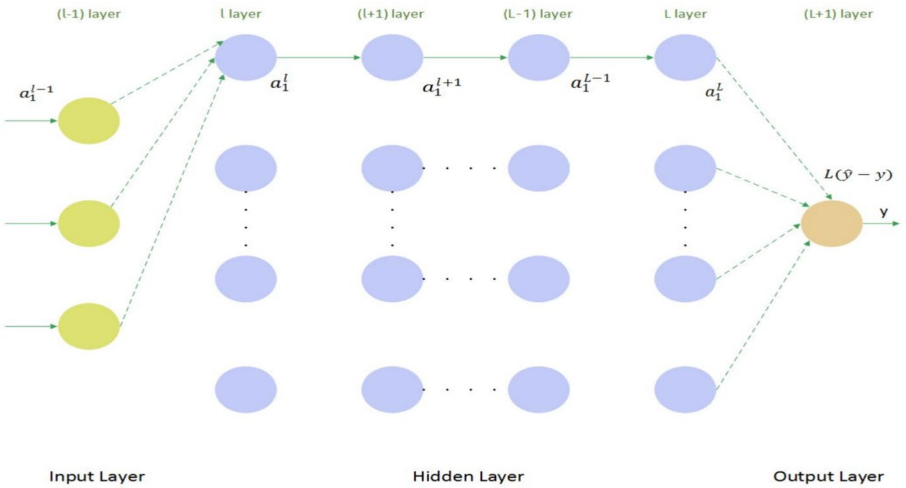


Fig. 5 Backpropagation network

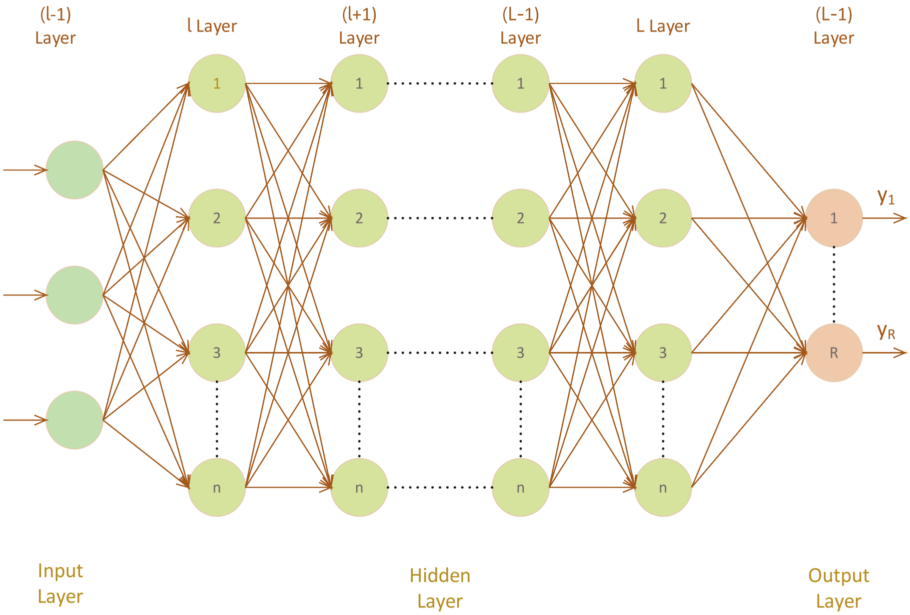


Fig. 6 Forward run in fully connected layer

tion as shown in Figs. 5 and 6. During the Backpropagation, gradient of loss function of final layers with respect to the parameters is computed first whereas the gradient of first layer is computed in last. Also derivative of one layer is reused by chain rule in cross entropy loss function.

$$\begin{aligned}\frac{\delta L\left(\hat{y}^{L+1}, y\right)}{\delta \hat{y}_i^{L+1}} &= \frac{1}{N}\sum_{j=1}^{N} - \frac{\delta\left[y_j \log\left(\hat{y}_j^{L+1}\right) + \left(1-y_j\right)\log\left(1-\hat{y}_j^{L+1}\right)\right]}{\delta \hat{y}_i^{L+1}} \\ &= \frac{1}{N}\left(-\frac{y_i}{\hat{y}_i^{L+1}} + \frac{1-y_i}{1-\hat{y}_i^{L+1}}\right) \\ &= \frac{1}{N}\left(\frac{-y_v}{\sigma\left(z_u^L\right)} + \frac{1-y_v}{1-\sigma\left(z_u^L\right)}\right)\sigma\left(z_u^L\right)\left(1-\sigma\left(z_u^L\right)\right)a_u^{L-1} \\ &= \frac{1}{N}\left(\sigma\left(z_u^L\right) - y_L\right)a_u^{L-1}\end{aligned} \tag{1}$$

Similarly, derivative of loss function with respect to bias in u^{th} neuron L^{th} layer is

$$\begin{aligned}\frac{\delta L\left(\hat{y}^{L+1}, y\right)}{\delta b_i^L} &= \sum_{j=1}^{N} \frac{\delta L\left(\hat{y}^{L+1}, y\right)}{\delta\sigma\left(z_j^L\right)} \frac{\delta\sigma\left(z_j^L\right)}{\delta z_j^L} \frac{\delta z_j^L}{\delta b_i^L} \\ &= \frac{1}{N}\left[\sigma\left(z_i^L\right) - y^i\right]\end{aligned}$$

To perform the learning of convolution network, it is necessary to update the kernel bank weights and bias values in the convolution layers as well as in pooling layers.

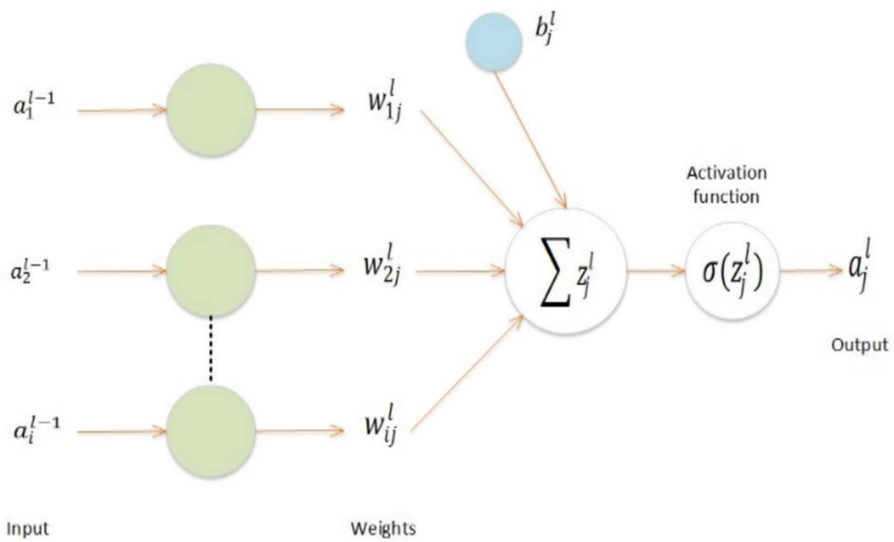


Fig. 7 Loss minimize function using gradient descent rule

3.3.5.2 Parameter Updates To minimize the loss function, we used gradient descent rule as shown in Fig. 7. The following update rule is applied on weight, bias and kernel:

$$W_{new}^{l} = W_{old}^{l} - \propto \frac{\delta L\left(\hat{y}^{L+1}, y\right)}{\delta W^{L}}$$

$$b_{new}^{p,q} = b_{old}^{p,q} - \propto \frac{\delta L\left(\hat{y}^{L+1}, y\right)}{\delta b^{p,q}}$$

$$K^{p,q} = K^{p,q} - \propto \frac{\delta L\left(\hat{y}^{L+1}, y\right)}{\delta K_{uv}^{p,q}}$$

where $\propto$ is learning rule.

For multiclass classification, the loss function of classification layer $L+1$ is:

$$\begin{bmatrix} \frac{\delta L(\hat{y}^{L+1}, y)}{\delta \hat{y}_1^{L+1}} \\ \frac{\delta L(\hat{y}^{L+1}, y)}{\delta \hat{y}_2^{L+1}} \\ \cdot \\ \cdot \\ \cdot \\ \frac{\delta L(\hat{y}^{L+1}, y)}{\delta \hat{y}_i^{L+1}} \\ \cdot \\ \cdot \\ \cdot \end{bmatrix} = \frac{1}{N} \begin{bmatrix} -\frac{y_1}{\hat{y}_1^{L+1}} + \frac{1-y_1}{1-\hat{y}_1^{L+1}} \\ -\frac{y_2}{\hat{y}_2^{L+1}} + \frac{1-y_2}{1-\hat{y}_2^{L+1}} \\ \cdot \\ \cdot \\ \cdot \\ -\frac{y_i}{\hat{y}_i^{L+1}} + \frac{1-y_i}{1-\hat{y}_i^{L+1}} \\ \cdot \\ \cdot \\ \cdot \end{bmatrix} \tag{2}$$

Now, calculate derivative with respect to weight W_{uv}^{L} in final layer, which is the L^{th} layer using the chain rule

$$\begin{aligned} \frac{\delta L\left(\hat{y}^{L+1}, y\right)}{\delta W_{uv}^{L}} &= \sum_{j=1}^{N} \frac{\delta L\left(\hat{y}^{L+1}, y\right)}{\delta \hat{y}_i^{L+1}} \frac{\delta \hat{y}_i^{L+1}}{\delta W_{uv}^{L}} \\ &= \frac{1}{N} \sum_{j=1}^{N} \left(-\frac{y_i}{\hat{y}_j^{L+1}} + \frac{1-y_i}{1-\hat{y}_j^{L+1}} \right) \frac{\delta \sigma\left(z_j^L\right)}{\delta W_{uv}^{L}} \\ &= \frac{1}{N} \sum_{j=1}^{N} \left(-\frac{y_i}{\hat{y}_j^{L+1}} + \frac{1-y_j}{1-\hat{y}_j^{L+1}} \right) \frac{\delta \sigma\left(z_j^L\right)}{\delta z_j^L} \frac{\delta z_j^L}{\delta W_{uv}^{L}} \end{aligned} \tag{3}$$

If sigmoidal activation function is used for non-linear transformation

$$\sigma\left(z_i^L\right) = \frac{1}{1 + \exp\left(z_i^L\right)} \tag{4}$$

And

$$z_i^L = \sum_{i=1}^{n} W_{ij}^L a_i^{L-1} b_i^L \tag{5}$$

Using Eq. 4 and 5 we have

$$\frac{\delta L\left(\hat{y}^{L+1}, y\right)}{\delta W_{uv}^L} = \frac{1}{N}\sum_{j=1}^{N}\left(-\frac{y_i}{\sigma\left(z_j^L\right)} + \frac{1-y_i}{1-\sigma\left(z_j^L\right)}\right)\sigma\left(z_j^{i+1}\right)\left(1-\sigma\left(z_j^{i+1}\right)\right) * \delta jva_u^{L-1}$$

In the similar process, the output value of last layer L is

$$a^L = \sigma\left(\left(W^L\right)^T a^{L-1} + b^L\right) = \sigma\left(z^L\right)$$

Expanding this to classification layers, final output predicted value $\hat{y}_i^{L+1} = \sigma(\left(W^L\right)^T \ldots .\sigma(\left(W^2\right)^T\left(\sigma\left(\left(W^1\right)^T a^1 + b^1\right) + b^2\right) + \ldots + b^2$

Where $\hat{y}_i^{L+1}$ is calculated value and y_i is actual value then the performance of model can be measured by the loss function.

These feature maps are passed through a non-linear activation function σ as:

$$C_{m,n}^{p,q} = \sigma\left(\sum_{m=1}^{K}\sum_{n=1}^{K} I_{m-u,n-v} K_{u,v}^{p,q} + b^{p,q}\right)$$

where σ is a ReLU activation function.

Max pooling layer can be calculated as

$$P_{m,n}^{p,q} = \max\left(C_{m,n}^{p,q}\right)$$

This pooling layer $P^{p,q}$ is concatenated to form a long vector with the length $p \times q$ and feed to fully connected layers, the data points $a_i^{l-i} = f(P^{p,q})$

This vector feed to a fully connected layers from l^{th} layer to $(L+1)^{th}$ layer e.g. If the fully connected layers are developed with L no. of layers and n number of neurons, then l is the first layer. Forward run between the layers is given by:

$$\begin{bmatrix} z_1^l \\ \cdot \\ \cdot \\ z_i^l \\ \cdot \\ \cdot \end{bmatrix} = \begin{bmatrix} w_{11}^l w_{12}^l \ldots .w_{1n}^l \\ \ldots.. \\ \ldots.. \\ w_{i1}^l w_{i2}^l \ldots .w_{in}^l \\ \ldots.. \\ \ldots.. \end{bmatrix}\begin{bmatrix} a_1^{l-1} \\ \cdot \\ \cdot \\ a_i^{l-1} \\ \cdot \\ \cdot \end{bmatrix} + \begin{bmatrix} b_1^l \\ \cdot \\ \cdot \\ b_i^l \\ \cdot \\ \cdot \end{bmatrix} \tag{6}$$

$$z^l = \left(W^l\right)^T a^{l-1} + b^l \tag{7}$$

Consider one single neuron in a fully connected layer l, then

$$a_j^l = \sigma\left(\sum_{i=1}^{n} W_{ij}^l a_j^{l-1} + b_i^l\right)$$

and

$$a^l = \sigma\Big(\left(W^l\right)^T a^{l-1} + b^l \Big) = \sigma\left(z^l\right)$$

where

$$\left(a^l\right)^T = \left[\sigma\left(z_1^l\right) \ldots ..\sigma\left(z_2^l\right), \ldots \sigma\left(z_i^l\right), ..\right]$$

3.4 Loss Function

It maps an event of one or more variable onto a real number associated with some cost. Loss function is used to measure the performance of the model and consistency between actual y_i and predicted value $\hat{y}_i^{L+1}$. Performance of the model is good if the loss function value decreases.

If the input vector $x = [x_1, x_2, ..., x_n]$ and output $y = [y_1, y_2, ..., y_K]$ then the mapping of x_i and y_i is given by:

$$\hat{y}_i^{L+1} = f(\sigma(x), w, b)$$

and define loss function:

$$L\left(\hat{y}^{L+1}, y\right) = \frac{1}{N} \sum_{i=1}^{N} y_i, f(\sigma(x), w, b)$$

where L is the loss function, σ is an activation function, w is the weught parameter, and b is the bias term and $\langle . \rangle$ gives the difference between true y_i and output given by proposed model.

Mean square *error:*

$$L\left(\hat{y}^{L+1}, y\right) = \frac{1}{N} \sum_{i=1}^{N} \left(\hat{y}_i^{L+1} - y_i\right)^2$$

Mean square logarithmic error:

$$L\left(\hat{y}^{L+1}, y\right) = \frac{1}{N} \sum_{i=1}^{N} (log\left(y_i + 1\right) - log\left(\hat{y}_i^{L+1} + 1\right))^2$$

L_2 function:

$$L\left(\hat{y}^{L+1}, y\right) = \sum_{i=1}^{N} \left(y_i - \hat{y}_i^{L+1}\right)^2$$

L_1 function:

$$L\left(\hat{y}^{L+1}, y\right) = \sum_{i=1}^{N} \left|y_i - \hat{y}_i^{L+1}\right|$$

3.5 Cross Entropy

If the probability that the output y_i is in the training set label $\widehat{y}_i^{L+1}$:

$P\left(y_i|z_i^{l-1}\right) = \widehat{y}_i^{L+1} = 1$ and the probability that output y_i is not in the training set label $\widehat{y}_i^{L+1}$ is:

$$P\left(y_i|z_i^{l-1}\right) = \hat{y}_i^{L+1} = 0$$

The expected label is y, then

$$P\left(y_i|z_i^{l-1}\right) = \left(\hat{y}_i^{L+1}\right)^{y_i}\left(1 - \hat{y}_i^{L+1}\right)^{(1-y_i)}$$

To maximize the likelihood, which is equivalent to minimize

$$-logP\left(y_i|z_i^{l-1}\right) = -log\left[\left(\hat{y}_i^{L+1}\right)^{y_i}\left(1 - \hat{y}_i^{L+1}\right)^{(1-y_i)}\right]$$

For N training samples, the cost function of cross entropy is:

$$L\left(\hat{y}^{L+1}, y\right) = -\frac{1}{N}\sum_{i=1}^{N} y_i log\left(\hat{y}_i^{L+1}\right) + \left(1 - y_i\right)log\left(1 - \hat{y}_i^{L+1}\right)$$

3.6 Learning of CNN

Let the input and kernel of size or of dimension $(W \times H \times D)$ and $K \times K \times D$ respectively.

Convolution operation over multi-dimensional tensor can be written as:

$$(I \otimes K)_{ij} = \sum_{m=1}^{K}\sum_{n=1}^{K}\sum_{c=1}^{C} K_{m,n,c} I_{i+m,j+n,c}$$

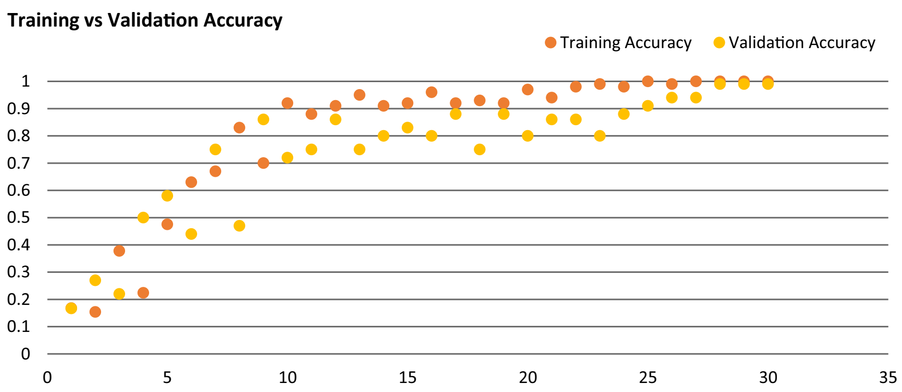
If a kernel bank $K_{u,v}^{p,q}$ convolved with image $I_{m,n}$ with stride value of 1 and zero padding value of 0, the feature maps of the convolution layer $C_{m,n}^{p,q}$ can be computed by

$$C_{m,n}^{p,q} = \sum_{m=1}^{K}\sum_{n=1}^{K} I_{m-u,n-u} K_{u,v}^{p,q} + b^{p,q}$$

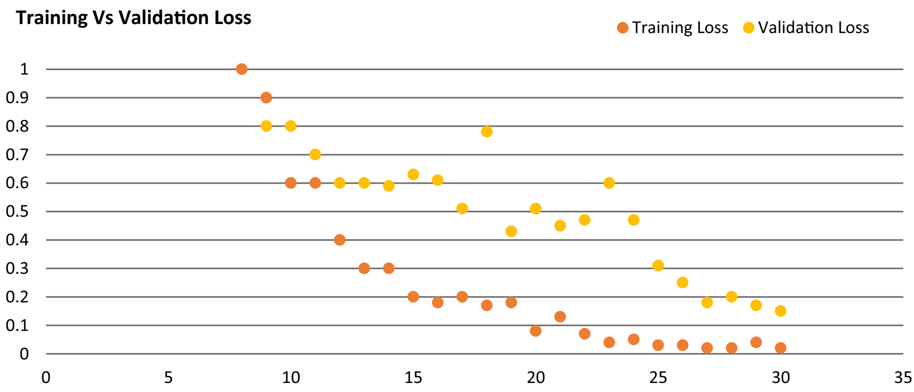
where $b^{p,q}$ are bias term.

4 Results

The proposed model uses a deep convolutional neural network for the classification of the plant species, the results were computed on different epoch counts. Figure 7a represents the comparison of training accuracy vs the validation accuracy for our model. For validation, we used the cross-validation method with 25% of data used for validation and 75% data used for training. The trade-off behind selecting the number of epochs is that as we increase the number of epochs the accuracy tends to increase but as we can see from Fig. 8a that after epoch count 22 the accuracy curve goes flat; we calculated the accuracy



(a)



(b)

Fig. 8 Model **a** accuracy and **b** loss comparison with respect to the epoch count

up to 30 epochs count only. The loss curve for training is also shown in Fig. 8b, and it also shows the same pattern as of accuracy curve as it goes flat after epoch count 22. The model achieves an accuracy of 99% in training.

The classification accuracy for each and every plant species is obtained with the help of the confusion matrix. The confusion matrix is epitomized in the form of a heat map where a dark region represents a low accuracy and a light shade represents a high accuracy. All the plants were accurately classified from one another with full accuracy excluding Garlic (P2) which has an accuracy of 93%.

On the basis of the parameters of the confusion matrix a detailed description of accuracy, precision, recall, and f1-score are shown in Table 2. In this table, we also represented the overall accuracy and support count for the proposed model.

Further, to evaluate the performance of the proposed methods, five different existing deep learning models has been considered from the literature. The results have been validated for classification accuracy and are given by Table 3, proves that the proposed model results in higher performance.

Table 2 Accuracy parameters for classification of plants

Plants/Evaluation parameters	Precision	Recall	F1-Score	Support
Ashwagandha (P0)	1.00	1.00	1.00	30
Black pepper (P1)	0.97	1.00	0.98	30
Garlic (P2)	1.00	0.97	0.98	29
Ginger (P3)	1.00	1.00	1.00	30
Basil (P4)	1.00	1.00	1.00	30
Turmeric (P5)	1.00	1.00	1.00	30
Accuracy	0.99	0.99	0.99	179
Macro average	0.99	0.99	0.99	179
Weighted average	0.99	0.99	0.99	179

Table 3 Performance comparison of proposed model with existing methods

Methods	Accuracy (%)
MCNN [11]	97.23
CNN [16]	96.59
Mask R-CNN [27]	94.35
RCNN [34]	98.14
Proposed CNN	99.00

Table 4 Performance comparison of machine learning algorithms [1]

Methods	Accuracy ± S.D
Logistic regression (LR)	0.93 ± 0.0421
Decision tree (DT)	0.81 ± 0.1038
Random forest (RF)	0.95 ± 0.0358
Gaussian naïve Bayes (NB)	0.71 ± 0.12
Support vector machine (SVM)	0.94 ± 0.0466

4.1 Deep Learning versus Machine Learning

We also applied the machine learning techniques conferred in [1] for the classification of the leaf images. Then we compared the performance of the best machine learning technique with the proposed deep learning-based model, the comparison result is shown in Table 4. The machine learning algorithms we used to classify the images were the Decision Tree, Gaussian Naive Bayes, Support Vector Machine, Random Forest, and Logistic Regression. The performance of the random forest algorithm found to be best among all of the machine learning algorithms tested with an accuracy score of 95.3% (std. deviation of 0.0358). Still, the accuracy of the proposed deep learning model (99% on the training set) results to be better than the random forest algorithm. A comparative graph is given by Fig. 9.

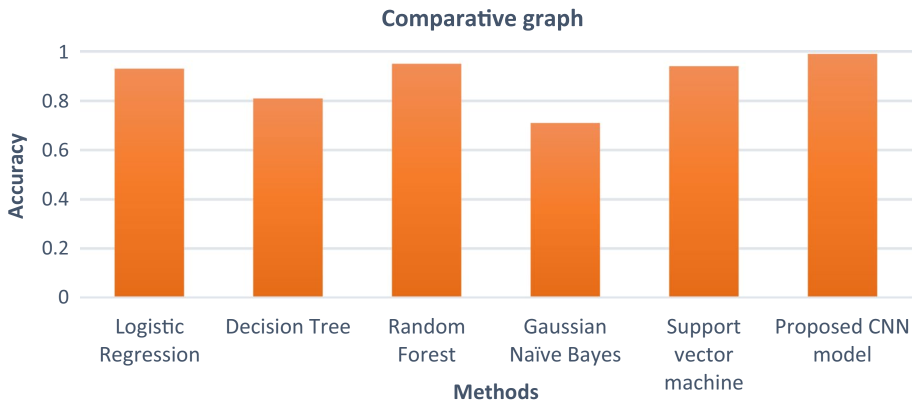


Fig. 9 Comparative graph for different ML methods and proposed CNN

5 Discussion

In this work, the proposed framework operates on the dataset consisting of images for 6 different plants used for medicinal purposes. As the size of the dataset is small, images were augmented first, and then the cross-validation technique was applied with 75% of the data being used for the training purpose and 25% of the data used for testing. The model adapts to the features of the different plant species so that it can further identify the category of any unidentified plant. Even if the model achieves an accuracy of about 99% in the training still it is not intended to replace the human intervention in classifying the correct category of a plant. But it may still provide a quick and reliable method to do the same. Also, the research has to be extended in identifying and distinguishing the plants that are useful for medicinal usages. Further prospects will include image acquisition by using IOT devices, sensors, drones, etc.

6 Conclusion

Ayurveda is among the oldest medical healing sciences that is being heavily practiced in the Indian subcontinent. Plants with medicinal benefits are the core part of this system. In the present time, because of an imbalance in the environmental conditions, a lot of distress can be experienced in plants. Therefore, instant problem-solving will help attain precise data/information for timely diagnosing and treating the plants for their distress. Artificial Intelligence techniques like Machine Learning, Computer Vision, and Deep learning methodologies has outperform the traditional state-of-the-art practices in number of applications. An amalgamation of these methodologies delivers higher efficiency, consequently, in this manuscript, we propose such an approach for real-time classification of different plant species. For this purpose, we have opted for six well-known medicinal plants *Ashwagandha*, *Black Pepper*, *Garlic*, *Ginger*, *Basil*, and *Turmeric*. Further, we propose a convolutional neural network for classifying the plants based on their leaf images. Results proved that our model accomplish higher accuracy of 99% when compared with other deep learning models. Further, to extend the comparative analysis we have also evaluated results with machine learning algorithms like LR, DT, RF, NB, and SVM proves the supremacy of the proposed work. Timely plant growth monitoring is an essential task in agriculture, so our future work will be directed towards this domain.

Acknowledgment This work was supported by grant No. 8-68/FDC/RPS (POLICY-1/2019/20) from the All-India Council of Technical Education (AICTE), India.

Author contributions Author A.B. Conceptualization, investigation, Author A.B.C Writing—original draft preparation, Author A.C. Writing—Reviewing, Author A. B. Editing, methodology, Author D. Supervision.

Funding Not applicable.

Data availability Not applicable.

Declarations

Conflict of interest Not applicable.

Ethical Approval Not applicable.

References

1. Chouhan, S. S., et al. (2021). Leaf disease segmentation and classification of *Jatropha curcas* L. and *Pongamia* pinnata L. biofuel plants using computer vision based approaches. *Measurement, 171*, 108796. https://doi.org/10.1016/j.measurement.2020.108796
2. Srivastava, A., & Das, D. K. (2022). A comprehensive review on the application of internet of thing (IoT) in smart agriculture. *Wireless Personal Communications, 122*, 1807–1837. https://doi.org/10.1007/s11277-021-08970-7
3. Uguz, S., & Uysal, N. (2020). Classification of olive leaf diseases using deep convolutional neural networks. *Neural Computing and Applications*. https://doi.org/10.1007/s00521-020-05235-5
4. Jearanaiwongkul, W., et al. (2019). A semantic-based framework for rice plant disease management. *New Gener Comput, 37*, 499–523. https://doi.org/10.1007/s00354-019-00072-0
5. Chouhan, S. S., Singh, U. P., & Jain, S. (2021). Automated plant leaf disease detection and classification using fuzzy based function network. *Wireless Personal Communications, 121*, 1757–1779. https://doi.org/10.1007/s11277-021-08734-3
6. Too, E. C., et al. (2018). A comparative study of fine-tuning deep learning models for plant disease identification. *Computers and Electronics in Agriculture*. https://doi.org/10.1016/j.compag.2018.03.032
7. Rasti, S., et al. (2020). Crop growth stage estimation prior to canopy closure using deep learning algorithms. *Neural Computing and Applications*. https://doi.org/10.1007/s00521-020-05064-6
8. Qiao, X., et al. (2019). Underwater sea cucumber identification based on principal component analysis and support vector machine. *Measurement, 133*, 444–455. https://doi.org/10.1016/j.measurement.2018.10.039
9. Chouhan, S. S., et al. (2020). Applications of computer vision in plant pathology: a survey. *Archives of Computational Methods in Engineering, 27*, 611–632. https://doi.org/10.1007/s11831-019-09324-0
10. Sabzi, S., et al. (2018). Using video processing to classify potato plant and three types of weed using hybrid of artificial neural network and particle swarm algorithm. *Measurement, 126*, 22–36. https://doi.org/10.1016/j.measurement.2018.05.037
11. Kamilaris, A., et al. (2018). Deep learning in agriculture: A survey. *Computers and Electronics in Agriculture, 147*, 70–90. https://doi.org/10.1016/j.compag.2018.02.016
12. Singh, U. P., et al. (2019). Multilayer convolution neural network for the classification of mango leaves infected by anthracnose disease. *IEEE Access, 7*, 43721–43729. https://doi.org/10.1109/ACCESS.2019.2907383
13. Ahila Priyadharshini, R., et al. (2019). Maize leaf disease classification using deep convolutional neural networks. *Neural Computing and Applications*. https://doi.org/10.1007/s00521-019-04228-3

14. Chouhan, S. S., et al. (2018). Bacterial foraging optimization based radial basis function neural network (BRBFNN) for identification and classification of plant leaf diseases: An automatic approach towards plant pathology. *IEEE Access, 6*, 8852–8863. https://doi.org/10.1109/ACCESS.2018.2800685
15. Bhojani, S. H., & Bhatt, N. (2020). Wheat crop yield prediction using new activation functions in neural network. *Neural Computing and Applications*. https://doi.org/10.1007/s00521-020-04797-8
16. Chouhan, S. S., et al. (2020). Web facilitated anthracnose disease segmentation from the leaf of mango tree using radial basis function (RBF) neural network. *Wireless Personal Communications*. https://doi.org/10.1007/s11277-020-07279-1
17. Bisen, D. (2020). Deep convolutional neural network based plant species recognition through features of leaf. *Multimedia Tools and Applications*. https://doi.org/10.1007/s11042-020-10038-w
18. Aquino, A., et al. (2020). Identification of olive fruit, in intensive olive orchards, by means of its morphological structure using convolutional neural networks. *Computers and Electronics in Agriculture, 176*, 105616. https://doi.org/10.1016/j.compag.2020.105616
19. da Costa, A. Z., et al. (2020). Computer vision based detection of external defects on tomatoes using deep learning. *Biosystems Engineering, 190*, 131–144. https://doi.org/10.1016/j.biosystemseng.2019.12.003
20. Wu, L., et al. (2019). A deep learning model to recognize food contaminating beetle species based on elytra fragments. *Computers and Electronics in Agriculture*. https://doi.org/10.1016/j.compag.2019.105002
21. Srivastava, S., et al. (2020). A novel deep learning framework approach for sugarcane disease detection. *SN Computer Science*. https://doi.org/10.1007/s42979-020-0094-9
22. Kaya, A., et al. (2019). Analysis of transfer learning for deep neural network based plant classification models. *Computers and Electronics in Agriculture, 158*, 20–29. https://doi.org/10.1016/j.compag.2019.01.041
23. Vaishnnave, M. P., et al. (2020). Automatic method for classification of groundnut diseases using deep convolutional neural network. *Soft Computing, 24*, 16347–16360. https://doi.org/10.1007/s00500-020-04946-0
24. Tetila, E. C., et al. (2019). Automatic recognition of soybean leaf diseases using UAV images and deep convolutional neural networks. *IEEE Geoscience and Remote Sensing Letters*. https://doi.org/10.1109/LGRS.2019.2932385
25. Thenmozhi, K., & SrinivasuluReddy, U. (2019). Crop pest classification based on deep convolutional neural network and transfer learning. *Computers and Electronics in Agriculture*. https://doi.org/10.1016/j.compag.2019.104906
26. Anami, B. S., et al. (2020). Deep learning approach for recognition and classification of yield affecting paddy crop stresses using field images. *Artificial Intelligence in Agriculture*. https://doi.org/10.1016/j.aiia.2020.03.001
27. Esgario, J. G. M., et al. (2020). Deep learning for classification and severity estimation of coffee leaf biotic stress. *Computers and Electronics in Agriculture*. https://doi.org/10.1016/j.compag.2019.105162
28. Le, T. T., et al. (2016). Deep learning for noninvasive classification of clustered horticultural crops—A case for banana fruit tiers. *Postharvest Biology and Technology*. https://doi.org/10.1016/j.postharvbio.2019.05.023
29. Coulibaly, S., et al. (2020). Deep neural networks with transfer learning in millet crop images. *Computers in Industry, 108*, 115–120. https://doi.org/10.1016/j.compind.2019.02.003
30. Kang, H., & Chen, C. (2019). Fast implementation of real-time fruit detection in apple orchards using deep learning. *Computers and Electronics in Agriculture*. https://doi.org/10.1016/j.compag.2019.105108
31. Cecotti, H., et al. (2020). Grape detection with convolutional neural networks. *Expert Systems with Applications*. https://doi.org/10.1016/j.eswa.2020.113588
32. Qiufeng, Wu., et al. (2019). Identification of soybean leaf diseases via deep learning. *Journal of The Institution of Engineers (India): Series A*. https://doi.org/10.1007/s40030-019-00390-y
33. Fan, S., et al. (2020). On line detection of defective apples using computer vision system combined with deep learning methods. *Journal of Food Engineering*. https://doi.org/10.1016/j.jfoodeng.2020.110102
34. Khamparia, A., et al. (2019). Seasonal crops disease prediction and classification using deep convolutional encoder network. *Circuits, Systems, and Signal Processing*. https://doi.org/10.1007/s00034-019-01041-0
35. Zhu, X., et al. (2018). Method of plant leaf recognition based on improved deep convolutional neural network. *Cognitive Systems Research, 52*, 223–233. https://doi.org/10.1016/j.cogsys.2018.06.008

36. Grinblat, G. L., et al. (2016). Deep learning for plant identification using vein morphological patterns. *Computers and Electronics in Agriculture, 127*, 418–424. https://doi.org/10.1016/j.compag.2016.07.003
37. Ghazi, M. M., et al. (2017). Plant identification using deep neural networks via optimization of transfer learning parameters. *Neurocomputing, 235*, 228–235. https://doi.org/10.1016/j.neucom.2017.01.018
38. Kumar, A., & Sachar, S. (2023). Deep learning techniques in leaf image segmentation and leaf species classification: a survey. *Wireless Personal Communications, 133*, 2379–2410. https://doi.org/10.1007/s11277-024-10873-2
39. Jadhav, S. B., & Patil, S. B. (2024). Plant leaf species identification using LBHPG feature extraction and machine learning classifier technique. *Soft Computing, 28*, 5609–5623. https://doi.org/10.1007/s00500-023-09358-4
40. Manaouch, M., et al. (2023). Predicting potential reforestation areas by *Quercus ilex* (L.) species using machine learning algorithms: Case of upper Ziz, southeastern Morocco. *Environmental Monitoring and Assessment, 195*, 1094. https://doi.org/10.1007/s10661-023-11680-1
41. Meenakshi, T. (2023). Automatic detection of diseases in leaves of medicinal plants using modified logistic regression algorithm. *Wireless Personal Communications, 131*, 2573–2597. https://doi.org/10.1007/s11277-023-10555-5
42. Singh, A., et al. (2023). Smart agriculture framework for automated detection of leaf blast disease in paddy crop using colour slicing and GLCM features based random forest approach. *Wireless Personal Communications, 131*, 2445–2462. https://doi.org/10.1007/s11277-023-10545-7
43. Thanikkal, J. G., et al. (2023). An efficient mobile application for identification of immunity boosting medicinal plants using shape descriptor algorithm. *Wireless Personal Communications, 131*, 1189–1205. https://doi.org/10.1007/s11277-023-10476-3
44. Tholkapiyan, M., et al. (2023). Performance analysis of rice plant diseases identification and classification methodology. *Wireless Personal Communications, 130*, 1317–1341. https://doi.org/10.1007/s11277-023-10333-3
45. Hama, H. M., et al. (2024). Houseplant leaf classification system based on deep learning algorithms. *Journal of Electrical Systems and Inf Technol*. https://doi.org/10.1186/s43067-024-00141-5
46. Kayaalp, K. (2024). A deep ensemble learning method for cherry classification. *European Food Research and Technology, 250*, 1513–1528. https://doi.org/10.1007/s00217-024-04490-3
47. Varma, T., et al. (2024). Automatic mango leaf disease detection using different transfer learning models. *Multimed Tools Appl*. https://doi.org/10.1007/s11042-024-19265-x
48. https://towardsdatascience.com/data-augmentation-techniques-in-python-f216ef5eed69.
49. Cristin, R., et al. (2020). Deep neural network based Rider-Cuckoo search algorithm for plant disease detection. *Artificial Intelligence Review*. https://doi.org/10.1007/s10462-020-09813-w
50. Sharma, U., et al. (2022). A transformer-based model for evaluation of information relevance in online social-media: A case study of covid-19 media posts. *New Gener. Comput, 10*, 1–24. https://doi.org/10.1007/s00354-021-00151-1
51. Remya, S., & Sasikala, R. (2019). Classification of rubberized coir fibres using deep learning-based neural fuzzy decision tree approach. *Soft Computing, 23*, 8471–8485. https://doi.org/10.1007/s00500-019-03961-0
52. Koirala, A., et al. (2019). Deep learning for real-time fruit detection and orchard fruit load estimation: Benchmarking of 'MangoYOLO.' *Precision Agriculture*. https://doi.org/10.1007/s11119-019-09642-0
53. Mao, S., et al. (2020). Automatic cucumber recognition algorithm for harvesting robots in the natural environment using deep learning and multi-feature fusion. *Computers and Electronics in Agriculture*. https://doi.org/10.1016/j.compag.2020.105254
54. Azimi, S., et al. (2021). A deep learning approach to measure stress level in plants due to Nitrogen deficiency. *Measurement*. https://doi.org/10.1016/j.measurement.2020.108650

Siddharth Singh Chouhan currently working as an Assistant Professor in School of Computing Science and Engineering, VIT Bhopal University, Kothri-Kalan, Sehore, 466,114, Madhya Pradesh, India. He has completed Ph.D. in Computer Science and Engineering from Shri Mata Vaishno Devi University Katra, 182,320, UT of Jammu and Kashmir, India. He is a Post Doctorate from the University of Malta, Europe. He has received a B.Tech. degree in 2010 and an MTech. degree in 2013 in Computer Science and Engineering from RGPV University, Bhopal, India. His area of interest includes developing Soft Computing, Computer vision, Drone Technology, Deep Learning, and Precision Agriculture. He had authored several research papers published in reputed journals and conference.

Uday Pratap Singh presently working as an Associate Professor in Department of Mathematics, Central University of Jammu, 181,143, Jammu, India. He has obtained B.Sc. and M.Sc. (Mathematics and Statistics) Degree from Dr. R.M.L. Awadh University, Faizabad, (U.P.), India, M.Sc. (Mathematics and Computing) Degree from Indian Institute of Technology, Guwahati, India, and received Ph.D. in Computer Science from Barkatullah University, Bhopal. His area of interest includes Soft Computing, Nonlinear Systems, and Image Processing. He had authored several research papers in the many reputed journals and conferences. He is a life member of Soft Computing Research Society (SCRC), Barata Ganita Parishad, Computer Society of India and Member of IEEE and AMS.

Utkarsh Sharma is currently pursuing his Ph.D. Degree from Department of Computer Science and Engineering at Jaypee Institute of Engineering and Technology, Madhya Pradesh, 473,226, India. He has about 10 years of experience in teaching. His area of interest includes Machine Learning, Internet of Things.

Sanjeev Jain currently working as a Vice Chancellor of the Central University of Jammu. He received a Master's degree in Computer Science and Engineering from IIT Delhi, New Delhi, India, in 1992, and a Ph.D. degree in Computer Science and Engineering. He has over 35years of experience in teaching, research and administration. He has the credit of making a significant contribution to research and development in the area of image processing, Soft Computing, IoT and mobile ad-hoc network. He had authored several research papers in the many reputed journals and conferences. He has undertaken a number of major Research and Development projects sponsored by the Government and Private Agencies. He is a member of various societies like the Computer Society of India, IEEE.