

RESEARCH ARTICLE



Flower Disease Diagnosis Model Using Deep Convolutional Neural Network

Bisrat Getnet^{1*}, Desalegn Ashebir², Birhanu Gardie¹

¹ Department of Information Technology, School of Computing & Informatics, Mizan-Tepi University, Ethiopia

² Department of Software Engineering, School of Computing & Informatics, Mizan-Tepi University, Ethiopia



Received: 23-05-2022

Accepted: 14-11-2022

Published: 18-07-2023

Citation: Getnet B, Ashebir D, Gardie B (2023) Flower Disease Diagnosis Model Using Deep Convolutional Neural Network. Indian Journal of Science and Technology 16(27): 2047-2057. <https://doi.org/10.17485/IJST/v16i27.1109>

* **Corresponding author.**

bisratgetinet@mtu.edu.et

Funding: None

Competing Interests: None

Copyright: © 2023 Getnet et al. This is an open access article distributed under the terms of the [Creative Commons Attribution License](https://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Published By Indian Society for Education and Environment ([ISee](https://www.isee.org/))

ISSN

Print: 0974-6846

Electronic: 0974-5645

Abstract

Objective: Design a convolutional neural network model for flower disease detection that can recognize and categorize flower diseases according to their disease types. **Method:** A 10-way SoftMax classification system is used to group the majority of diseases that affect flowers into distinct disease classes (normal, powdery mildew, rose aphid, botrytis blight, downey mildew, red spider mites, Japanese beetles, rose resettle, gray mold, and black spot). A total of 4200 images—70% for training, 15% for validation, and 15% for testing—were used to train and validate the model's performance. From the Tana Flora flower cultivation center in Bahir Dar Amhara, Ethiopia, we obtained a dataset. The proposed model was trained by generating additional images using image augmentation techniques to overcome over fitting problem. **Findings:** Our flower disease identification model produced cutting-edge results with a test accuracy of 94.67%. When used on the same dataset, state-of-the-art models like Alex Net's test accuracy was 85.6%, Google Net's test accuracy was 90.98, and VGG19's test accuracy was 89.3%. Our flower disease detection model has been shown to be faster to train and has a smaller model size thanks to the median filter that we used to improve the quality of the photos and a novel segmentation method that fit our dataset. **Novelty:** We proposed a new CNN architecture and new segmentation algorithm to identify flower disease. Through our experiments, we have shown median filtering, and ways of segmentation improve the performance of our model.

Keywords: Deep learning; Feature Learning; Segmentation; Flower Disease; Median Filtering

1 Introduction

Although flower production is a vital part of horticulture, flower disease can have a big negative influence on the amount and quality of the crop, which has a significant negative impact on Ethiopia's export of flower products. Therefore, disease control is essential for the horticultural industry^(1,2). The key for successful disease control is a correct and fast diagnosis of diseases, so that pesticide control measures can be applied timely. Currently, the most widely used method to diagnose flower diseases is manual

judgment based on the appearance of diseases^(1,2). There are not enough people across the region with skills to perform such tasks in a timely manner. Therefore, a more efficient and convenient method for disease diagnosis of flower is required⁽³⁾. In the farming of flowers, there is number of challenging problems, one of the problems is a disease that attacks them⁽⁴⁾. Physiopathology is the discipline dealing with the diagnosis and management of flower disease. Plant pathologists study plant diseases caused by fungi, bacteria, and viruses⁽⁴⁾. Most of the diseases that attack the flower are powdery mildew, rose Aphid, Botrytis blight, Downey mildew, red spider mites, Japanese beetles, Rose resettle, Gray mold, and Blackspot^(1,4,5). The naked eye visualization of experts is the main old-style approach adopted for the recognition and identification of flower disease, this requires continuous monitoring of experts in a large area of flowers, further in Ethiopia, the farmer may have to go the distance to communicate with an expert this makes checking experts too expensive and time-consuming^(6,7). Nowadays, the development of technology makes it possible to properly identify flower disease. Hence, we can develop a system that identifies flower disease using deep learning techniques particularly convolutional neural networks. flower disease is one of the problems in reducing agricultural products⁽⁴⁾. Many flower diseases produce symptoms that are the main indicators in field diagnosis⁽⁵⁾.

For accurate and effective identification, it is very helpful to design a proper process for floral disease identification. Deep convolutional neural networks (DCNNs) have not yet been investigated for floral disease diagnosis; nevertheless, earlier plant disease identification work has been trained and evaluated utilizing cutting-edge CNN architecture, such as Alex Net and VGG19. This architecture uses a single, massive filter, which lowers performance and increases computational costs. The architecture does not fit with the dataset in our case, CNN overcomes the problem of manual-based feature extraction in classical machine learning algorithms and gives a state of the art result⁽⁸⁻¹⁰⁾ another thing is the characteristic feature of flower images like shape, color, texture, and holes are better to applied CNN rather classical machine learning techniques. The Novelty of this paper is that applied median filter and proposed segmentation algorithm that fit with our dataset finally construct deep convolutional neural networks to achieve a state-of-the-art result to identify and classify nine flowers. The study seeks to make the following contributions .

With the following key advances, we offer a 17-layer deep CNN architecture capable of automatically identifying flower illness. We demonstrate a proposed segmentation approach that enhances the capacity of deep learning systems by enabling the learning of representative characteristic features. The experimental finding demonstrates that using segmentation as opposed to raw picture data as the deep learning model's inputs results in more accurate and computationally efficient identification of flower illness. Technique for choosing the filter size: To get more accuracy when performing convolution processes, filter sizes must be chosen methodically. Based on the length of the bloom in the upper part of the stem and the length of the flower leaves, we chose the filter size. Enhancement of performance: The accuracy, loss, training duration, and model size of the suggested model are all improved. In terms of accuracy, flower disease detection obtained 83% accuracy; however, in our work, we evaluated the model using the test set, and were able to get 94.67% test accuracy.

Image processing techniques, SVM classifiers, K-means clustering, and artificial neural networks are examples of traditional approaches in plant and agricultural disease recognition that take a long time to extract the most representative feature in the agriculture industry^(11,12). In⁽¹³⁾ Proposed Detection and Classification of Plant Diseases, to categorize the objects based on a set of features into n number of classes. By reducing the sum of squares of detachments between the objects and the corresponding class centroid, classification is completed. The Color Co-occurrence method, which is predictable for extracting the feature set, has been tested with well-thought-out five diseases that distress plants: Early scorch, Cottony mold, Ashen mold, late scorch, and tiny paleness. Without taking into account the health plant class, the experiment's accuracy is 84%. Since the approaches are color and stage-dependent and manual feature extraction, both the color and texture of an image are engaged in justification to achieve the single features, which characterize that image. In⁽⁸⁾ proposed plant disease detection using faster region-based convolutional neural networks. However, the dataset used in the research is from different varieties of plant, which highly affected the performance of the model. Disease from different species contain different characteristic feature, which highly affected the result as we consider the number of datasets they used. They have used 36146 datasets to train and validate the model, from which 70% were used for training the model and 30% for testing the final model. In⁽¹⁴⁾ Proposed mango leaf disease identification using CNN to identify anthracnose, sooty mold, red rust, and healthy lea Mango, to train the model they have used 2080 leaf images and split into 80% for training and 20% for testing finally they achieved 81.01% overall test accuracy, this fills to a local minimum, the dataset they used are very small and difficult to achieve the best result, may use traditional machine algorithm to achieve better than this result.⁽⁷⁾ They proposed potato disease identification for two different diseases Early Blight (EB) and Late Blight (LB), and they have used 450 images of healthy and diseased potato leaves, which is taken from publicly available plant village database. From this dataset they used 80% for training and 20% testing. finally, a traditional machine learning classifier is used for identification and classification of diseased and healthy leaves. finally, Random Forest classifier gives a test accuracy of 97%, even if achieved state of the art result but feature extraction is performed in handcrafted based or manual feature extraction since its time consuming and expensive whenever change the dataset. The selection of artificial

features is a complex, time-consuming process that needs to be changed whenever the issue or dataset changes, therefore, constitutes a costly effort that depends on the knowledge of experts and is not well generalized. In⁽⁶⁾ proposed a model that identifies about Ethiopian Enset disease, they used traditional machine learning for classifying the disease, the author used 430 Enset images to classify into bacterial wilt, fusarium wilt, and normal Enset, finally they achieved 92.44% test accuracy. But features are extracted manually which leads to time consume, and expensive whenever the dataset changes.

In⁽⁴⁾ proposed flower disease identification using a machine learning algorithm and applied Gabor feature extraction and they computed seven different measures of dispersion and central tendency to reduce the dimensionality of features, they considered eight classes of diseases, and A total of 320 flower images were used and finally, they achieved 83.3% test accuracy, the performance was little small and fil to a local minimum, feature extraction performed manually and it is expensive. In⁽¹⁵⁾ proposed convolutional neural network for multi-class image classification in plant disease they have used 21,184 images belonging to 10 classes from this 18,520 for training and 2,664 for testing. finally, they achieved 96.02% test accuracy. In⁽¹⁶⁾ proposed identification of rice diseases and pests using CNN, they are adapted and trained using state of the art VGG16 and InceptionV3 CNN architectures, they used a total of 1426 images of rice diseases and pests, finally achieved 97.12% test accuracy.

In⁽¹⁷⁾ proposed Prediction of Liver Disease Using Grouping of Machine Learning Classifiers, they used the data of 583 patients amongst which 416 are the liver patient records and 167 non liver patient records and trained using Logistic Regression, Naive Bayes, KNN, Decision Tree, Random Forest, SVM, ANN model, and authors achieved 0.92 accuracy but the author used traditional machine learning algorithm, so this algorithms need experienced expert to extract learnable features whenever change the dataset, this lead time consuming and expensive In⁽¹⁸⁾ proposed Cancer Detection Using Convolutional Neural Network, The Invasive Ductal Carcinoma Dataset is comprised of two cancer cell types specified as Benign and Malignant, they used Class 0, 44478 images , Class 1 15522 images and finally achieved test accuracy of 79.10%,authors identified and classify only two class and the performance a little small fill to local minima for binary classification its better applied SVM

Before supplying the network with raw data, we introduce a unique segmentation technique and apply median filtering to improve image quality. Another contribution is the strategy used to choose the filter size for the model’s training and validation. Build the CNN model for floral disease detection within a new layer layout to finish.

2 Methodology

Deep learning methods and two-component image processing methods make up the suggested system. We used image preprocessing to resize and rescale the image in the image processing section. We then used a median filter to improve the image quality and proposed image segmentation to reduce computation. The three steps of the proposed system are training, validation, and identification. Each phase comprises SoftMax for categorization and feature learning. The learning model is constructed from training and validation of samples and used for the identification of flower disease depicts in Figure 1.

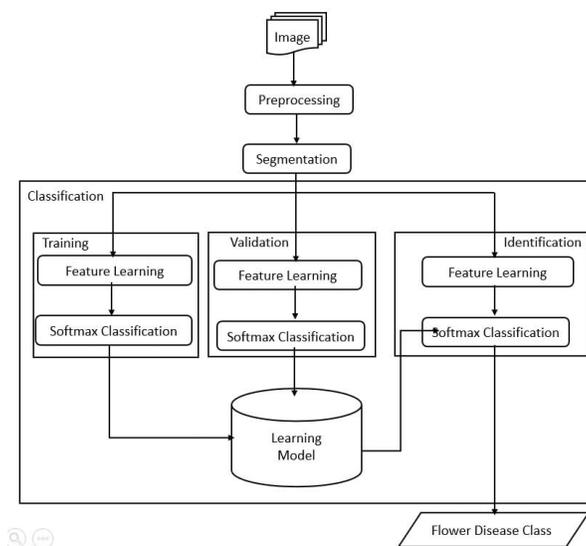


Fig 1. High-level architecture of the proposed flower disease identification model

2.1 Preprocessing

By keeping the camera at a distance of 15 cm and a 45° angle to the ground, the sample photographs were taken in a controlled environment with lighting on a consistent white background. The 16.2 MP Canon digital camera was utilized to capture consistent photos with a resolution of 4608 3570 pixels. To prevent the impacts of sunshine and other environmental factors, photos were taken in a similar controlled atmosphere. Additionally, we have implemented data augmentation techniques and fairly dispersed them to eliminate concerns with data imbalance in every class. The image that was taken may have been impacted by noises and other ambient conditions. As a result, the necessary information might not be obvious. By removing noise and altering the contrast while maintaining the information, preprocessing techniques help to increase the image’s quality. It was determined to employ median filtering and histogram equalization for preprocessing based on the sample photos that were captured. Because median filtering preserves the edge information, which was crucial to our work, it was the best strategy for boosting by removing noises.



Fig 2. Original image

Histogram equalization was applied to enhance the contrast which was necessary to separate the flower disease part from the background as clearly shown in Figure 3.

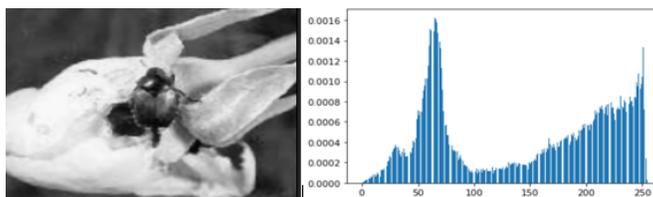


Fig 3. Median filtered image and its histogram

Process before segmentation of the input images as described follows:

Began

1. Encode the JPEG file using the Keras package.
2. Decoding the JPEG file content into different color channels (RGB) grids pixels.
3. Convert these RGB grid pixels into floating-point tensor (3D volume).
4. To reduce computational complexity, images are resized to a standard size (224x224) similar to the size used by most state-of-the-art models^(9,11).

End

2.2 Segmentation

The unique characteristics are not visible across the whole images. This method separates the single distinctive feature of our dataset (area of interest) and does not include the entire image because the experts’ recommended coverage pixel for the image indicates that the disease affected the flower. The method is run based on the image’s length and breadth (dimensions).

Proposed segmentation Algorithm

Began

Step 1: Convert the images into an array.

Step 2: get the dimensions (Height=row and width=column) of the image.

Step 3: Select the initial kernel points;

Step 4: assign x and y integers' values such that their sum is equal=255 like x=89, and y=166. (X, y) contain 8 points for each and connected to the kernel point (xi, Yi) and |T (x, y)-I (xi, Yi)

Step 5: assign I, j to cover the points x and y using this iteration, the loop continues until we cover the segmented region that we want to segment, and the loop continues until the length becomes zero, while the length is reduced by 8 units, each time length and width are reduced by 8 since the size of array x and y is eight and print value of pixel then update the x and y coordinate of the point p, where we want to segment.

Step 6: check whether the point p is inside the region points (an area where we want to segment) If p is inside the region points assign the image region (we want to segment) a value of 1 Otherwise, assign the image region (that cannot be segmented) a value of 0

Step 7: find a region of point 0 and a region of point 1 If the image pixel intensity value is greater than 125 (the threshold value) assign that point value of 1, otherwise 0, since the threshold value selected using empirical optimization (manually searching that are easily to separate region of interest from unwanted background part of the image). As clearly shown in Figure 4 the segmented image was not only separate from the background it also removes an unwanted part of the image.



Fig 4. Median filter image and segmented image from the proposed algorithm

2.3 Feature Learning

In this paper-Training phase has two major tasks feature learning and SoftMax classification. Once the representative features are learned, classification is performed using the SoftMax classifier. Feature Learning for the training phase encompasses ten convolution layers, five pooling layers, and one flattening layer stacked on top of each other for learning the most characteristic or distinguishing representative features. The layout of the layers is depicted in Table 1.

The first convolution layer (Conv1x) down samples the input feature by 2 since the stride size is set at a value of 2. Then, a 1x1 convolution operation (Conv1y) is applied to learn more representative features. Following it, max-pooling (MaxPool1x) is applied for down sampling. These operations (i.e., convolution followed by max-pooling) are repeated (Conv2x, Conv2y, MaxPool2x) to learn and compose more representative features. The convolution module is then applied three times (ConvMod1x, ConvMod1y, and ConvMod1z) to use a convolution operation with different filter sizes at a single layer. In convolution modules, the filter size of each convolution operation is concatenated. The pooling module is then applied (PoolMod1x) for down sampling. We then apply the convolution module two times (ConvMod2x, ConvMod2y) and the pooling module (PoolMod2x) once. The convolution module and pooling module are the building blocks of the proposed system and are used to control the width of the proposed system. We introduce a new level of organization in CNN architecture which is different from the basic CNN architecture where different layers or operations are stacked on top of each other linearly. Finally, a 7x7 max pooling is applied (maxPool12x), which is followed by a flattening operation to flatten the dimension of the feature map, the result of which is used as input for the SoftMax classifier.

Convolution layers are composed of convolution operation followed by an activation function. In addition, filter concatenation is performed in the case of convolution modules. Following is a description of each operation applied to learn the characteristic features that characterize the symptoms of each type of flower disease. Convolution: Convolution (denoted by * operator) over a two-dimensional input image I and two-dimensional kernel K is defined as:

$$S(i, j) = (I * K)(i, j) = \sum_m * \sum_n I(m, n) . (K(i + m, j + n)) \tag{1} \tag{1}$$

Table 1. Description of the proposed classification Deep CNN model

Function	Name	Filter Stride	Size/ Output Size	#1x1 Filters	#3x3 Filters	Parameters
Convolution	Conv1x	3x3/2	112x112x32			6.7K
Convolution	Conv1y	1x1/1	112x112x32		32	11.2K
Max Pooling	MaxPool1x	1x2/2	56x56x32			
Convolution	Conv2x	3x3/1	56x56x64			51.5K
Convolution	Conv2y	1x1/1	56x56x64		64	47K
Maximum Pooling	MaxPool2x	2x2/2	28x28x64			
Convolution Module	ConvMod1x		28x28x112	64	32	32K
Convolution Module	ConvMod1y		28x28x160	96	48	132K
Convolution Module	ConvMod1z		28x28x216	128	64	198K
Pooling Module	PoolMod1x		14x14x432			
Convolution Module	ConvMod2x		14x14x248	160	64	580K
Convolution Module	ConvMod2y		14x14x288	192	64	392K
Pooling Module	PoolMod2x		7x7x576			
Average Pooling	AvgPool1x	7x7/1	1x1x576			
Dropping out	Dropout		1x1x576			
Flattening	Flatten		1x1x10			4000
Classification	SoftMax		1x1x10			
Total						1.447M

Where i and j are an image I coordinates, and m and n are kernel K coordinates. There are 17 convolution operations in the training phase. The input to the first convolution layer is a 224x224x3 image filtered with segmentation. The convolution operation requires four parameters: number of filters, filter size, stride size, and zero Padding. In our model, we have applied 32, 64, 96, 128, 160, and 192 filters. The different number of convolution layers and a different number of filters are tested and those that achieve higher accuracy are selected. A 3x3 and the 1x1 filter size is applied at a single layer. We have used a stride size of two (2x2) and one (1x1). When stride size is two, the image dimension is reduced by half vertically and horizontally. In some cases, stride size is set to one with the same padding, to preserve the size of the input.

2.3.1 Activation

The output of the activation function is always the same as the size (dimension) of the input. Hence, the width, height, and depth of the output layer are the same as the corresponding dimensions of the input layer. ReLU activation function is carried out in the activation layer throughout our model. ReLU activation function returns zero if the values in the input layer are negative, otherwise, it returns the existing value. It is used to introduce non-linearity (to generate nonlinear mappings from inputs to outputs) to make the CNN learn and model more complex functional mappings that exist in representative features. In addition, it is also used to improve the accuracy and performance of the deep CNN model. The effect of ReLU on our data and network.

$$\text{ReLU} = F(X) = \text{Max}(0, X) \tag{2}$$

2.3.2 Filter concatenation

Performed at convolution module and pooling module. In the convolution module, we have used 1x1 and 3x3 filter sizes at the same layer. The filter sizes are selected systematically based on the characteristic features of each category of the flower disease. It has been selected based on the size of the texture shape of the flower, the length of the broken leaf the whole leaf, and the length of the flower leaf. Those characteristics are expected to cover between

4 pixels and 8 pixels. Those pixels' sizes are determined empirically from the dataset. The aforementioned pixel sizes can be detected by using a 1x1 filter size. A 3x3 filter size at a time can learn features covered by 4 pixels whereas a 1 x 1 filter at a time can learn features covered by 8 pixels. A characteristic feature that cannot be learned by a 3x3 filter size can be learned by a filter size of 1x1.

2.3.3 Pooling layer

Pooling layers are maximum pooling. Similarly in the convolution layers, in the case Of pooling modules. Following is a description of each operation pooling: Pooling operation is performed to progressively reduce the width and height of the input volume. The pooling operation requires two parameters. The first parameter is pool size, which controls the window size or block regions on which pooling operation is applied. Since our input texture feature size is huge (224 x 224 pixels), a pooling size of two (2x2) is applied after the Consecutive convolution layers. In addition, a 7x7 pooling operation is applied to flatten the feature map at the final stage. The second parameter is the stride size, which determines the number of pixels skipped while doing the pooling operation. An astride size of 2 (2x2) is used each time we do a pooling operation.

2.3.4 Flattening layer

Only one linear layer is applied to compute the final output probabilities for each class before applying classification. It holds ten nodes (number of classes), which are directly applied to the SoftMax classifier. The flattening layer done by flatten operation is followed by dropout operation to reduce overfitting. Dropout is used to reduce overfitting by randomly disconnecting inputs from the previous layer to the next layer in the network architecture. Random disconnection ensures that no single node is responsible.

SoftMax Classification for Training the output of the final maximum pooling layer is given as input to the SoftMax classifier. A 10-way SoftMax classifier is used for classifying into a specific class (normal powdery mildew, rose Aphid, Botrytis blight, Downey mildew, red spider mites, Japanese beetles, Rose resettle, gray mold, Blackspot). Stochastic gradient descent optimization: It is an optimization algorithm that computes the gradient of the network loss function concerning each weight in the network. Each forward pass through the network results in a certain parameterized loss function. Each of the gradients created for each of the weights is multiplied by a certain learning rate, to move our weights in whatever direction its gradient is pointing^(11,19). Dropping out: It is to be noted that dropout is applied at the final stage during the training phase. In the validation phase, however, dropout layers are applied with a smaller probability, $p = 0.1$ following the activation layer and between two convolution layers e momentum to 0.9. Batch normalization: It normalizes the activations of the given input volume before being applied to the next layer in the network. Even though, it is used to reduce the number of epochs taken to train a neural network and stabilize training⁽⁸⁾. To stable the network during the training and validation phase of the model, we need to normalize the input and adjust the scale since it is used to speed up the training process and allows every layer of the network to learn by itself. But the batch size depends on the amount of dataset if the amount of images is high it needs a large value of batch size and the reverse is true⁽¹⁹⁾.

$$Softmax(z_i) = \left(= \frac{e^{z_i}}{\sum_{j=1}^k e^{z_j}} \right) \tag{3} \tag{3}$$

Here, M is the number of classes. SoftMax is a kind of normalizing function. Finally, 10-way SoftMax used for classification Z =input vector, e^{z_i} =standard exponential function for input vector, k =number of classes in the multi-class classifies, e^{z_j} =standard exponential function for output vector.

3 Results and Discussion

We have used Canon T6 Camera the images were taken in the same controlled environment to avoid external effects of sunlight and other environmental conditions. Accordingly, the distance between the flowers and the camera was approximately 30 cm and we collected the data after sunset. We have a total of 4200 primary datasets before data augmentation within 10 classes and split the complete data into training, validation, and test sets. Then perform the model training on the training set and use the test set for validation purposes, ideally split the data into 70:15:15 ratio. With this approach, there is a possibility of high bias if we have limited data because we would miss some information about the data which we have not used for training. If our data is huge and our test sample and train sample has the same distribution then this approach is acceptable^(11,19). The proposed system is implemented with Keras (Tensor Flow as a backend) using Python programming language, our model is trained in google colab in the cloud to fast training and testing.

3.1 Performance metrics

The performance metrics we used in our study are

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{4} \tag{4}$$

Where TP is True Positive, TN is True Negative, FP is False Positive, and FN is False Negative.

To make the model stable we applied different hyper parameters like an epoch, batch size, and learning rate. We have selected this parameter using empirical optimization (manually searching) after train and validating our model in different hyper parameters we have achieved a state-of-the-art result at 100 epochs and with a batch size of 16, 64,128, and we verified the proposed model with a learning rate of 0.01, 0.001, 0.0001, and 0.00001. In terms of convergence speed and accuracy, the learning rate of 0.0001, batch size of 128, and 50 epochs.

Table 2. Performance of flower disease identification model

Model	Training accuracy	Testing accuracy	Time (Hr.)
Proposed model from raw image	90.56%	87.53	5:25
Proposed model within segmentation	94.5%	92.4%	5:00
Proposed model within median filtering and segmentation	96%	94.67%	4:48
Alex Net	89.6%	85.6%	3:25
VGG19	91.2%	89.3%	5:62
Google Net	92.3%	90.98%	6:12

3.2 Training of Proposed (PDCNN Model by applied median filtering

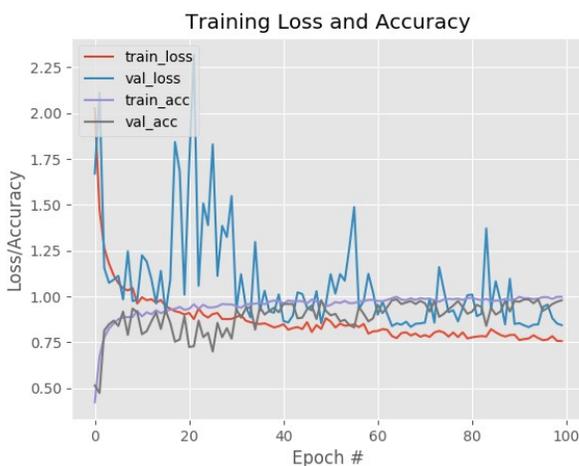


Fig 5. Training and validation curve by inputting raw image into the model

3.3 Training of Proposed (PDCNN) Model by applied segmentation

Segmentation algorithms were used to separate the region of interest from the background and removed unnecessarily part of the flower image to simplify learning features from the input image and computationally free to analyze image information. As clearly shown in Figure 5 training with epoch 50 since it takes 5:00 hours and, it's gotten 94.5% training accuracy, and 92.4 % test accuracy. Since the dataset given to the model after applying the proposed segmentation algorithm minimizes the feature learning process by removing unwanted parts of a flower.

The training progress shows in Figure 6, training and validation accuracy increase and simultaneous decrease in the loss as the number of epochs increases. During the training and validation, the loss is the summation of error for each sample in the training and validation sets. The lower the loss, the better the model and recognition result. The classification accuracy of training and validation is better than in Figure 5. Training and validation loss are also decreased from epoch to epoch. Therefore, we can say that our model generalization capability became much better since validation accuracy is better than training accuracy and validation loss is less than training loss. When we compared to Figure 5 we have got 4.81 % better performance.

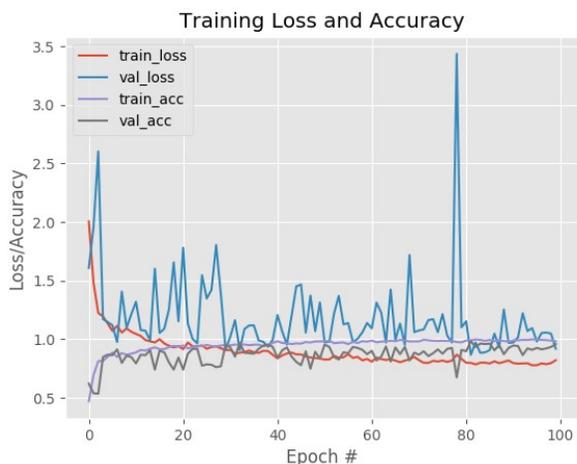


Fig 6. Training and validation curve using a proposed segmentation algorithm

3.4 Training of Proposed (PDCNN) Model by applied median filtering and segmentation

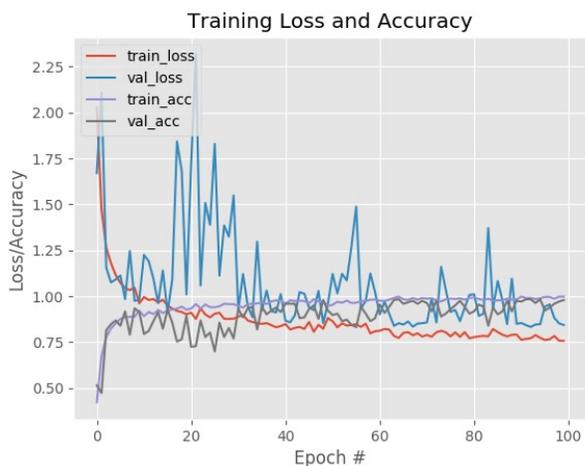


Fig 7. Training and validation curve by applied median filtering and segmentation

Training accuracy outperforms validation accuracy across the curve, as is shown in Figure 7. The difference between training accuracy and validation accuracy is nearly identical to the prior situations, though. In comparison to the previous instance, the validation loss was relatively lower. The model performs quite well since it uses segmentation and median filtering. After epoch 40, training accuracy increases almost linearly and surpasses 96, while validation accuracy fluctuates slightly and eventually reaches 94% accuracy. The validation loss oscillates up and down and approaches 0.8, while the training loss also drops sharply to 0.76. This experiment clearly shows applying median filtering and segmentation highly improves the learning capability of the proposed deep learning system we have achieved 96.45 training accuracy and 94.67 testing accuracy.

3.5 Confusion matrix of proposed (PDCNN) model

As clearly shown in Figure 8 the classification performance of each class from the figure rose rosette and red spider mites class achieved slower performance this has a small amount of dataset used in these two classes, however, deep learning needs a large amount of data the generalized the model easily.

Let’s compare our work with related works; the first research in flower disease was proposed in (4) to identify eight flower diseases and a total of 320 flower images were used finally. They achieved 83.3% test accuracy. The performance was little

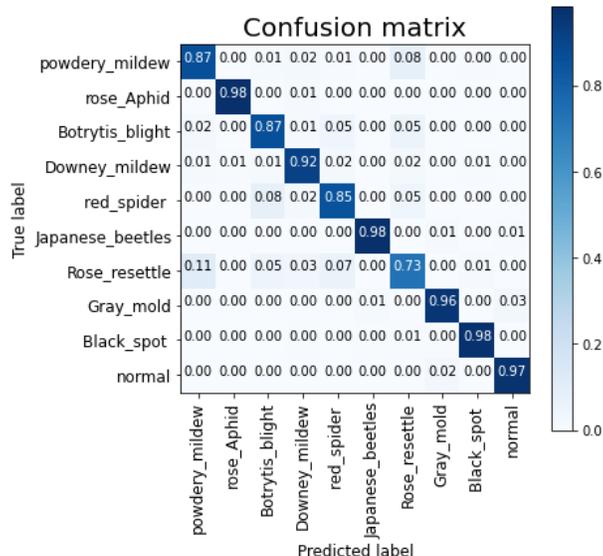


Fig 8. Confusion matrix of proposed flower disease identification model

small and fit to a local minimum, feature extraction performed manually and it's expensive. In⁽⁶⁾ identify two Enset diseases using classical machine learning for classifying the disease, the author used 430 Enset images to classify into bacterial wilt, fusarium wilt, and normal Enset, finally they achieved 92.44% test accuracy. But features are extracted artificially which leads to time consumed and expensive whenever the dataset changes.⁽⁷⁾ Identify two potato diseases Early Blight (EB) and Late Blight (LB), using 450 images from this dataset they used 80% for training and 20% for testing. Random Forest classifier used for identification and classification of diseased and healthy leaves. Achieved 97% test accuracy. Even if however feature extraction is performed in handcrafted based or manual feature extraction since its time consuming and expensive whenever change the dataset.⁽⁸⁾ Plant disease detection using faster region-based convolutional neural networks. They used 36146 datasets to train and validate the model. Finally, they achieved 94.6% test accuracy, however, the dataset used in the research is from a different variety of plants which highly affected the performance of the model. Disease from different species contains different characteristic features which highly affected the performance of the model as we consider the number of datasets they used. In⁽¹⁵⁾ proposed convolutional neural network for multi-class image classification in plant disease they have used 21,184 images belonging to 10 classes. finally, they achieved 96.02% test accuracy. In our work we have considered 10 flower disease identification and built a new structure of CNN architecture.

We have used 4200 image datasets and split them into 70:15:15 ratios for training, validation, and testing respectively, and finally, we have achieved state the art results 94.67% test accuracy with less memory consumption, the reason that we improve the result has we have applied a median filter to enhance the image and proposed segmentation algorithm to separate region of interest from background or remove unwanted part of the image.

4 Conclusion

A limited study has been done on flower disease identification. Therefore, we aimed to propose a convolutional neural network model for flower disease identification. We have evaluated the proposed model through different experiments with 4200 flower diseases and healthy images. The model outperforms several state-of-the-art CNN architectures after being evaluated on the same dataset. A result of 94.67% test accuracy was achieved. This indicates that our proposed flower disease identification CNN-based model can be used to identify new or previously unseen flower diseases more accurately than the other models, Alex Net 85.6% test accuracy, Google net 90.98 test accuracy, and, VGG19 89.3 test accuracy used on the same dataset. We believed that applying median filter enhanced image quality and the proposed segmentation algorithm improved the performance of the model by removing unwanted part of the dataset. For future work, we will aim to improve the model training process by automating the search and selection of the key influencing parameters (i.e., number of the filter size, number of filters, and number of fully connected (dense) layers) that jointly result in the optimal performing CNN model. Furthermore, increasing the size of data, and trained with pre-trained models may improve the identification performance of CNN model.

References

- 1) Sustainability to Ethiopia's Growing Floriculture Industry _ SourceTrace Systems_1. . Available from: <https://www.sourcetrace.com/blog/bringing-in-sustainability-to-ethiopia-booming-floriculture-industry/>.
- 2) Ethiopia Flowers _ Exporter & Supplier - Good Prices_best. . Available from: <https://www.selinawamucii.com/produce/flowers/ethiopia-flowers/>.
- 3) The Ethiopian flower industry has shown remarkable growth over the past years. . Available from: <https://www.floraldaily.com/article/9319845/the-ethiopian-flower-industry-has-shown-remarkable-growth-over-the-past-years/>.
- 4) Tigistu G, Assabie Y. Automatic identification of flower diseases using artificial neural networks. *AFRICON 2015*. 2015. Available from: <https://doi.org/10.1109/AFRCON.2015.7332020>.
- 5) Jain V, Yadav A. Analysis of Performance of Machine Learning. *2019 IEEE International Conference on E-health Networking*. 2021;p. 706–709. Available from: <https://doi.org/10.1109/HealthCom46333.2019.9009442>.
- 6) Ganore KA, Tigistu G. Ethiopian Enset Diseases Diagnosis Model Using Image Processing and Machine Learning Techniques. *International Journal of Intelligent Information Systems*. 2020;9(1):1–5. Available from: <https://doi.org/10.11648/j.ijis.20200901.11>.
- 7) Iqbal MA, Talukder KH. Detection of Potato Disease Using Image Segmentation and Machine Learning. *2020 International Conference on Wireless Communications Signal Processing and Networking (WiSPNET)*. 2020;p. 43–47. Available from: <https://doi.org/10.1109/WiSPNET48689.2020.9198563>.
- 8) Shrestha G, Deepsikha, Das M, Dey N. Plant Disease Detection Using CNN. *2020 IEEE Applied Signal Processing Conference (ASPCON)*. 2020;12:109–113. Available from: <https://doi.org/10.1109/ASPCON49795.2020.9276722>.
- 9) Yuan Y, Chen L, Wu H, Li L. Advanced agricultural disease image recognition technologies: A review. *Information Processing in Agriculture*. 2022;9(1):48–59. Available from: <https://doi.org/10.1016/j.inpa.2021.01.003>.
- 10) Thaiyalnayaki K, Joseph C. Classification of plant disease using SVM and deep learning. *Materials Today: Proceedings*. 2021;47:468–470. Available from: <https://doi.org/10.1016/j.matpr.2021.05.029>.
- 11) Wagle SA, Harikrishnan R, Ali SHM, Faseehuddin M. Classification of Plant Leaves Using New Compact Convolutional Neural Network Models. *Plants*. 2022;11(1):24. Available from: <https://doi.org/10.3390/plants11010024>.
- 12) Rao DS. Plant disease classification using deep bilinear cnn. *Intelligent Automation and Soft Computing*. 2022;31(1):161–176. Available from: <http://dx.doi.org/10.5373/JARDCS/V12SP1/20201058>.
- 13) Panchal P, Raman VC, Mantri S. Plant Diseases Detection and Classification using Machine Learning Models. *2019 4th International Conference on Computational Systems and Information Technology for Sustainable Solution (CSITSS)*. 2019;4:1–6. Available from: <https://doi.org/10.1109/CSITSS47250.2019.9031029>.
- 14) Orbien C. Identification of Carabao Mango Leaf Disease using Convolutional Neural Network. *Journal of Advanced Research in Dynamical and Control Systems*. 2020;12:152–158. Available from: <http://dx.doi.org/10.5373/JARDCS/V12SP1/20201058>.
- 15) Negi A, Kumar K, Chauhan P. Deep Neural Network-Based Multi-Class Image Classification for Plant Diseases. *Agricultural Informatics*. 2021;p. 117–129. Available from: <https://doi.org/10.1002/9781119769231.ch6>.
- 16) Rahman CR, Arko PS, Ali ME, Khan MAI, Apon SH, Nowrin F, et al. Identification and recognition of rice diseases and pests using convolutional neural networks. *Biosystems Engineering*. 2020;194:112–120. Available from: <https://doi.org/10.48550/arXiv.1812.01043>.
- 17) Shreya K, Singh M, Kuma K. Prediction of Liver Disease Using Grouping of Machine Learning Classifiers. *Materials Today: Proceedings*. 2021. Available from: https://DOI:10.1007/978-3-030-67187-7_35.
- 18) Dabral I, Singh M, Kumar K. Cancer Detection Using Convolutional Neural Network. *Conference Proceedings of ICDLAIR2019*. 2021;p. 290–298. Available from: http://dx.doi.org/10.1007/978-3-030-67187-7_30.
- 19) Kumar K, Kumar M, Samsonu C, Krishna KV. Role of convolutional neural networks for any real time image classification, recognition and analysis. *Materials Today: Proceedings*. 2021. Available from: <https://doi.org/10.1016/j.matpr.2021.02.186>.