Journal of Autonomous Intelligence (2024) Volume 7 Issue 1
doi: 10.32629/jai.v7i1.1002

ORIGINAL RESEARCH ARTICLE

Basil plant leaf disease detection using amalgam based deep learning models

Deepak Mane[1,*], Mahendra Deore[2], Rashmi Ashtagi[1], Sandip Shinde[1], Yogesh Gurav[3]

[1] *Department of Computer Engineering, Vishwakarma Institute of Technology, Bibwewadi, Pune 422037, Maharashtra, India*

[2] *Department of Computer Engineering, MKSSS's Cummins College of Engineering for Women, Pune 411052, Maharashtra, India*

[3] *Department of Computer Engineering, Dr. D. Y. Patil Technical Campus, Varale-Talegaon, Pune 410507, Maharashtra, India*

*** Corresponding author:** Deepak Mane, dtmane@gmail.com

ABSTRACT

Medicinal plants have been found and utilized in traditional medical practices from ancient times. Many medicinal plants play a vital role in curating many life threatening diseases. Very few of the medicinal herbs are commercially cultivated. Many plant diseases are there which destroys these medicinal plants. Early detection of plant diseases can prevent the huge loss of these medicinal plants. Here, we presented a hybrid model that makes use of SVM along with the traditional convolutional neural network (CNN) for predicting *Basil* plants leaves diseases. We transformed the conventional CNN model by adding a classification layer Support Vector Machine (SVM) and K-Nearest Neighbor (KNN) after feature extraction and this approach tends to perform better than traditional CNN as we make the dataset balanced by data augmentation and SVN and KNN tend to perform better in case of balanced samples. CNN is used for training, SVM/KNN is used for classification. The advantages of CNN and SVM are used in proposed the CNN and SVM and KNN model. It is assumed that such a combined model would incorporate the benefits of CNN and SVM. Here, we identified the four types of diseases that affect *basil* plant leaves as Leaf spot, Downy mildew, Fusarium wilt, Fungal, and Healthy. Since there isn't a standard dataset for *basil* leaves, we created our own 803 picture data set and used various machine learning techniques to train and evaluate the model. However, over other existing algorithms, our hybrid model i.e., CNN+SVM has produced more accurate results. For five classes of *basil* plant leaves, the proposed model produced 95.02% accuracy of for leaf diseases.

Keywords: plant leaf diseases; image processing; deep learning; CNN; pattern classification; SVM; KNN

ARTICLE INFO

Received: 18 July 2023
Accepted: 28 September 2023
Available online: 27 November 2023

1. Introduction

India is the birthplace of many different medicinal plants and is home to a vast knowledge base of conventional medicine. There are helpful tools available to help treat a variety of ailments. In India, 10% of medicinal herbs are commercially cultivated, the rest are extracted from the forest henceforth there is an immediate need to increase the commercial cultivation of such medical herbs[1]. The major challenge in the commercial plantation is plant diseases which poses a major economical threat and medicinal plants are no exception they are susceptible to various pathogens attacks which may lead to huge crop loss and degradation of medicinal properties. So, early disease detection may prove to be very helpful and beneficial in preventing crop loss and making commercial cultivation of such herbs economically feasible. So to detect such diseases in their early stages we proposed a machine learning-based approach where we used multiple machine learning techniques such as CNN, SVM and hybrid CNN with KNN and CNN with SVM to identify and classify type of plant disease in medicinal plants. The proposed hybrid model algorithm's primary objectives are:

- Collect and preparing a large size dataset for the Indian *basil* plant.
- Design and implement amalgam based deep leaning model.
- Improve the performance of *Basil* leaf disease classification using amalgam based deep learning models.

Overview of the paper is as follow: in section 2, related work is mentioned. In section 3, description of the system architecture is given. Section 4, start by data collection of various disease sample images of medicinal plants. The dataset has a total of 803 images and there are five types of diseased leaves in dataset they are Downy mildew, Fungal, Fusarium Wilt, Healthy, Leaf spot. Then the dataset is splitted into training, testing, and validation. After splitting our data we go for data preprocessing and data augmentation. After data preprocessing, we created our model and trained our model on training data. After the training part, we test our model on the test dataset and evaluate our model using various performance parameters and do a comparative study of different algorithms like CNN, SVM, CNN with KNN and CNN with SVM. In section 5, explained the experimental results and in section 6 we concluded our research.

2. Related work

The automated plant and disease detection method was demonstrated in this study[2] employing superior leaf photos for testing. To distinguish between diseased and healthy regions, the system employs color-based and region-based thresholding approaches, as well as the Random Forest method and Gabor texture feature selection. For binary and multi-label classification, Support Vector Machines and Random Forests are employed. The system has 98.7% accuracy, and a graphic user interface is provided for easy understanding. The system uses edge identification algorithms based on Sobel and Canny models, image resizing and grayscale conversion, and mini-batch techniques to speed up processing time. Database includes 800 images of holy *basil* leaves, including healthy and infected leaves. Experiments demonstrate that the suggested technique is successful in identifying holy *basil* leaves that are curling. In order to identify leaf diseases in plants like *basil*, the article[3] focuses on creating an automated voice-controlled robotic automobile utilising an image sensor network. Users are informed about common ailments and given preventative strategies via the system. Using straightforward K clustering and SVM algorithms for accurate identification without human involvement, it incorporates robot mobility in uneven terrain. The outcome is an autonomous robotic vehicle that is simple to operate and streamlines the diagnosis of plant leaf diseases. High-resolution photos of sick plant leaves were created using DoubleGAN[4], a double generative adversarial network, to balance datasets. The procedure consists of two stages: WGAN, which creates pictures of healthy leaves in the 64 × 64 pixel size, and SRGAN, which expands the imbalanced dataset by creating images in the 256 × 256 pixel size. The enlarged dataset offers sharper graphics in comparison to DCGAN, and it was 99.80% and 99.53% accurate in classifying plant species and illnesses. The results of recognition are better than those of the first dataset. This work[5] proposes a deep CNN model for using leaf images to identify diseases in apple crops. The model's accuracy on the PlantVillage dataset was 98% after 1000 iterations of training on 3171 apple leaves. It outperformed pre-trained CNN models and other existing methods in accuracy and precision, with a range of 97% to 99% for different diseases. The model successfully balanced accuracy and precision, benefiting non-expert farmers and reducing stress on plant pathologists. Utilizing AD Convolution and LAD-Inception, this article[6] proposes a unique real-time early apple leaf disease identification model. Model employs global average pooling to decrease parameters and extracts multiscale characteristics from disease patches of various sizes. On the HUAWEI P40 and Jetson Nano, LAD-Net's size of 1.25 MB results in a recognition performance of 98.58% with a minimal delay of 15.2 ms. This highlights the LAD-Net's capacity to precisely identify early apple leaf pests and illnesses using portable tech support systems. A lightweight model called SE-VRNet[7], based on advanced residual networks and attention mechanisms, and has been proposed to extract more accurate regions of interest and lesions. The model incorporates deep variant residual network and a squeeze-and-excitation module, addressing the difficulty in feature extraction due to leaf disease dispersion. SE-VRNet model achieved top-1 and top-3 accuracy on NewData and SelfData, outperforming other techniques. Outcomes of the experiment demonstrate usefulness and viability of SE-VRNet for detecting leaf diseases on mobile devices. This study[8] presents a MaizeNet deep learning (DL) approach for localizing and categorizing maize plant leaf diseases utilizing an enhanced Faster-RCNN strategy. The approach uses ResNet50 model with spatial-channel attention as key points. This research[9] aims to solve major diseases in Rice and Betel plants using TL-based methods, including ensemble combinations. A robust deep ensemble model called PlantDet that incorporates Global Average Pooling (GAP) layers, Dropout approach, L2 regularizers, and PReLU functions was presented to solve the issues of under-fitting and over-fitting. In this work[10], a deep learning model was trained from beginning to end without any pre-processing or feature-handling. The categorization of skin lesion images into eight different types of skin cancer was proposed using the ISIC 2019 dataset and an upgraded MobileNet with a DL model. The model outperformed dermatologists' accuracy, which was 81 percent, with an average accuracy of 83 percent. Using pre-trained MobileNet architecture and training techniques like focus loss and class-weighted, the model was modified and improved. It exhibited quicker, lighter, more stable and dependable architecture. The model may be used with other publically accessible datasets for skin cancer. In order to support agro-based enterprises, this work[11] proposes a unique approach for identifying tomato leaf diseases utilizing Deep Neural Networks. For a varied dataset, the architecture integrates PCA and PCA DeepNet with a Generative Adversarial Network (GAN). Faster Region-Based CNN is utilized for the detection. The average precision is 98.55%, the total classification accuracy is 99.60%, and the IOU score is 0.95. In order to extract deep characteristics from photos of maize plants, this model[12] leverages the pre-trained CNNs EfficientNetB0 and DenseNet121. Concatenation is used to combine these features into a single, more complicated feature set. The dataset diversity and image count are increased using data augmentation techniques, which enables the model to learn more difficult scenarios. The suggested model outperforms ResNet152 and InceptionV3 by achieving a classification accuracy of 98.56%, proving its superiority over two models. By substituting depth-wise and point-wise convolutions for the original convolutions, the study[13] enhanced the Inception module. High-quality picture features were retrieved using the altered Inception module in conjunction with trained MobileNet. For identifying and detecting different forms of agricultural diseases, a Softmax layer and Single Short Detector (SSD) block were included. For effective model training, two-stage transfer learning was utilized. The U2-Net architecture[14] removes complex background, ensuring high-quality results without compromising original image quality. CNN, EfficientNet, and EfficientNetV2 models were trained, with EfficientNetV2-S and EfficientNetV2-L outperforming others. Using a method[15], infection types are categorised into four categories: healthy, resistant, moderate, and vulnerable. The National Agricultural Research Centre in Islamabad is where the wheat rust dataset is gathered. Stripe rust severity levels are classified using two deep learning classifiers, Xception and ResNet-50, after background noise is removed by a previously trained U2 Net model. ResNet-50 excels, with 96% accuracy. In order to identify *basil* leaf illnesses, this study[16] presents a novel classification model that employs a survival of the fittest strategy and a three-level hierarchical method. Healthy and sick leaves with cercospora spots and downy mildew are used as experimental items in the model. Using a Contrast Limited Adaptive Histogram Equalisation technique, texture and colour information are merged to improve contrast. To create features that are very informative, Random Forest feature selection is used. The model accurately diagnoses illnesses in *basil* leaves with a detection rate of 95.73%.

This paper[17] presents a customized convolutional neural network (CCNN) model for predicting *basil* plant leaves diseases. The model is modified to handle n-dimensional features and unbalanced class data, predicting fungal, downy mildew, fusarium wilt, and healthy diseases. Using a 916 image dataset, the model is trained and tested using various machine learning approaches. It achieves an accuracy of 94.24% for all four *basil* plant leaves categories. Customized based deep learning models produced highest accuracy in the different applications of image processing[18–20].

3. System architecture

When categorizing images that are highly similar to the dataset, CNN performs exceptionally well. CNNs typically struggle to categorize or predict images that have any degree of tilt or rotation, though. SVMs have strong generalization capabilities, making it possible for them to accurately categorize newly discovered data. Hence, we proposed the combined architecture to detect *Basil* plant disease. The CNN model's last output layer was swapped out for an SVM classifier to create the hybrid model architecture. The result values of the hidden layer serve as input characteristics for other classifiers in addition to helping the CNN model make sense. The hybrid architecture combines two machine learning algorithms, namely CNN and SVM or KNN. System architecture of proposed modelis represented in **Figure 1**.

The architecture is divided into two main parts:

(1) Feature extraction using CNN:

To extract valuable features from the input image, it undergoes a preliminary step in which a pre-trained CNN is applied. This involves the utilization of multiple convolutional layers followed by pooling layers. The outcome of this process is a collection of feature vectors that effectively represent the image.

(2) Classification using SVM:

The feature vectors obtained from the CNN are then passed through an SVM/KNN classifier to classify the image into different classes based on the extracted features. The SVM model is trained using labeled data to learn the decision boundary between different classes. KNN algorithm is used for classifying the image based on its nearest neighbors in the feature space. Overall, the hybrid architecture combines the strengths of CNN in feature extraction and SVM/KNNin classification to achieve high accuracy in image classification tasks.

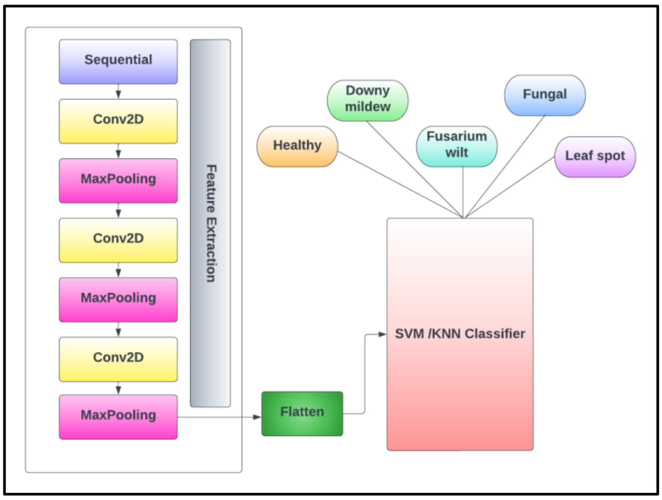

Figure 1. System architecture of hybrid model.

The proposed model effectively handles n-dimensional feature data as well as unbalanced class data. It uses customized convolutional neural networks for classification. The features extracted using CNN are then passed to SVM and KNN for classification. This hybrid approach helps in achieving higher accuracy and works well for unseen test images.

The CNN architecture which consists of convolutional layers and fully connected layers is shown in **Figure 2**. Mathematical model for a CNN involves the combination of various layers such as convolutional layers, pooling layers, and fully connected layers.

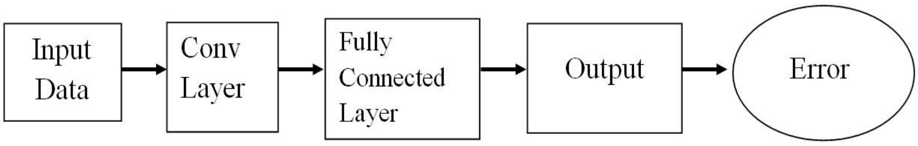

Figure 2. CNN architecture.

The mathematical formula for each layer type varies slightly. Here's an overview of the mathematical formulas for the main components of a CNN:

(1) Convolutional Layer:

The mathematical formula for a convolutional layer, as explained earlier, involves applying a set of filters to input data, performing element-wise multiplication, and adding a bias term. The formula for a single feature map F in a convolutional layer can be expressed as:

$$F[i,j] = activation\left(\sum(X[i{:}i + filter\_height, j{:}j + filter\_width, :\right] \times K[:,:,:]) + b \tag{1}$$

where:

$F[i, j]$ denotes value at position (i, j) in feature map.

$X[i{:}i{+}filter\_height, j{:}j{+}filter\_width, :]$ represents the region of the input data that is multiplied element-wise with the filter.

$K[:, :, :]$ represents the filter weights.

b represents the bias term.

activation( ) represents the activation function applied element-wise to the summed result.

(2) Pooling Layer:

The feature maps produced from the convolutional layers are down sampled using pooling layers. Max pooling is the most popular pooling process, where the highest value inside a window is chosen as the representative value. The mathematical formula for max pooling can be expressed as:

$$P[i,j] = \max\left(F\left(stride_i{:}stride_{(i+1)}, stride_j{:}stride_{(j+1)}\right]\right) \tag{2}$$

where:

$P[i, j]$ denotesvalue at position (i, j) in pooled feature map.

$$F\left(stride_i{:}stride_{(i+1)}, stride_j{:}stride_{(j+1)}\right]$$

represents the window of the feature map being pooled. stride represents the stride or step size at which the pooling window moves.

(3) Fully Connected Layer:

Fully connected layers are utilized to connect features from previous layers to final outcome. The mathematical formula for a fully connected layer involves multiplying the input vector by weight matrices, adding a bias term, and applying an activation function. The formula can be expressed as:

$$Y = activation(W \times X + b) \tag{3}$$

where:

Y represents the output vector of the fully connected layer.

W represents the weight matrix connecting the input X to the output Y.

b represents the bias term.

activation( ) represents the activation function applied element-wise to the summed result.

These mathematical formulas represent the core operations of a CNN. The actual architecture and arrangement of the layers can vary depending on the specific network design and task requirements. The parameters (filter weights, bias terms, and weight matrices) are learned through a process called backpropagation during training, where the network adjusts the values to minimize a given loss function.

4. Dataset preparation and methodology

Data-set preparation:

The *Basil* Plant dataset contains 803 images of five different classes namely Downy mildew, Fungal, Fusarium wilt, Healthy, Leaf spot. There are 186 images of Downy mildew diseased leaf, 129 images of fungaldiseaseleaf, 82 images of Fusarium wilt diseased leaf, 263 healthy images and 78 images with leaf spot. The images are of size 256 × 256.

Data set samples:

The images in **Figure 3** are of healthy leaves. They do not have any disease. They are dark green and do not have any spots.

Figure 3. Healthy images.

Figure 4 shows downy mildew diseased *basil* leaves samples. These leaves are yellow in areas restricted by major leaves.

Figure 4. Downy mildew images.

Figure 5 shows Fusarium wilt diseased images. The growth of these plants is stunned and there are dark streaks on lower side of leaf.

Figure 5. Fusarium wilt images.

Fungal disease images are shown in **Figure 6**. There are dark circles on the leaves which are affected by Fungal disease leaves have spots on them which can be seen in **Figure 7**. **Table 1** describes about dataset samples with its categories.

Figure 6. Fungal images.

Figure 7. Leaf spot images.

Table 1. Description of different classes in dataset.

Disease name	**Symptoms**	**Size**
Downy mildew	Wilting of leaves, brown spots on leaves	186
Fungal	Orange or brown spot on the leaves	129
Fusarium wilt	Yellowing of leaf and brown streak on the stem	82
Healthy leaf	Dark green leaf with no spot	328
Leaf spot	Small brown spots on the leaves and may be surrounded by yellow halo	78

Data augmentation:

For any prediction and classification of image data the quality and number of data entities in our data set is very crucial. The quality of data can be controlled by properly examining the data but what about the quantity of data. It may happen that you might have very small amount data in the data set so in that case we can synthetically increase the quantity of our data by data augmentation. In data augmentation we take a flip an image vertically, rotate the image, scale the image, crop the image, perform translation (move along *x* and *y* axis) on image and by adding gaussian noise. After data augmentation we split our dataset into training and testing set.

4.1. Feature extraction and classification

In CNNs, feature extraction and classification are essential steps for image recognition and classification. Feature extraction is process of detection of important patterns and features from an input image, while classification involves assigning a label or category to the image based on extracted features. In CNNs, feature extraction is performed by applying convolutional filters to input image. These filters capture local patterns and features such as edges, corners, and textures, which are then combined and transformed through a series of non-linear operations to form higher-level features. After feature extraction, classification is typically performed using a fully connected layer and a softmax function, which produces a probability distribution over the possible output classes. Hybrid CNN+SVM and CNN+KNN approaches are techniques that combine the strengths of both CNNs and traditional machine learning algorithms for image classification. In the

CNN+SVM approach, the output features from the CNN are fed into an SVM classifier, which learns to separate the different classes based on these features. The SVM can be trained using a variety of different kernels, depending on the nature of the problem. In the CNN+KNN approach, the output features from the CNN are used to construct a feature space, where the distance between features is used to determine the nearest neighbors of a test image. The class of the test image is then determined based on the most frequent class label among its k-nearest neighbors. Both hybrid approaches have been shown to be effective for image classification tasks, especially when dealing with small datasets or datasets with high class imbalance.

Convolutional Neural Network:

CNN is a deep learning algorithm that is used mainly with image data. CNN is preferred over ANN for image type of data for two main reason first being that in image data we have to consider the value associated with each and every pixel of image when we feed such type of data to fully connected dense neural network then the computation power becomes a major issue as in dense neural network. Each neuron in the next layer is connected to each and every neuron of the previous layer. The second and the most important reason being that there is no feature detection from the image on the global scope in ANN. In CNN for feature detection we have filters and we scan the whole image with this filter to create a feature map that shows the activation of a particular feature in the image after that we apply we use ReLU activation function to bring non linearity to our model after that we perform the pooling operation to further reduce the dimension of the feature map (For our purpose we use Max pooling) after this we flatten our data and feed it to a dense neural network further for classification purpose.

Pooling Layer:

Pooling is the process to merge the data and reduce the size of the data. In this, we make a window of size $k \times k$. Then we have to slide the window from our two-dimensional data. Now further it has two types, one is maximum pooling and average pooling. In maximum pooling we take the highest value and in average pooling we take the average of all the values. This will reduce the size of $n \times n$ sized two-dimensional array into $n/k \times n/k$ sized two-dimensional array.

Flatten Layer: In flattening we convert the data into one dimensional array to feed it to the next layer.

Dense Layer: In neural network, a layer that is connected deeply to its preceding layer is called dense layer. In this layer neurons are connected to every neuron in the preceding layer.

Figure 8 describes the CNN model layers which are arranged sequentially to process input data and extract meaningful features.

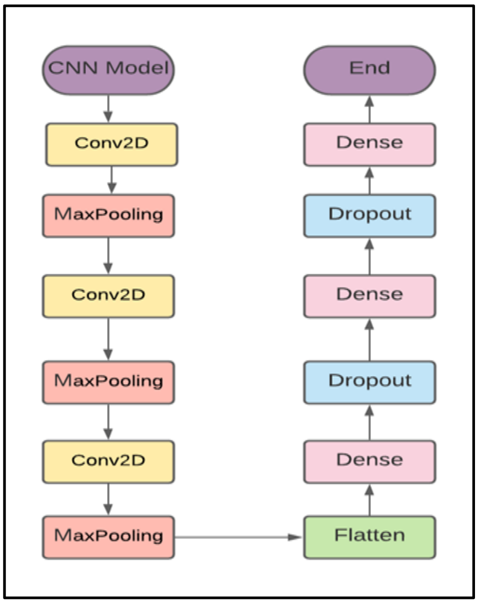

Figure 8. CNN layers.

K nearest neighbour:

KNN is a supervised machine learning algorithm, which is useful to solve classification as well as regression problems. Basically, it works by calculating the distance between the two data points. We calculate the distance of the data point from its *k* neighbors. Then we classify it to the class to which its most number of neighbors belongs to from *k*. We calculate the accuracy by changing the value of *k*, *k* is the number of neighbors.

So, the key idea here is to extract images features using CNN through convolutional layers and after extracting all features we get a feature map which is used to detect feature in image once the features are detected flattening of input is performed and the given is further given to the KNN layer for classification

Support vector machine:

SVM, a widely utilized machine learning algorithm, is employed to solve classification and regression problems. Fundamental principle of SVM is to identify a hyperplane that best divides data into various classes. SVM is a robust algorithm that can work with complex data and deliver accurate results. However, depending on the regularization parameter and kernel function used, it may be computationally expensive for large datasets.

5. Experimental result

Performance evaluated on prepared *basil* leaf datatset. Here, three different models (CNN, CNN with KNN, CNN with SVM) applied on *basil* dataset for predicting the leaf diseases or whether the leaf is healthy or not. Comapre with all models, 95.02% highest accuracy achieved by CNN with SVM based radial basis function (rbf) kernel. Performance graph of training and validation loss represented in **Figure 9**. The validation accuracy initially is very low but as number of epochs increase the validation accuracy also tends to increase and after certain number of epochs the value tends to achieve saturation.

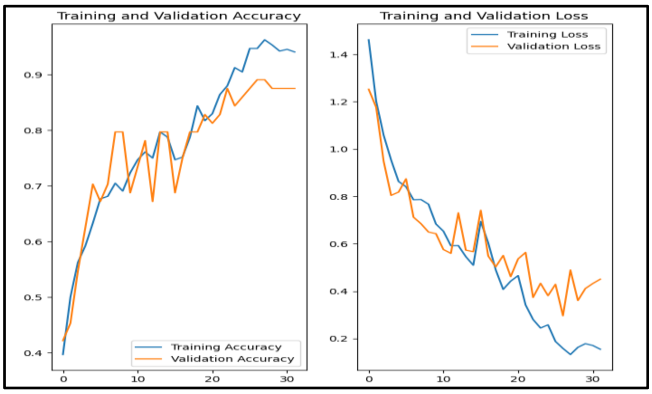

Figure 9. Performance graph between Training vs. Validation.

Predictions results for some test samples can be seen in **Figures 10** and **11** respectively. In the figures, we can see test samples images along with their Actual label and predicted label **Figures 12** and **13** depicts confusion matrices for CNN+KNN and CNN + SVM approaches respectively. These confusion matrices can be used to evaluate the performance of the model as it provides us with information about the number of correctly classified images and misclassified images for each class. The values in the first matrix are scaled.

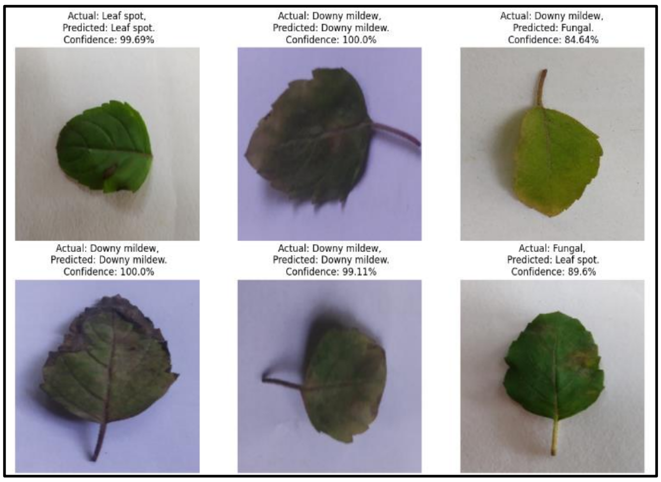

Figure 10. Prediction results of test images.

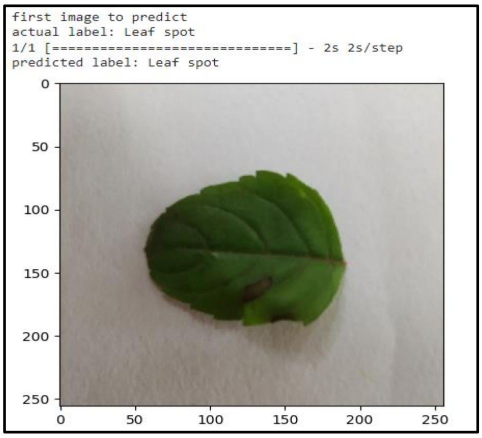

Figure 11. Predictions results.

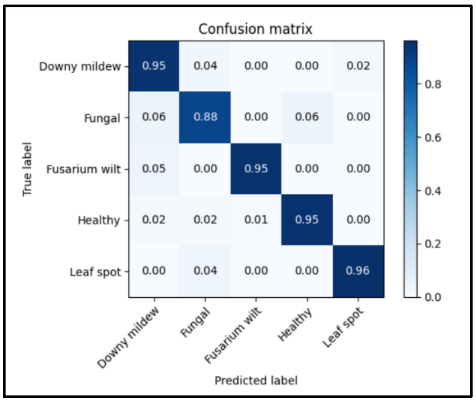

Figure 12. Confusion matrix (CNN+KNN).

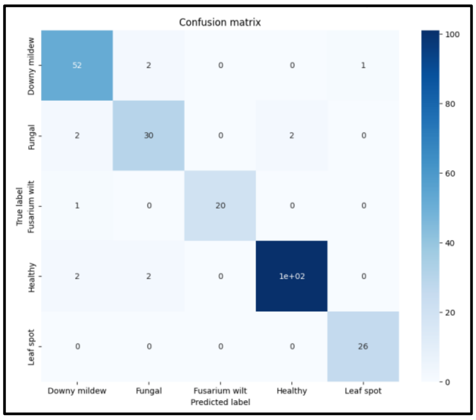

Figure 13. Confusion matrix (CNN + SVM).

Comparing accuracies of three approaches:

Table 2 represents accuracy of CNN model which is 89.84% while hybrid CNN+KNN model accuracy is depicted in **Table 3** (94.19%). **Table 4** represents accuracy of hybrid CNN+SVM model with different kernel. **Table 5** represents comparison between different classification models. CNN is used for training and SVM is used for classification. Hybrid model is proposed to combine the advantages of both the models. The results from **Table 5** clearly shows the proposed model outperforms compared to existing algorithms.

Table 2. Accuracy of CNN.

Algorithm	Accuracy
CNN	89.84

Table 3. Accuracy of CNN+KNN.

Algorithm	Accuracy
CNN+KNN	94.19

Table 4. Accuracy of CNN+SVM.

Algorithm	Kernel	Accuracy
CNN+SVM	poly	94.19
CNN+SVM	sigmoid	94.60
CNN+SVM	rbf	95.02
CNN+SVM	linear	43.56

Table 5. Comparison between different classification models.

Classification models	Accuracy (%)
Random Forest	94.31
Naive Bayes	91.86
KNN	92.43
SVM	92.89
Discriminant analysis	92.30
Bayesian generalized linear	72.04
Extreme gradient boosting	93.87
Customized CNN	94.24
Proposed hybrid model	95.02

6. Conclusion

In this paper, we proposed amalgam based deep learning models to predict *basil* leaf diseases. The Plant Disease Detection system using Machine Learning hybrid models such as CNN, CNN+SVM, and CNN+KNN applied on our own dataset which achieved an accuracy of 95.02%. The dataset consisted of images of plants with and without diseases. The images were preprocessed and augmented to enhance the performance of the models. Overall, the outcomes reveal that the CNN+SVM model outperformed the other models. This may be because the CNN model is better at learning complex features from images, which is crucial for identifying the different types of plant diseases. However, the CNN and CNN+KNN models still achieved high accuracy, indicating that they can be used as alternatives when training a CNN model is not feasible due to time or resource constraints. In conclusion, hybrid deep learning models produced better results to detect plant diseases early and accurately, which can help reduce crop losses and improve agricultural productivity. Further research can focus on exploring other deep learning models or incorporating more advanced techniques such as Transfer Learning to improve the accuracy of the system.

Author contributions

Conceptualization, methodology, writing—original draft preparation, DM; validation, MD; formal analysis, data preparation, SS; writing—review and editing, visualization, RA; result analysis, YG. All authors have read and agreed to the published version of the manuscript.

Conflict of interest

The authors declare no conflict of interest.

References

1. Sofowora A, Ogunbodede E, Onayade A. The role and place of medicinal plants in the strategies for disease prevention. *African Journal of Traditional, Complementary, and Alternative Medicines: AJTCAM* 2013; 10(5): 210–229. doi: 10.4314/ajtcam.v10i5.2
2. Kavitha R, Kavitha M, Srinivasan R, et al. Basil leaf diseases detection using deep learning architectures. In: Proceedings of 2022 IEEE 19th India Council International Conference (INDICON); 24–26 November 2022; Kochi, India. pp. 1–5. doi: 10.1109/INDICON56171.2022.10040158
3. Nooraiyeen A. Robotic vehicle for automated detection of leaf diseases. In: Proceedings of 2020 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT); 2–4 July 2020; Bangalore, India. pp. 1–6. doi: 10.1109/CONECCT50063.2020.9198326
4. Zhao Y, Chen Z, Gao X, et al. Plant disease detection using generated leaves based on DoubleGAN. *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 2022; 19(3): 1817–1826. doi: 10.1109/TCBB.2021.3056683

5. Vishnoi VK, Kumar K, Kumar B, et al. Detection of apple plant diseases using leaf images through convolutional neural network. *IEEE Access* 2022; 11: 6594–6609. doi: 10.1109/ACCESS.2022.3232917
6. Zhu X, Li J, Jai R, et al. LAD-Net: A novel light weight model for early apple leaf pests and diseases classification. *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 2023; 20(2): 1156–1169. doi: 10.1109/TCBB.2022.3191854
7. Xiao Z, Shi Y, Zhu G, et al. Leaf disease detection based on lightweight deep residual network and attention mechanism. *IEEE Access* 2023; 11: 48248–48258. doi: 10.1109/ACCESS.2023.3272985
8. Masood M, Nawaz M, Nazir T, et al. MaizeNet: A deep learning approach for effective recognition of maize plant leaf diseases. *IEEE Access* 2023; 11: 52862–52876. doi: 10.1109/ACCESS.2023.3280260
9. Shovon SH, Mozumder SJ, Pal OK, et al. PlantDet: A robust multi-model ensemble method based on deep learning for plant disease detection. *IEEE Access* 2023; 11: 34846–34859. doi: 10.1109/ACCESS.2023.3264835
10. Mane D, Ashtagi R, Kumbharkar P, et al. An improved transfer learning approach for classification of types of cancer. *Traitement du Signal* 2022; 39(6): 2095–2101. doi: 10.18280/ts.390622
11. Roy K, Chaudhuri SS, Frnda J, et al. Detection of tomato leaf diseases for agro-based industries using novel PCA DeepNet. *IEEE Access* 2023; 11: 14983–15001. doi: 10.1109/ACCESS.2023.3244499
12. Amin H, Darwish A, Hassanien AE, Soliman M. End-to-end deep learning model for corn leaf disease classification. *IEEE Access* 2022; 10: 31103–31115. doi: 10.1109/ACCESS.2022.3159678
13. Chen J, Chen W, Zeb A, et al. Lightweight inception networks for the recognition and detection of rice plant diseases. *IEEE Sensors Journal* 2022; 22(14): 14628–14638. doi: 10.1109/JSEN.2022.3182304
14. Sunil CK, Jaidhar CD, Patil N. Cardamom plant disease detection approach using EfficientNetV2. *IEEE Access* 2022; 10: 789–804. doi: 10.1109/ACCESS.2021.3138920
15. Shafi U, Mumtaz R, Qureshi MDM, et al. Embedded AI for wheat yellow rust infection type classification. *IEEE Access* 2023; 11: 23726–23738. doi: 10.1109/ACCESS.2023.3254430
16. Dhingra G, Kumar V, Joshi HD. Basil leaves disease classification and identification by incorporating survival of fittest approach. *Chemometrics and Intelligent Laboratory Systems* 2019; 186: 1–11. doi: 10.1016/j.chemolab.2019.01.006
17. Mane DT, Sangve S, Jadhav S, et al. Basil leaf disease detection and classification using customized convolutional neural network. In: Suma V, Lorenz P, Baig Z (editors). *Inventive Systems and Control*. Springer, Singapore; 2023. doi: 10.1007/978-981-99-1624-5_50
18. Mane DT, Kulkarni UV. Modified fuzzy hypersphere neural network for pattern classification using supervised clustering. *Procedia Computer Science* 2018; 143: 295–302. doi: 10.1016/ j.procs. 2018.10.399
19. Bhujbal A, Mane DT. A survey on deep learning approaches for vehicle and number plate detection. *International Journal of Scientific & Technology Research* 2019; 8(12): 1378–1383.
20. Mane DT, Kumbharkar PB, Dhotre PS, Borde S. Vehicle-type classification using customized fuzzy convolutional neural network. In: Bhateja V, Satapathy SC, Travieso-González CM, et al. (editors). *Data Engineering and Intelligent Computing. Advances in Intelligent Systems and Computing*. Springer, Singapore; 2021. doi: 10.1007/978-981-16-0171-2_40