The Visual Computer
https://doi.org/10.1007/s00371-023-03127-y

ORIGINAL ARTICLE

A nightshade crop leaf disease detection using enhance-nightshade-CNN for ground truth data

Barkha M. Joshi[1] · **Hetal Bhavsar**[2]

Accepted: 27 September 2023
© The Author(s), under exclusive licence to Springer-Verlag GmbH Germany, part of Springer Nature 2023

Abstract
In the fast-growing agricultural world, early detection of plant diseases is crucial for maintaining crop health and ensuring successful harvests. Advancements in computer vision technology have led to the development of advanced methods for diagnosing plant diseases. However, factors like lighting, weather, and the number of diseases in a single image can make it difficult to detect plant diseases. Traditional deep learning-based algorithms have drawbacks, such as high hardware investment, inference speed, and generalization. This research article aims to raise awareness among farmers about cutting-edge technology for detecting plant leaf disease in nightshade crops. The Enhance-Nightshade-CNN model was used to enhance the quality of nightshade crop leaf disease samples, achieving good accuracy compared to existing algorithms. The model accurately identified healthy and unhealthy leaves in the real environment, with ground truth results showing a 95–100% accuracy rate.

Keywords Background removal methods · Enhance-nightshade-CNN (proposed model) · Disease identification · Model complexity

1 Introduction

A turning point and a catalyst for early human social development is agriculture. Agriculture has been developed thanks to the significant developments in science and technology made possible by the wealth of human society. Humans can now generate enough food to feed seven billion people thanks to advances in technology. However, many developing nations are experiencing a food crisis and are losing money and becoming hungry. It is undeniable that developing nations lack advanced agricultural research and technology, even while political considerations in those countries, like economic instability and social unrest, do affect food production & distribution. Furthermore, other external factors, like climate change, plant pests and diseases, and others frequently pose a danger to food security. Additionally, impoverished countries rely heavily on their labour force to perform leaf categorization in terms of quality or skip this step because of its high cost at the time of plant leaf quality testing. This choice causes the agricultural sector in these areas to stagnate and further impedes the raising of the standard of living for the local populace. Therefore, it is essential to create a low-cost, positively automated system for detecting leaf disease.

The primary objective of the current work is to detect the healthiness of nightshade crop leaf disease from a complex background. Identify the specific disease due to happen by fungus, bacteria, and viruses [1]. Suggested remedial measures so that the same illness would not happen in the nearer future. The proposed model use the less no of layers so that it generates the less no of parameters. Due to generation of less parameters model required the less time for the execution and required the less no of resources. The model is not much expensive as compare with other existing models. Hence small farmer can also use this model for the disease detection on night shade crop leaf. The model is generalized so that it identified the fungal bacterial and virus disease very accurately for the nightshade crop leaf. Hence farmer can identify the Nightshade crop leaf disease accurately of the movement of drone.

✉ Barkha M. Joshi
barkhajoshi.comp@svitvasad.ac.in

Hetal Bhavsar
hetal.bhavsar-cse@msubaroda.ac.in

1 Computer Engineering Department, Sardar Vallabhbhai Patel Institute of Technology, Vasad, India

2 Computer Science and Engineering Department, The Maharaja Sayajirao University of Baroda, Vadodara, India

Published online: 27 October 2023

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

The paper is design in the section wise. Section 1 is covered with the introduction. Literature survey is covered in Sect. 2. The methodology, implementation and results cover in Sect. 3 and 4 respectively. The model complexity cover in Sect. 5 and conclusion in Sect. 6. Figure 1 displays the flow of the paper with the graphical representation.

Plant diseases' object detection has a variety of agricultural application possibilities, offering prompt feedback on plant health, directing crop cultivation as well as post-treatment, and therefore sharply reducing expenses. Hardware support for plant object detection has been made available by the growth of microcomputers and mobile computing devices. Convolutional neural networks in particular, which are based on backpropagation algorithms, offer software support. The development of autonomous DL-based methods for plant disease detection uses has so been started by some researchers.

2 Literature survey

Many nations rely heavily on their agricultural sectors. Food demand is rising steadily as a result of population growth. It is essential to boost agricultural productivity and safeguard crops to meet this urgent need. Horticulture might use technology that facilitates the advancement of plant knowledge and development to close this gap. This technique uses image processing and ML to diagnose leaf illnesses in the current techniques However, because of the numerous pathogens that are prevalent in their environment, crops are extremely vulnerable to various diseases. These disease pathogens include bacteria, fungi, and viruses, among others [2]. Crop diseases could cause productivity losses of between 10 and 95% [3], which significantly reduces both the amount and agricultural quality output. Thus, early illness detection is essential to avoid serious losses and to cut down on pesticide overuse, which may be harmful to human health as well as the environment. Farmers generally detect crop ailments by eye based on visual indications, particularly in developing countries and on small farms. This laborious task necessitates extensive treatment time and plant pathology knowledge [4]. Additionally, if a rare disease attacks a field, farmers seek expert help to make a precise and timely diagnosis, which inevitably drives up treatment costs [5]. Hence, this type of visual observation is neither viable nor practical for big farms, and it might even produce incorrect estimations as a result of biased decisions [6]. To meet growing customer expectations and lessen the negative environmental impacts of chemical inputs on the human health and environment, researchers have been inspired to develop technical approaches for the precise, quick, as well as reliable early identification of crop illnesses. In this regard, a number of techniques [7–10] have been put forth to automate the disease diagnosis process. Direct and indirect methods are the two categories into which these techniques for the automatic recognition of agricultural diseases are split [11–13].

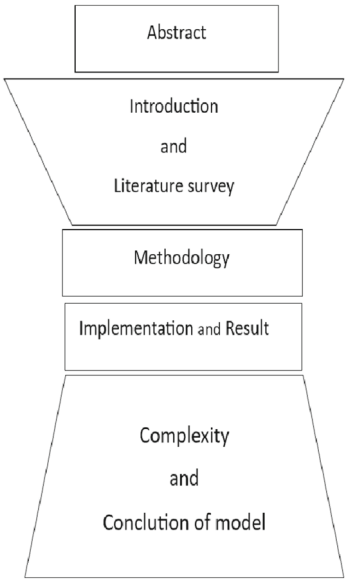

Fig. 1 Paper flow diagram

The [14] authors Thanjai Vadivel, and R. Suguna have been going through diseases in crops. The author's main objective is to unite farmers with cutting-edge technologies to reduce plant leaf diseases. To implement the concept, 'Tomato' is chosen in which leaf diseases are anticipated and recognized by AI (Artificial Intelligence) algorithms, convolution neural network (CNN), and computer technology. Farmers could compare the symptoms with the expected illnesses based on the leaf forms of tomato plants [15, 16]. Its comparison of different classifications and approaches/filters with several methods, like SVM (Support Vector Machine), BPNN (Back-Propagation Neural Network), Optimised MLP (Multilayer perceptron), RBF (Radial Basis Function), NN classifier, CNN Classifier, Kernel, and K-Means classifier. The current approach's classification accuracy after the experiment is CNN–94.4%, BPNN–85.5%, NN–97, Optimised MLP–91.4%, SVM–88.8%, k-means–85.3%, and RBF − 89%.

The authors M. Ferni Ukrit and J. Sujithra have implemented the methods [17] that use image processing procedures like classification, feature reduction, feature extraction, segmentation, and pre-processing to classify leaf illness. It uses many processing steps and region-based classification but only recognizes the disease kind. To segregate the leaf diseases obtained from the augmented leaf dataset, a combination technique of regional-based CNNs and UNet(CRUN) is suggested in this research [18–20]. The proposed algorithm determines the severity of the disease in the leaf, a morphological technique is used for the segmented pictures. The

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

Table 1 Background removal methods

	Methods	Advantages	Limitation
[26]	Principle Components analysis	- Remove the correlated features - Improve the performance - Enhances visualization - Decreases overfitting	- Independent variables become less interpretable - Data standardization is should before PCA - Data Loss
[27]	Independent Component analysis	- Recovering 'sources' or unseen signals from observed mixes, - Utilizing only the notion of the signals' mutual independence	- Signals
[48]	Linear descriptor analysis	- Higher dimensional space to lower dimensional space	- Multilinear regression analysis
[49]	Locally linear Embedding	- Being able to handle a lot of high-dimensional data	- Unsupervised learning algorithm
[50]	Histogram	- The usage of histograms for continuous, discontinuous, as well as even unordered data is simple and useful	- Make analysis difficult when too many blocks - Too few can leave out important data - Difficult for categorical data
[51]	Otsu	- Speed very fast. operates on histogram - Ease coding	- Complex computation - Required more time for processing
[52]	Optical flow	- It often performs well as a close approximation of the real physical motion projected onto the image plane	Motion detection
[53]	Grab cut	Minimum cut algorithm to separate the image in background and foreground	- The segmentation is initialized by user interaction which may lead to bad segmentation if initialization quality is poor
[54]	Watershed	- Capable of producing an exact segmentation of the image into distinct sections - Needs low computation time - Offers closed contours - Quick, easy, and intuitive process	- It is highly sensitive to local minima - If an image with noise, this will influence the segmentation
[55]	Contour feature	- It may be used to map an area's cross-section while showing the supplied characteristics - It offers the framework for the colouring technique	- Some of the heights are not displayed owing to the vertical interval restrictions - Not appropriate for a small-scale map since it can hide some details
[56]	Graph cut	- Low error rate, quick and precise, and produces a reasonable segmentation result with just two input strokes on pictures with big unknown regions	- It could not handle transparent or semi-transparent boundaries - The "shrinking bias" in graph cut makes it difficult to segment thin, lengthy objects

identification of the leaf's kind aids in preventing the disease from spreading to other leaves by using fertilizers correctly. The suggested approach is used to analyse real-time photographs of bacterial blight and red rot on sugarcane leaves and yellow and black Sigatoka on banana leaves. The illness zone is effectively segmented by the suggested CRUN algorithm. The morphological process aids in determining the severity of the illness and safeguards the plant against its spread. So, for automated leaf disease identification and prevention, the suggested CRUN as well as morphological tests are the most efficient.

Vijay Singh and A.K. Misra present an algorithm [6] for the image segmentation method which is utilized for the automatic detection along with the plant leaf illnesses classification. Image segmentation, which is a significant feature of disease identification in plant leaf disease, is performed with a genetic algorithm. The algorithm has been implemented by [21], and the salient images of PET are segmented by the "optimized GrabCut" algorithm. Experimental findings demonstrate that the suggested method has certain benefits over the other two algorithms in terms of easy operation and algorithm efficiency. It also efficiently increases image segmentation's efficiency while ensuring reliable findings [20, 22–25].

The principle components analysis have removed the correlated features and improved the performance [26]. The principle components analysis method is used to remove correlated features, improve performance and visualization, and also reduce the overfitting problem. The methods have some limitations too which is mentioned in Table 1. Independent

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

component analysis is discussed in [27]. This method is used for the signals for the recovery process. There are different approaches of image processing were used for the background removal process. Some methods were used for the signals, some used for the independent variables, and some for the categorical dataset. Table 1 displays the comparison of various image processing approaches with the limitation.

Though there had been a lot of study done in agriculture for disease detection. Early identification of plant leaf disease is essential today in order to maximise plan productivity. Deep learning technology is now utilised to detect plant leaf diseases as a result of technological advancements.

3 Methodology

The classification of plant species using leaf image analysis has received a lot of research attention. The ability to obtain measurements and shape-related characteristics from the leaf itself is essential to the categorization process. The presence of many leaf's stems, branches, and background objects in plant photographs, however, frequently hinders the investigation process and must be removed before a species can be identified [28–30]. The majority of the time, researchers manually modify the image to remove these obstructing elements. The cleaning of the leaf picture should be automated to make species categorization via plant photography more practicable.

This would enable the development of programs that identify the species of a plant from a user's provided photo of a plant. There are various phases involved in fully automating the selection and extraction of a target leaf from a picture. The plant must first be removed from the image's background elements, such as the ground and sky. The next step is to identify and divide each leaf in the image into a separate sub-image. The non-leaf items, such as stems and branches, must be eliminated from each of the leaf sub-images. Then, any partially obscured leaves should be disregarded as potential targets. If several non-occluded leaves are discovered, the best candidate for classification must be chosen [1, 31–35]. Although a fully automated solution is not inconceivable for the foreseeable future, this research concentrates on a few of the problems raised while leaving others for further investigation. Because backdrop objects frequently strongly resemble plants in terms of colour, segmenting the plant from other objects can be difficult. Because leaves from the same plant are frequently the same hue, ignoring partially obscured leaves is another difficult issue. Even the human eye sometimes has trouble telling where one leaf ends and another begins.

For an application that allows end users to input a plant image, the automated leaf extraction methods currently in use are not adequate. The automated leaf extraction method must be able to handle leaves of diverse colours, shapes, sizes, and placements within the image for such an application to be practical. This thesis will offer an algorithm to automatically extract potential target leaves from a plant image.

3.1 Morphological operations

Tools that can be used to extract image components include morphological operators [6]. These instruments are used on binary images to connect, isolate, or extend foreground (or white pixel) regions. Digital image processing morphological processes are based on set theory ideas. The pixels in a binary image are regarded as being in a set A for the sake of the morphological operations utilized in this study. Additionally, a structuring element, a smaller binary picture, is produced, and the pixels contained in this smaller image are regarded as belonging to set B. Structuring elements, which can come in a range of sizes and shapes, are typically formed at the start of a morphological procedure. Unless otherwise stated, it can be assumed that a circular organizing element was used for this thesis. The 4 convolutions of set A and set B with some specific set operation thus executes each morphological action.

3.1.1 Dilation

The foreground edges of a binary image are expanded or grown as a result of the morphological process of dilation. The union of sets A and B is referred to as dilatation in set theory. The figure shows a binary picture that has experienced some dilatation. The foreground area before the dilatation is shown in dark green. The circles represent the structural component of the procedure, while the light green depicts the foreground region that was added by the dilation [36]. If any of the pixels under the structuring element are foregrounded in the original picture, the pixel where the structuring element is centred is set to foreground in the dilated image. If there are no foreground pixels in the original picture below the structuring element, the pixel where the structuring element is centered is set to the background in the dilated image. As shown in Fig. 2, dilatation causes the foreground area to expand around the margins, including the dark and light green portions.

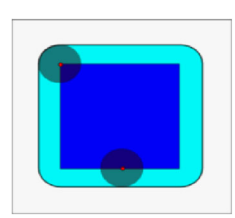

Fig. 2 Dilation

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

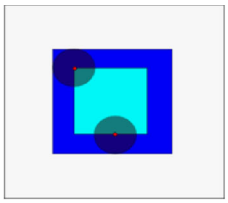

Fig. 3 Erosion

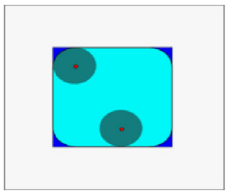

Fig. 4 Opening

3.1.2 Erosion

The morphological process of erosion is applied to eliminate the foreground region's edges from a binary image. The intersection of sets A and B can be used to define erosion in terms of set theory. An illustration of erosion on a binary picture can be seen in Fig. 3. The area in the dark green shows the foreground before erosion. The circles represent the structural element employed in the erosion, and the light green portion shows the area in the foreground following the erosion [37, 38]. The pixel where the structuring element is centered remains in the foreground in the degraded image if all of the pixels below it are in the foreground within the original image. In the eroded image, the pixel where the structuring element is centered is set to the background and any pixel beneath the structuring element within the original image is the background. As may be seen, erosion has resulted in the trimming of the foreground area's edges.

3.1.3 Opening

Erosion followed by dilation employing the same structural element is the morphological process of opening.

Foreground portions smaller than the structuring element are eliminated as a result of the opening process which is demonstrated in Fig. 4. By removing corners and any thin parts that connect the larger foreground regions, foreground regions larger than the structuring element are detached. The light green area in the figure shows the threshold value" chosen is too low foreground that results after the opening procedure, whereas the dark green areas show the foreground region before the opening process. As can be seen, the erosion process removes the tiny dark green rectangle, preventing it from expanding throughout the dilation step [39].

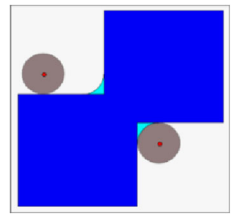

Fig. 5 Closing

The two bigger dark green squares, on the other hand, would be reduced in size by erosion and then stretched by dilatation to retain their entire original shape but for the corners.

3.1.4 Closing

The dilation and degradation of the image utilizing the same structural element constitute the morphological process of closing. Small holes in foreground regions or small gaps between foreground regions become the foreground as a result of the closing process, as seen in Fig. 5.

Figure 4's dark green sections show the foreground regions before the closure process. The mix of light green and dark green patches is the outcome of the closure process [40, 41]. A larger front area would result from the dilation. With the expectation of areas that were contiguous or near the image boundary in the original binary image, the erosion that results from the dilation trims the foreground region back to its roughly original shape.

The dilation, erosion, opening and closing these four steps use throughout the research for the different type of changes in the leaf. Based on the changes in leaf we can identify the diseases accurately. Throughout the research from starting to ending morphology operation is used for identify the leaf disease from the complex background. We have considered the leaf background, leaf detection, vein detection and protrusion detection from the complex environment. The leaf marker is located in the leaf region using greyscale morphological processing, but edge information is also utilised. For the foreground markers used the closing by reconstruction and opening by reconstruction method for the clean up the image.

3.2 Otsu thresholding

A threshold value is selected to turn a grayscale image into a binary image, where all values below the threshold turn into a zero, or black, in the binary image, and all values above the threshold value turn into a one or white. As a result, the outcome of the binary image depends heavily on the threshold value chosen. If the threshold value chosen is too high, elements that should be included as foreground objects in

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

the binary picture will be deleted as background areas. If the threshold value chosen is too low, background regions of the image would be included as foreground components within binary image [42–44]. For the aim of this study, the ideal threshold value for transforming grayscale photographs into binary images was determined using Otsu's thresholding technique. To get the value that optimizes the between-class variance, Otsu's technique examines the histogram of the grayscale image. The spread for pixel values above and below a certain threshold value is measured by the between-class variance. The best threshold value for foreground and background separation in the image is chosen by maximizing the between-class variance.

4 Implementation and results

4.1 Enhance-nightshade-CNN (proposed model)

Despite being capable of reliably identifying the disease, the Alex Net, VGG, and Google Net architectures need a lot of resources and computing time to do it. To identify diseases using deep learning, a model that can do it quickly and accurately while using a minimal number of resources was needed. The activation function, max-pooling layer, convolution layer, and fully linked layer are used in the recommended model for the categorization of the particular illness. Five "convolutional layers" and one fully linked layer make up the proposed model Night-CNN [32].

The Night-CNN [32] was used for the plain background leaf of nightshade crops. The model has accurately identified the diseases. Though always not possible that a leaf is available on a plain background. There may be possible to have a leaf in complex background. Hence leaf disease detection from complex background images needs to be considered. The proposed model, Enhance-Night-CNN considered the complex leaf which is available in the real environment. The leaf is extracted from the complex background. For the extraction of the leaf watershed and graph cut method is used. The proposed algorithm considered the black and white background, soil background, and highly detailed background.

Watershed is utilized to identify the leaf from the background and the graph cut approach is used to identify the boundary of the leaf. For the vein and protrusion detection graph cut method is used. Figure 6 displays the flow chart of the suggested algorithm Enhance-Nightshade-CNN. RGB colour image 256 * 256 size image applies to the watershed and graph cut method. This method extra the leaf from the complex background. The output image will be a partially removed background image. The partial pain background images are applied to the Night-CNN [32]. Finally get the output with the specific disease name based on the fungal, bacterial, and viral.

4.2 Dataset

The crops like Lycopersicon(tomato), Tuberosum(potato), Capsicum annuum (bell pepper), and Melongena (Eggplant) leaves healthy and diseases images have been downloaded from Google. Images of leaves with disorders are obtained to conduct tests for the diagnosis of leaf disease. The leaf image samples in the database are revealed in Fig. 7. All the images have been captured in the sunlight before 5 p.m. images have a total of 18 classes.

4.3 Pre-processing

Leaf disease detection is a very much challenging task for the farmer in a complex background environment. The leaf disease can be detected after giving the proper size to the convolution neural network. The complex leaf can be captured in the sunlight from 10 am to 5 pm duration. The leaf can be resized to 227 × 227. The image is considered in RGB format. The augmentation rotation range is 25, the width & height shift range is 0.1, the shear range is 0.2 and the zoom range is 0.2.

4.4 Algorithm

The disease identification of the nightshade crop leaf has been implemented in the Python language. The application has been implemented in the Keras and TensorFlow environments. The model is trained in the google cloud environment. We have used the Googlecolab facility for disease identification. For background detection, we have considered steps that are covered in Fig. 8.

The Enhance-Nightshade-CNN has two main steps. In the first step, it does partially remove the background of the image, and in the second phase applies the partially remove background to the Nightshade-CNN [32] for disease identification or classification.

4.4.1 Partial background removal method

There are many methods like principle component analysis, grab cut, graph cut watershed, histogram, etc. for removing the background [40, 45]. Out of the mentioned method some of the methods used for the audio signals video, etc., very few methods we can use for background removal. In this research work, we have used the grab cut and graph cut with the watershed. Though the grab cut removes the background precisely we must assign the co-ordinated to extract the leaf so sometimes it does not give a good result. Hence, we have used a method that combines the graph cut and watershed. We have considered the background detection, leaf detection, vein detection and protrusion detection. The background detection we have considered excess

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

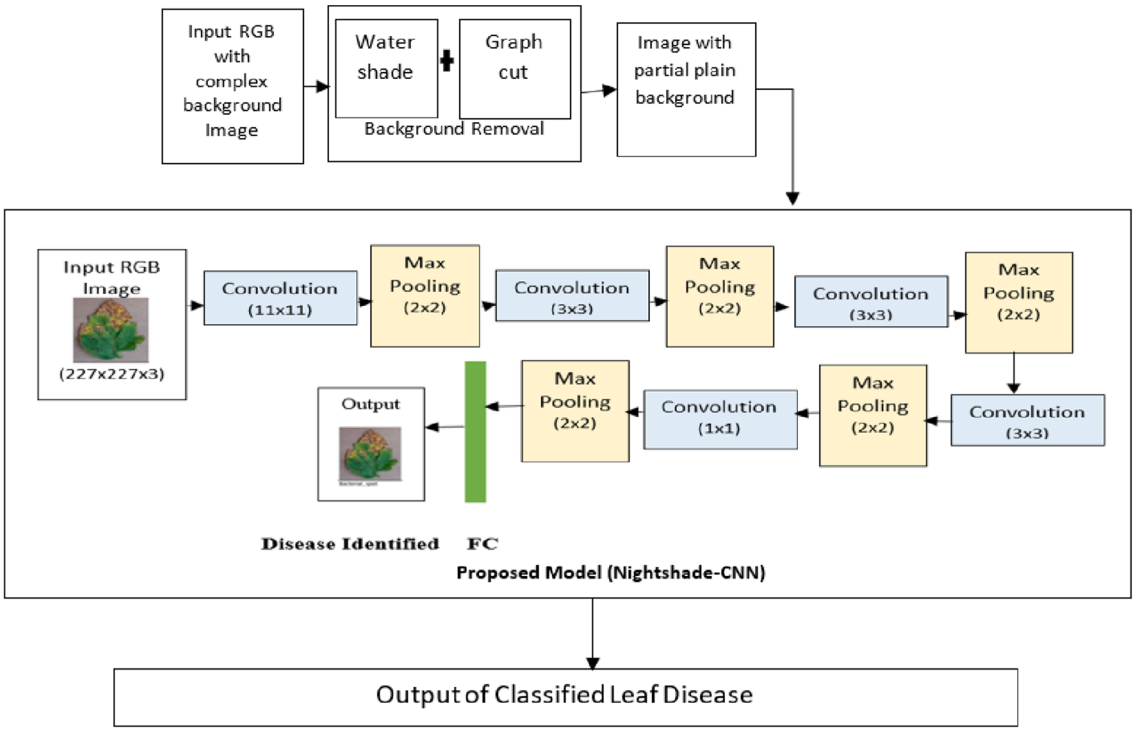

Fig. 6 Enhance-nightshade-CNN (proposed model)

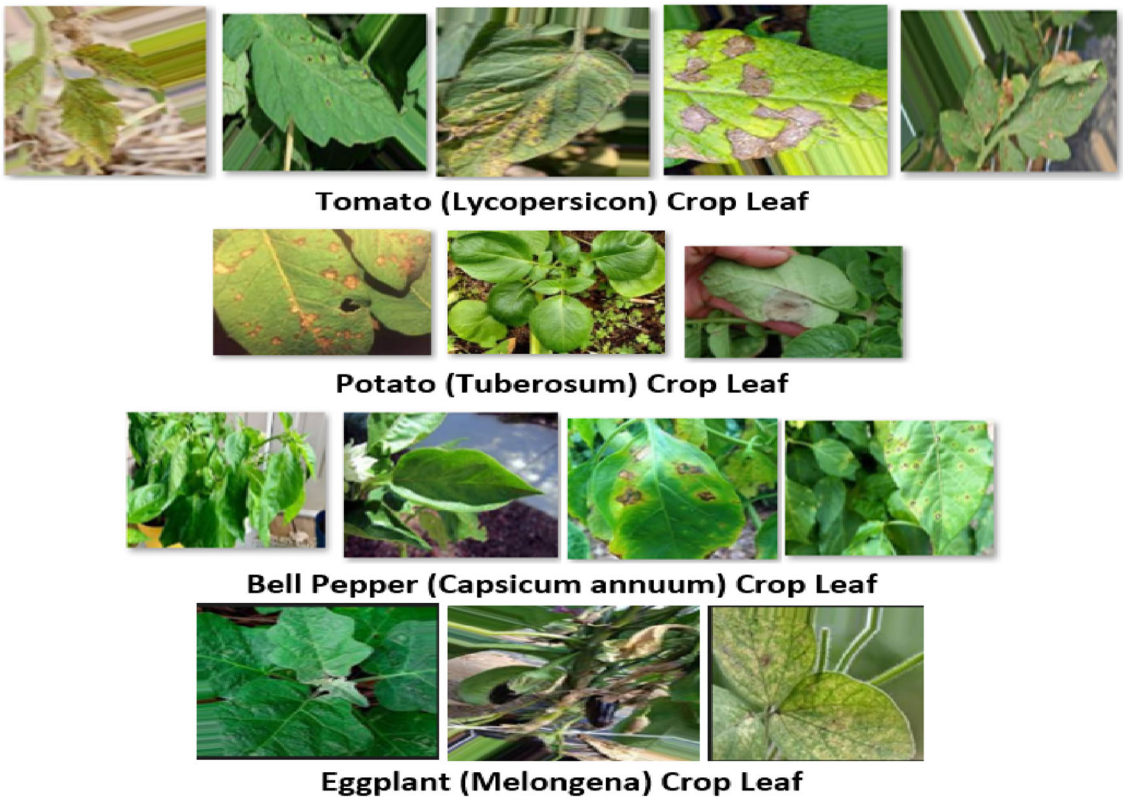

Fig. 7 Nightshade crop with complex background

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

Fig. 8 Enhance-nightshade-CNN algorithm

1. Input the RGB leaf image with complex background
2. Identification of background and leaf using watershed
   1. Background detection
      If (excess green index = = 2G-R-B) && (excess red index ==1.4R-G-B)) => soil background
      If color = = white (R>200, G>220, B>200) => background
      If color = = black (R, G, B <30) => background
      If (color = = highly detail) => texture background
      If (hole = = 100 pixel) => neglect
   2. Leaf detection
      - Greyscale morphology
      - Solidity measurement
      - Edge making
   3. Vein detection
   4. Protrusion detection
3. Cut out leaf region by Graph cut and watershed.
4. Leaf with partially plain background apply to the proposed CNN model for identify the accuracy
5. Test the dataset
6. Disease is classified based on the features.

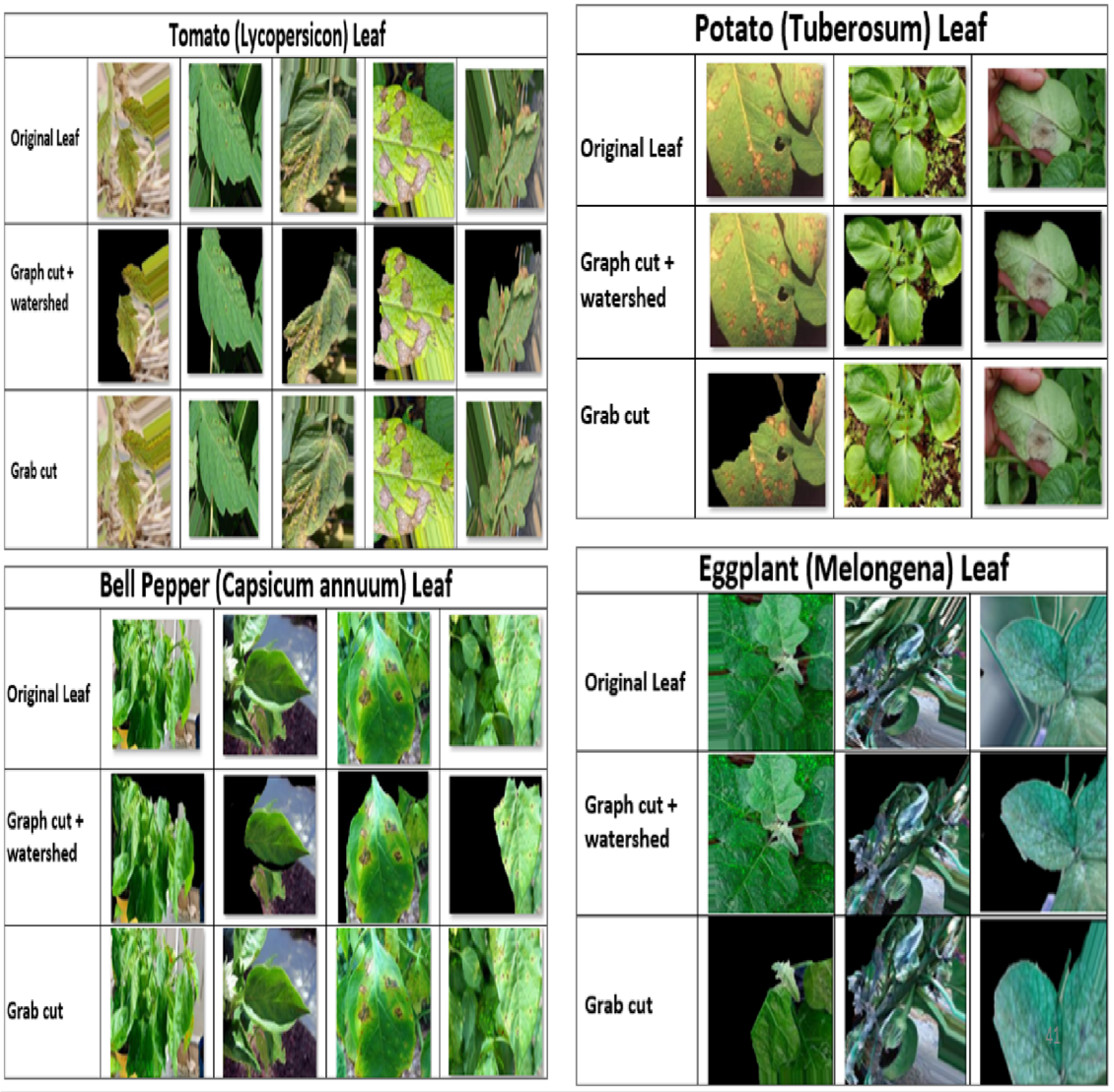

Fig. 9 Partial background removal methods

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

green index and excess red index [46]. White and black colour as background is also considered. Highly detail colour like grass we have considered as background and if small hole appears on the leaf then neglect that small portion. The leaf detection inputs are the inverse areas of the background markers. The marker-based watershed segmentation method is used here. The foreground markers are discovered by employing "opening-by-reconstruction" and "closing-by-reconstruction" techniques. Tidy up the image and connect blobs of pixels within each of the foreground items. The local maxima of greyscale picture are computed. The results are then forced to be the gradient magnitude map's local minima for the watershed transform. The transformed image's split image may have numerous sections. The largest green one is chosen first. Other regions that connect to the main region will be highlighted. Indication if their intensity and colour are similar, and if the combined region has a greater proportion of pixels in the convex hull than the largest region. If, following the region-combining operation, this proportion is still less than a set threshold, the edge information is utilized to clip off areas that are not in the target leaf. The discovered edges link to the result's boundary. The solidity measurement is used to differentiate small regions that may exist outside the actual leaf. The largest area is chosen and then eroded to obtain the front-most leaf marker. The strongest line is detected in the initial leaf so it is considered as primary vein. Other lines with different angle is to be considered as a secondary vein. Then after apply the watershed with graph cut method for leaf identification. The retrieved leaf's contour is transformed to polar coordinates. When detecting unlikely graph shapes, an approximation of the genuine outline is computed and utilised to replace the discovered outline. As a result, the protrusions and notches are gone. Figure 9 displays the image processing method comparison for the detection of leaf from the complex background. For the background removal we have used the watershed with graph cut and grab cut algorithm. The result of the background removal process proved that the combination of graph cut and watershed gives a very fruitful result. So, through the work for the background removal process we have considered the graph cut with a watershed method for partial removal of background from the real environment images, and now that partial plain background image is considered as input for the proposed Night-CNN [32].

For the leaf identification from the complex background, we have considered the texture of the leaf. The leaves have been identified from the complex environment and then will apply that leaf for the further step for the identification.

4.4.2 Disease identification or classification

The Enhance-Nightshade-CNN is used to classify healthy and unhealthy diseases as per the symptoms. The disease can be classified using convolution and hidden layers. In this research work, we have used the SoftMax function at the output layer. We have decided the learning rate of 0.0001 which controls the Night-CNN [32] and Enhance-Nightshade-CNN. Figure 10 demonstrates the different learning rate for the accurate result consideration. Adam optimizer is applied for the leaf disease classification. NightCNN [32] and Enhance-Nightshade-CNN have been implemented in the five convolutions and one hidden layer is used.

4.5 Hyperparameters

The proposed model NightCNN [32] and Enhance-Nightshade-CNN used the less no. of layers so that generated hyperparameters are less as compared with the current existing models like AlexNet,VGG and GoogleNet.. We have considered the hyperparameters for the layers are 500 Epochs, filter size is 11×11,3×3 and 1×1. The model used the 5-convolution layer and 1 fully connected layer for the disease identification. We have considered the 0.0001 learning rate for the model. The no. of layers are less used for the proposed model hence it required comparatively less time than the other existing models. Table 2 displays the comparison of the suggested model with the existing model. The graphical representation is covered in Fig. 11.

4.6 Result

The model accuracy has been evaluated by different parameters. In this research we have considered the parameters like F-measure, Recall, and Precession for the proposed Enhance-Nightshade-CNN model. Positive incidences as a percentage of total expected positive instances. The denominator in this case is the model prediction made as positive from the entire dataset. Consider it as determining "how much the model is correct when it says it is correct." Positive incidences as a percentage of overall positive instances. As a result, the denominator (TP + FN) here represents the actual number of positive cases in the collection. Consider it as determining "how many extra right ones the model missed when it showed the right ones." It is the summing of precision and recall. This factor considers both, so the greater the F1 score, the better. As you can see, if the product in the numerator is low, the final F1 score is substantially lower. So, a model performs well in F1 if the positives predicted are indeed positives (precision) and does not overlook positives and forecast them negatively (recall). The plain background for the nightshade crop accuracy and comparison is already discussed in [32]. The same concept applies to identifying the disease from a complex background. Table 3 displays the confusion matrix for the accuracy for the nightshade crop like Lycopersicon, turberosum, capsicum annuum and melongena leaf crop. Figures 12, 13 and 14 display the confusion matrix of tomato,

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

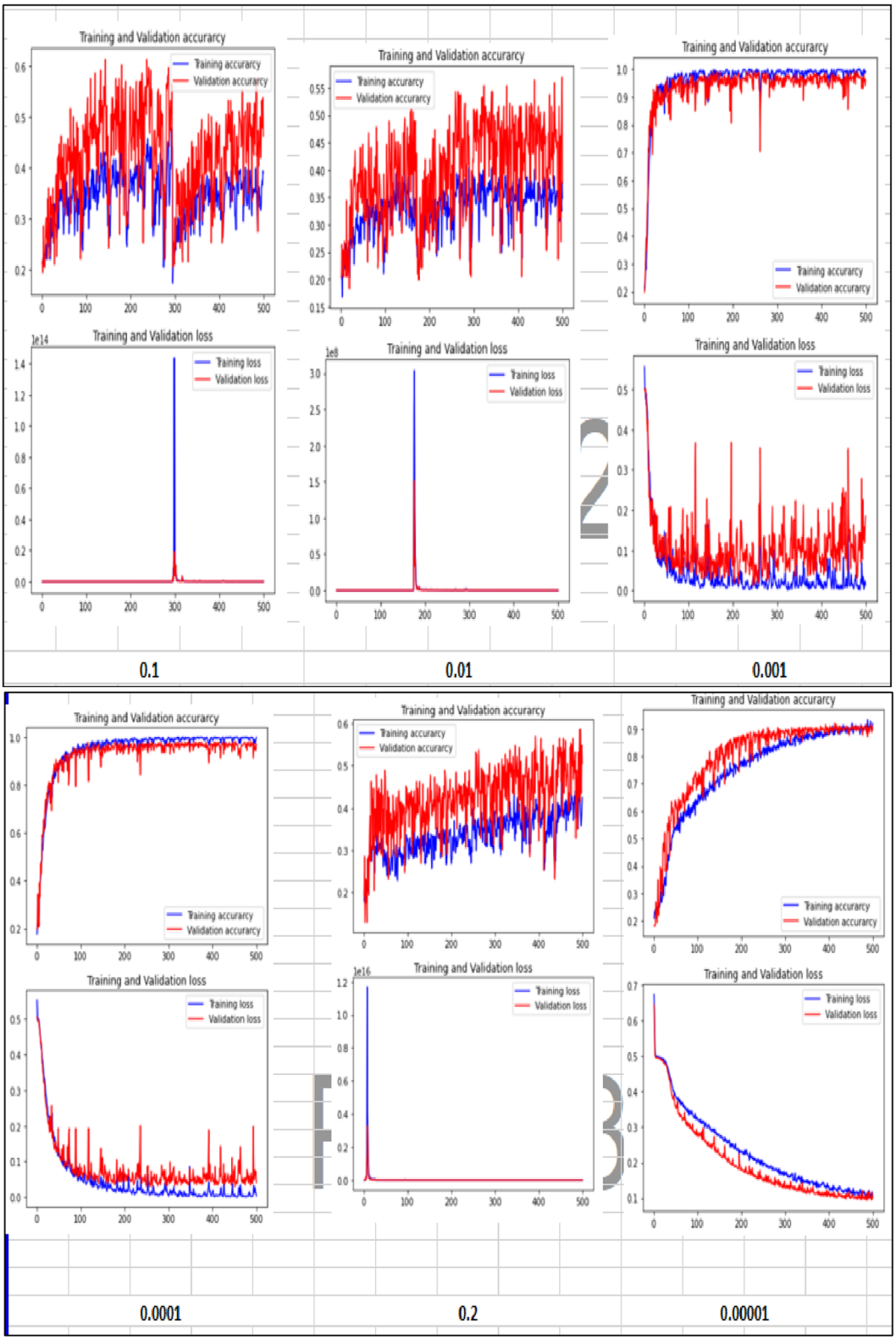

Fig. 10 Learning rate 0.0001 consideration graph

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

Table 2 Nights-CNN [32] and enhance-nightshade-CNN hyperparameter's comparison with existing model

Model name	Hyper parameters	Used filters	No of layers
AlexNet	62,388,354	11 × 11, 5 × 5 and 3 × 3	8
VGG	133,416,524	3 × 3 and 2 × 2	16
GoogleNet	360,233,624	1 × 1, 3 × 3 and 5 × 5	22
Nightshade-CNN	2,639,178	11 × 11, 3 × 3 and 1 × 1	6
Enhance-nightshade-CNN	2,639,178	11 × 11, 3 × 3 and 1 × 1	6

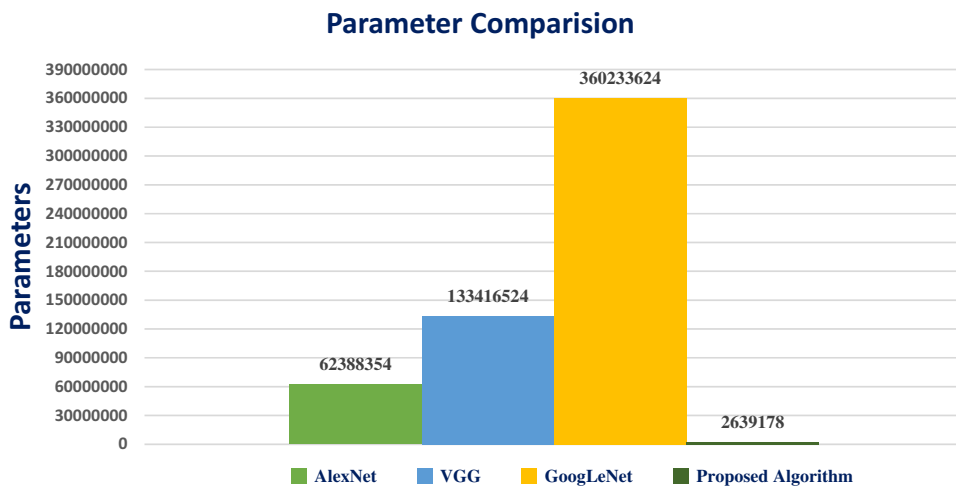

Fig. 11 Parameter comparison

bell pepper, potato and eggplant leaf disease. Figure 11 displays the Lycopersicon(tomato) leaf diseases like bacterial spot, early blight, late blight, leaf mould, Septoria leaf spot, spider mite, target spot, yellow curl, mosaic virus and healthy leaf. The result almost gives the 95.23% accuracy based on the precision, recall and fi score.

Figure 12 displays the Bell pepper leaf diseases of bacterial spot, early blight and the healthy leaf and the healthy leaf and late blight of potato leaf diseases. Around 97% accuracy we have achieved of the bell pepper and potato leaf diseases. We have achieved the 100% accuracy of the eggplant leaf diseases which is displayed in Fig. 13.

Though the algorithm required more time as first we remove the background from the leaf and then will apply the disease detection technique by using the proposed CNN [32]. It gives an accurate result with a less error rate as compared to the other existing models. Proposed Enhance-Nightshade-CNN identified the Lycopersicon crop leaf and turberosum and capsicum annuum and melongena leaf testing accuracy in Figs. 15, 16, and 17. Figures 15, 16, and 17 show that, when epochs are increased by 20 to 500 then the accuracy increases for the proposed Enhance-Nightshade-CNN model. After the 500 epochs the accuracy is constant so that, we have set the threshold for the 500 epochs. The error rate is demonstrated in Figs. 18, 19, and 20 for the Lycopersicon, turberosum , capsicum annuum and melongena leaf disease, respectively. The figures show that, in the initial state the proposed Enhance-Nightshade CNN model generate very much high error but when the epochs increase the error rate decreases.

The error rate for the Lycopersicon crop leaf is 6.20% which is demonstrated in the graph of Fig. 17. Figure 18 displays the error rate 5.34% of the enhance-Nightshade CNN for the bell pepper and potato leaf disease and for the Melongena leaf disease error rate of the proposed Enhance-Nightshade CNN is 6%.

The Lycopersicon crop, tuberosum crop, and "capsicum annuum" crop healthy as well as unhealthy leaves are tested on the Python-based application using the Enhance-Nightshade-CNN model. Figure 21 displays the different disease leaf and healthy leaf of the Lycopersicon crop. Figures 22 and 23 display the healthy and unhealthy leaves of the tuberosum and capsicum annuum crop leaf, respectively, and Melongena leaf diseases are covered in Fig. 24.

Enhance-Nightshade-CNN required less time to train the model as compared to the other standard model like AlexNet, VGG, and Google Net. The proposed model has been trained in the Google Colab using the Tesla T4 GPU environment. Figure 25 demonstrates the comparison chart for the time required to train the Enhance-Nightshade CNN model.

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

Table 3 Confusion matrix

Tomato crop leaf disease	Precision	Recall	F1-score
Tomato bacterial spot	0.96	1.00	1.00
Tomato early blight	0.95	0.94	1.00
Tomato late blight	1.00	0.93	1.00
Tomato leaf mould	0.94	0.94	0.95
Tomato septorial leaf spot	1.00	1.00	1.00
Tomato spider mites two spotted spider mite	0.94	0.95	1.00
Tomato target spot	1.00	1.00	0.94
Tomato yellow leaf curl	0.96	1.00	1.00
Tomato mosaic virus	1.00	0.95	0.95
Tomato healthy	0.95	1.00	1.00
Accuracy			95.23%
Bell pepper and potato crop leaf disease			
Bell pepper bacterial spot	0.99	1.00	0.99
Bell pepper healthy	1.00	0.99	0.93
Potato early blight	0.99	0.98	0.97
Potato healthy	1.00	0.99	1.00
Potato late blight	1.00	0.99	0.98
Accuracy			97%
Eggplant crop leaf			
Two spotted spider mite	1.00	1.00	1.00
Healthy	1.00	1.00	1.00
Phytophthora bright	1.00	1.00	1.00
Accuracy			100%

Fig. 12 Tomato Leaf disease confusion matrix

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

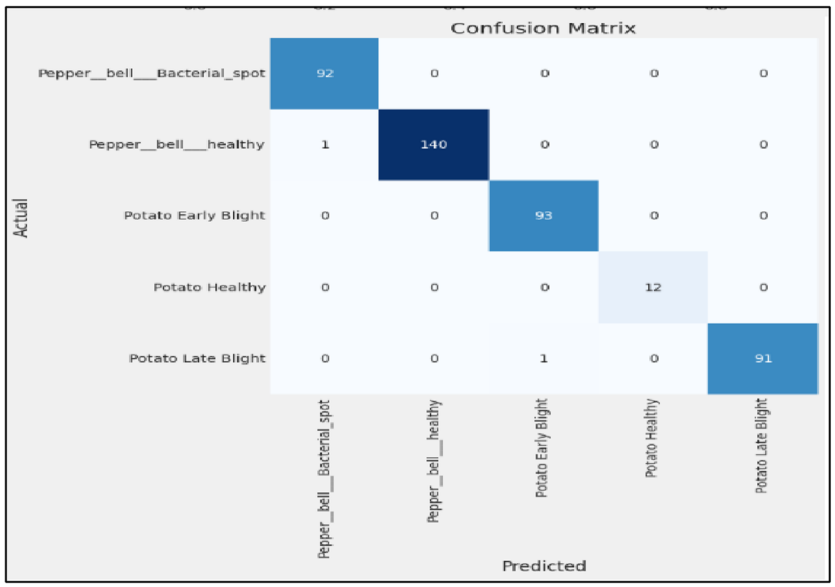

Fig. 13 Bell Pepper and Potato Leaf disease confusion matrix

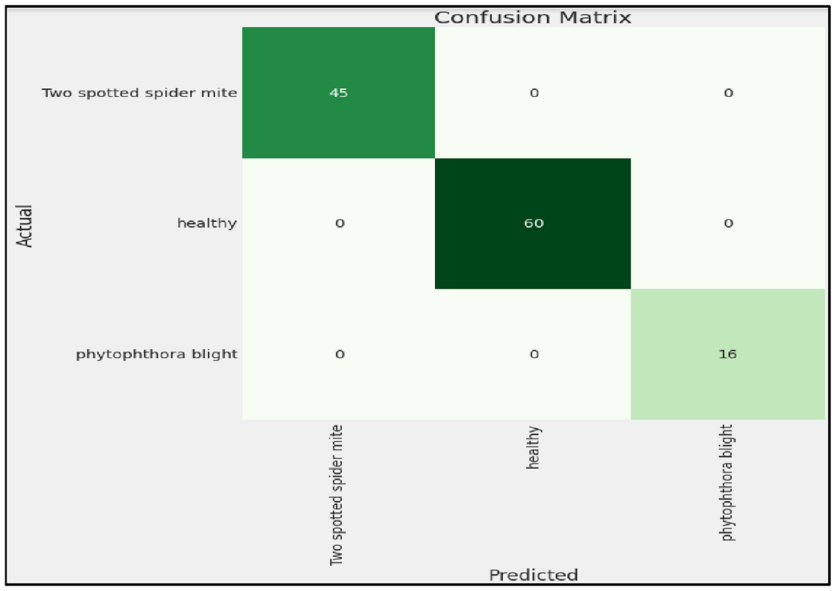

Fig. 14 Eggplant leaf disease confusion Matrix

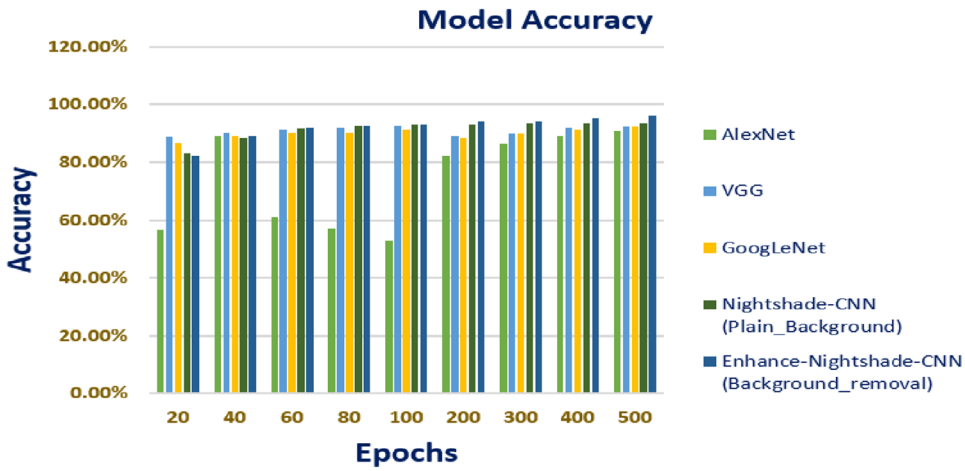

Fig. 15 Lycopersicon crop leaf disease accuracy

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

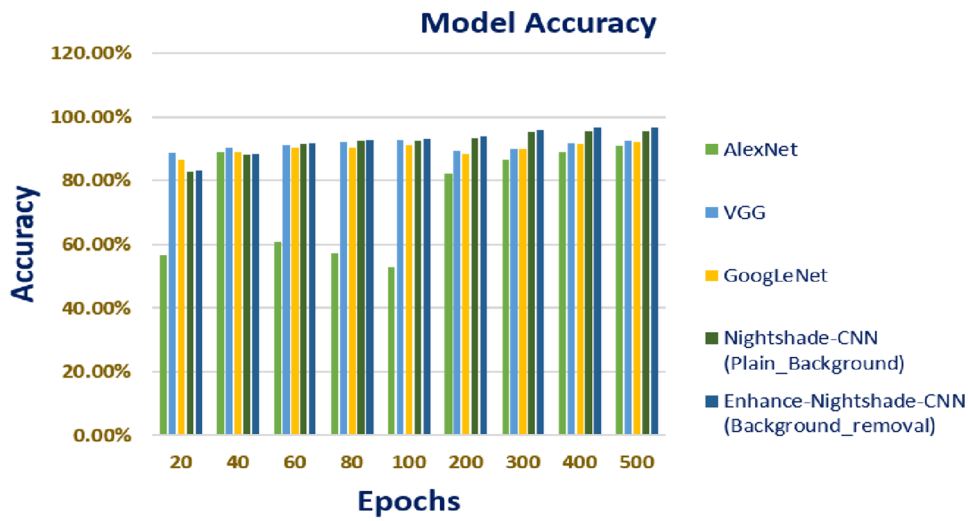

Fig. 16 Tuberosum and capsicum annuum crop leaf disease accuracy

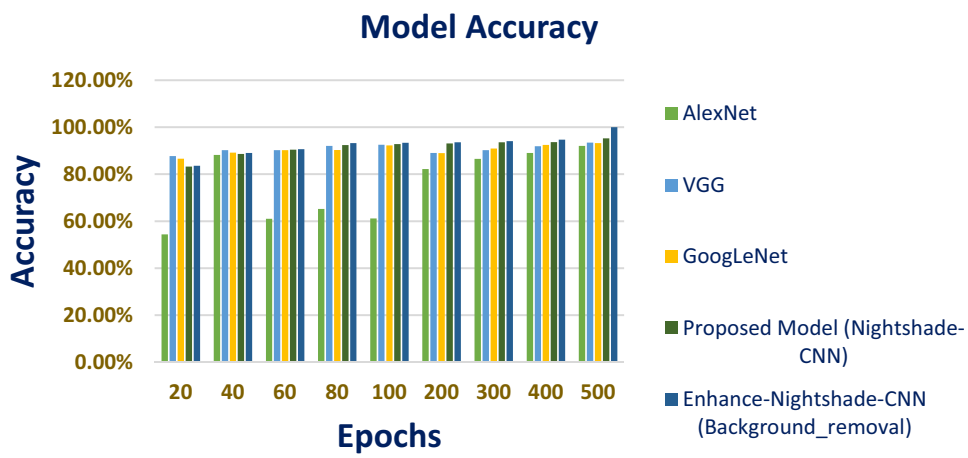

Fig. 17 Melongena crop leaf disease accuracy

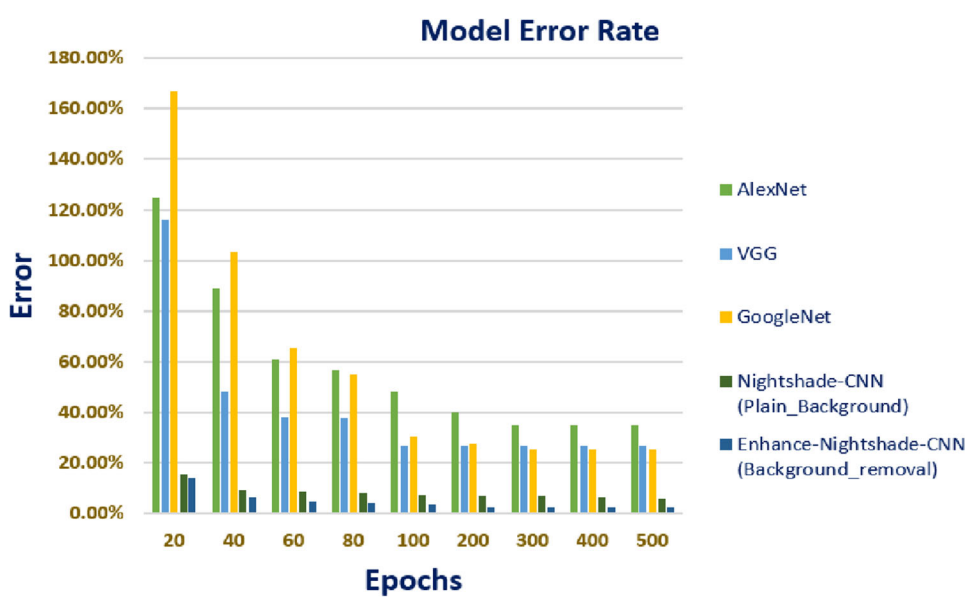

Fig. 18 Lycopersicon crop leaf disease error rate

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

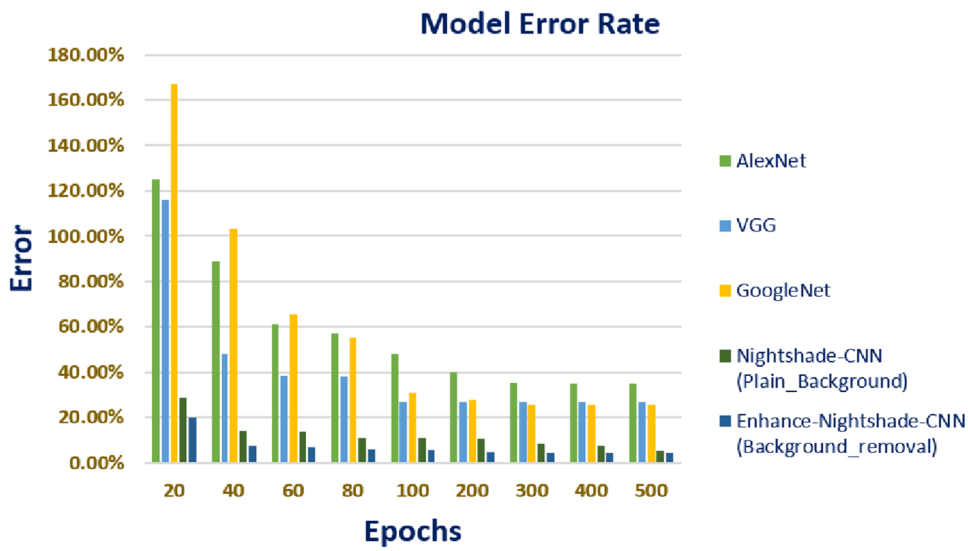

Fig. 19 Tuberosum and capsicum annuum crop leaf disease error rate

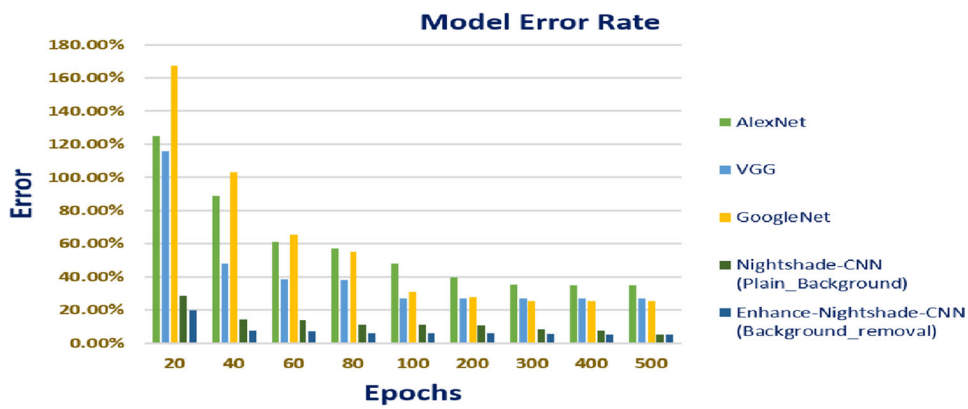

Fig. 20 Melongena crop leaf disease error rate

5 Model complexity

Convolution layers, fully-connected layers, and pooling layers are among the crucial procedures. The total number of "learnable" components in a layer is essentially examined by the number of parameters in that layer. It only supplies the form; the input layer has nothing to learn. The pooling layer has no learnable parameters, but it reduces the parameter count hence reducing the Computational complexity. Also, it helps with overfitting problems. Convolution layer is filtering images with a "smaller pixel" filter which decreases the image size without no losing the connection between pixels [35]. Output size depends on the total no of neurons, which is based on the convolution calculation you set down. A fully connected Layer is a network in which every parameter is connected to every other parameter to identify the relationship and influence of each parameter on the labels. The output size is proportional to the total number of neurons. We may lower the time complexity by selecting a number of convolutions and completely linking layers with care. Typically, the time cost of fully linked and pooling layers is 5 to 10 percent of the total calculation time. To determine the time complexity of a model, the convolution and fully linked layer time complexity must be determined [47]. Lower filter size comes with the cost of computational complexity which is very critical in the case of large datasets. Significantly higher computational costs result from large batch sizes.

Convolution neural networks required more resources and time to train the model. So space complexity and time complexity is the main issue. In this research proposed Night-CNN [32] and Enhance-Nightshade-CNN required the 11 × 11, 3 × 3, and 1 × 1 filter size and a total of six layers, hence it generates the less no of parameters so the space complexity is cAD + k(cwh + 1) + 2 k(A-w + 1) (D-h + 1) where c is a channel, A indicates the width of the image, D denotes the image height, w represents hidden layer width, h signifies the hidden layer height and k indicates bias. The time complexity again depends on the GPU environment. Throughout the research, we have used the Tesla GPU environment from Google lab services, so the time complexity for the algorithm is always in Big oh(O) notation.

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

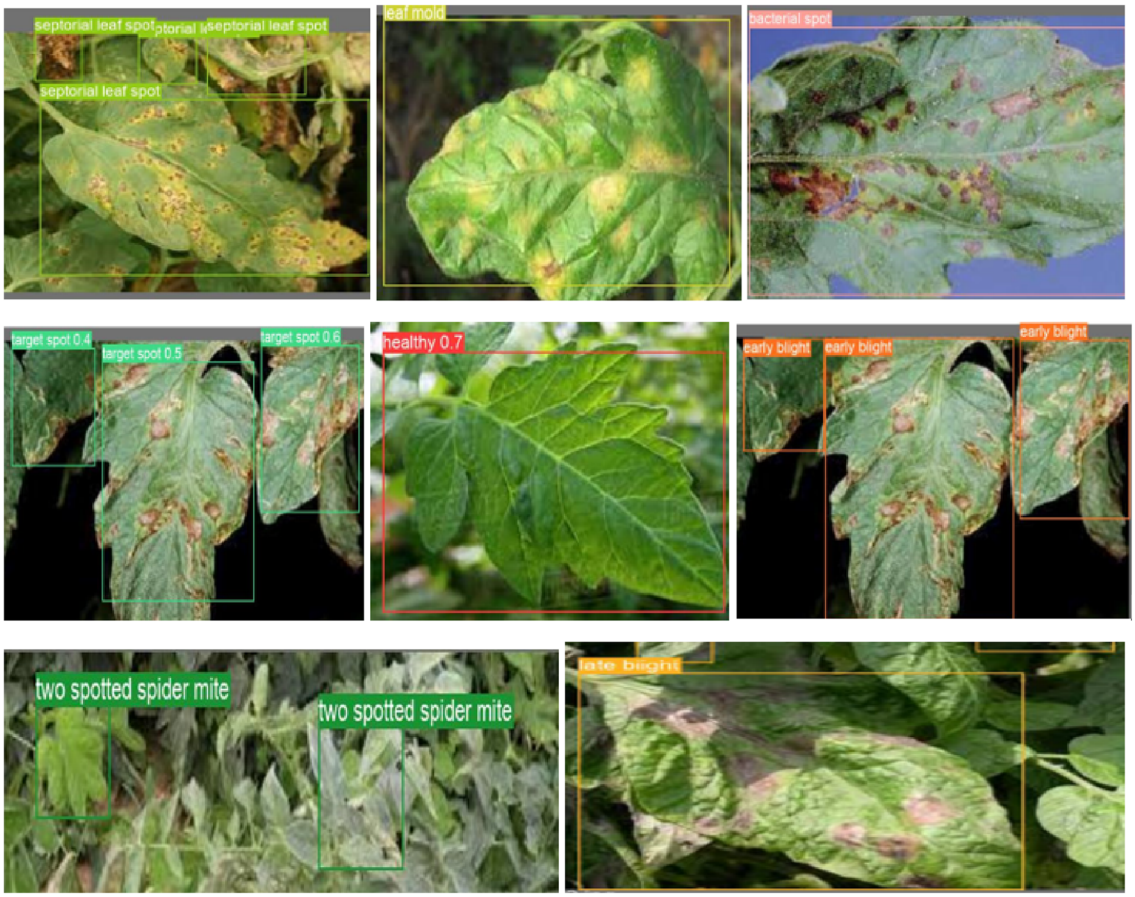

Fig. 21 Lycopersicon (Tomato) leaf diseases

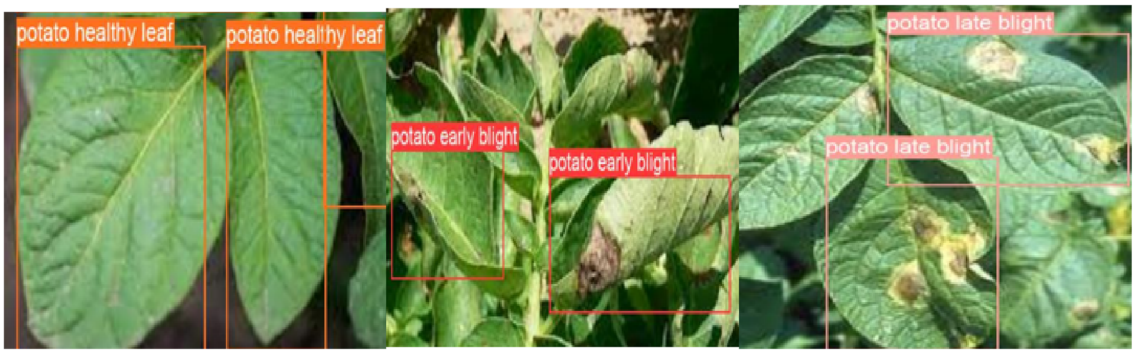

Fig. 22 Tuberosum (Potato) leaf diseases

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

Fig. 23 Capsicum annuum (Bell Pepper) leaf diseases

Fig. 24 Melongena (Eggplant) leaf diseases

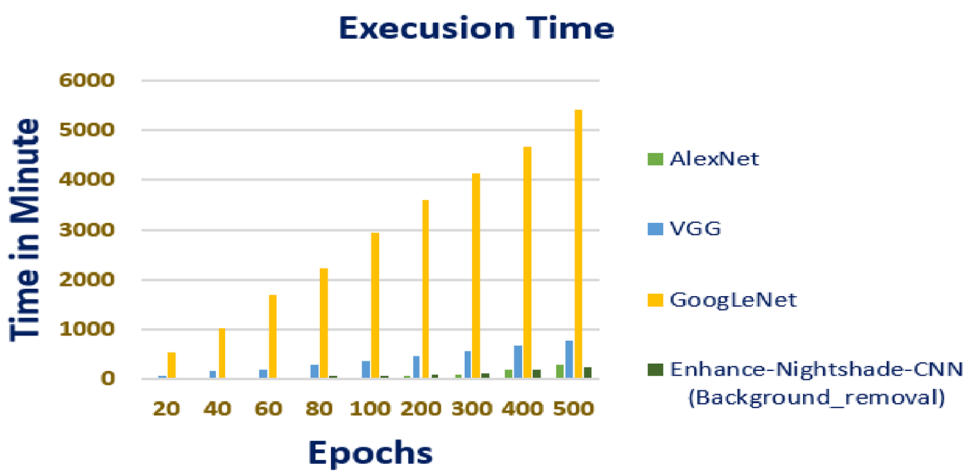

Fig. 25 Enhance-nightshade-CNN training time

6 Conclusion

The Enhance-Night-CNN required two stage to identify the leaf disease of the nightshade crops like Lycopersicon, Tuberosum, capsicum annuum, and melangena leaf disease. The model identified the healthy and unhealthy leaf disease very accurately. The model required the five-convolution layer and one fully connected layer so that it generates the less parameter hence it required to less time for the execution. The model identified the leaf disease 95–100% accurately with the 5–7% of the error rate.

Funding Not Applicable.

Data availability Plant Village repository was used for the laboratory dataset and for the real dataset we have used the google agriculture images.

Code availability Proposed CNN models has been implemented in python with the OpenCV environment and executed on Google colab environment.

Declarations

Conflicts of interest The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Ethics approval Not Applicable.

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

Consent to participate Not Applicable.

Consent for publication Dataset images we have used from the plant village repository.

References

1. Shoaib, M., Shah, B., Ali, A., Ullah, A., Alenezi, F., Gechev, T., Hussain, T., Ali, F.: An advanced deep learning models-based plant disease detection: a review of recent research. Front. Plant Sci. **14**, 1158933 (2023). https://doi.org/10.3389/fpls.2023.1158933
2. Lucas, G.B., Campbell, C.L., Lucas, L.T.: Introduction to plant diseases, p. 2006. AVI Publish. Co., Schuman GL, D'Arcy CJ, The (1985)
3. Shirahatti, J., Patil, R., and Akulwar, P.: A survey paper on plant disease identification using machine learning approach. In: 2018 3rd International Conference on Communication and Electronics Systems (ICCES) (pp. 1171–1174). IEEE. (2018). https://doi.org/10.1109/CESYS.2018.8723881.
4. Liu, L., et al.: A disease index for efficiently detecting wheat fusarium head blight using sentinel-2 multispectral imagery. IEEE Access **8**, 52181–52191 (2020). https://doi.org/10.1109/ACCESS.2020.2980310
5. Prasad, R., Ranjan, K.R., Sinha, A.K.: AMRAPALIKA: an expert system for the diagnosis of pests, diseases, and disorders in Indian mango. Knowl.-Based Syst. **19**(1), 9–21 (2006). https://doi.org/10.1016/j.knosys.2005.08.001
6. Singh, V., Misra, A.K.: Detection of plant leaf diseases using image segmentation and soft computing techniques. Inf Process Agric **4**(1), 41–49 (2017). https://doi.org/10.1016/j.inpa.2016.10.005
7. Afifi, A., Alhumam, A., Abdelwahab, A.: Convolutional neural network for automatic identification of plant diseases with limited data. Plants **10**, 28 (2021). https://doi.org/10.3390/plants10010028
8. Mugithe, P. K., Mudunuri, R. V., Rajasekar, B., & Karthikeyan, S.: Image processing technique for automatic detection of plant diseases and alerting system in agricultural farms. In: 2020 International Conference on Communication and Signal Processing (ICCSP) (pp. 1603–1607). IEEE. (2020) https://doi.org/10.1109/ICCSP48568.2020.9182065.
9. Parikshith, H., Naga Rajath, S. M., and Pavan Kumar, S. P.: Leaf disease detection using image processing and artificial intelligence–a survey. In: Computational Vision and Bio-Inspired Computing: ICCVBIC 2019 (pp. 304–311). Springer International Publishing. (2020). https://doi.org/10.1007/978-3-030-37218-7_35.
10. Sethy, P.K., Barpanda, N.K., Rath, A.K., Behera, S.K.: Deep feature based rice leaf disease identification using support vector machine. Comput. Electron. Agric. **175**, 105527 (2020). https://doi.org/10.1016/j.compag.2020.105527
11. Mitra, D.: Emerging plant diseases: research status and challenges. Emerg Trends Plant Pathol (2021). https://doi.org/10.1007/978-981-15-6275-4_1
12. Ferentinos, K.P.: Deep learning models for plant disease detection and diagnosis. Comput. Electron. Agric. **145**, 311–318 (2018). https://doi.org/10.1016/j.compag.2018.01.009
13. Ramcharan, A., Baranowski, K., McCloskey, P., Ahmed, B., Legg, J., Hughes, D.P.: Deep learning for image-based cassava disease detection. Front. Plant Sci. **8**, 293051 (2017). https://doi.org/10.3389/fpls.2017.01852
14. Vadivel, T., Suguna, R.: Automatic recognition of tomato leaf disease using fast enhanced learning with image processing. Acta Agric Scandinavica, Sect B Soil Plant Sci **72**(1), 312–324 (2022)
15. Sardogan, M., Tuncer, A., and Ozen, Y.: Plant leaf disease detection and classification based on CNN with LVQ algorithm. In: 2018 3rd International Conference on Computer Science and Engineering (UBMK) (pp. 382–385). IEEE. (2018). https://doi.org/10.1109/UBMK.2018.8566635.
16. Dhakate, M., & Ingole, A. B.: Diagnosis of pomegranate plant diseases using neural network. In: 2015 Fifth National Conference on Computer Vision, Pattern Recognition, Image Processing and Graphics (NCVPRIPG) (pp. 1–4). IEEE. (2015). https://doi.org/10.1109/NCVPRIPG.2015.7490056.
17. Sujithra, J., and Ukrit, M. F.: CRUN-based leaf disease segmentation and morphological-based stage identification. Math Probl Eng (2022).https://doi.org/10.1155/2022/2546873.
18. Francis, J., and Anoop, B. K.: Identification of leaf diseases in pepper plants using soft computing techniques. In: 2016 conference on emerging devices and smart systems (ICEDSS) (pp. 168–173). IEEE. (2016) https://doi.org/10.1109/ICEDSS.2016.7587787.
19. Sannakki, S. S., Rajpurohit, V. S., Nargund, V. B., and Kulkarni, P.: Diagnosis and classification of grape leaf diseases using neural networks. In: 2013 Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT) (pp. 1–5). IEEE. (2013). https://doi.org/10.1109/ICCCNT.2013.6726616.
20. Sankaran, S., Mishra, A., Ehsani, R., Davis, C.: A review of advanced techniques for detecting plant diseases. Comput. Electron. Agric. **72**(1), 1–13 (2010). https://doi.org/10.1016/j.compag.2010.02.007
21. Lu, L., Xiaoting, Y., Bo, D.: A fast segmentation algorithm of pet images based on visual saliency model. Procedia Comput Sci **92**, 361–370 (2016). https://doi.org/10.1016/j.procs.2016.07.391
22. Lucas, G. B., Campbell, C. L., and Lucas, L. T.: Introduction to plant diseases: identification and management. Springer Science and Business Media (1992)
23. Albahli, S., Masood, M.: Efficient attention-based CNN network (EANet) for multi-class maize crop disease classification. Front. Plant Sci. **13**, 1003152 (2022). https://doi.org/10.3389/fpls.2022.1003152
24. Diaz-Arias, A., Shin, D.: ConvFormer: parameter reduction in transformer models for 3D human pose estimation by leveraging dynamic multi-headed convolutional attention. Vis. Comput. (2023). https://doi.org/10.1007/s00371-023-02936-5
25. Ashwinkumar, S., Rajagopal, S., Manimaran, V., Jegajothi, B.: Automated plant leaf disease detection and classification using optimal MobileNet based convolutional neural networks. Mater Today: Proceed **51**, 480–487 (2022). https://doi.org/10.1016/j.matpr.2021.05.584
26. Hyvärinen, A.: Independent component analysis: recent advances. Philosoph Trans Royal Soc: Math Phys Eng Sci **371**(1984), 20110534 (2013). https://doi.org/10.1098/rsta.2011.0534
27. Jagiello, K., Grzonkowska, M., Swirog, M., et al.: Advantages and limitations of classic and 3D QSAR approach in nano-QSAR studies based on biological activity of fullerene derivatives. J. Nanopart. Res. **18**, 256 (2016). https://doi.org/10.1007/s11051-016-3564-1"
28. Jadhav, S., Lal, A.M.: Multi-class plant leaf disease detection using a deep convolutional neural network. Int J Inf Syst Model Des (IJISMD) **13**(1), 1–14 (2022). https://doi.org/10.4018/IJISMD.315126
29. Sladojevic, S., Arsenovic, M., Anderla, A., Culibrk, D., and Stefanovic, D.: Deep neural networks based recognition of plant diseases by leaf image classification. Comput. Intell. Neurosci. (2016). https://doi.org/10.1155/2016/3289801
30. Tang, Y., Qiu, J., Zhang, Y., et al.: Optimization strategies of fruit detection to overcome the challenge of unstructured background in field orchard environment: a review. Precision Agric. **24**, 1183–1219 (2023). https://doi.org/10.1007/s11119-023-10009-9
31. Joshi, B.M., Bhavsar, H.: Plant leaf disease detection and control: a survey. J. Inf. Optim. Sci. **41**(2), 475–487 (2020). https://doi.org/10.1080/02522667.2020.1734295

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

32. Joshi, B. M., & Bhavsar, H.: Lycopersicon crop leaf disease identification using deep learning. In: International Conference on Sustainable and Innovative Solutions for Current Challenges in Engineering and Technology (pp. 15–32). Singapore: Springer Nature Singapore. (2022)
33. Joshi, B.M., Bhavsar, H.: Deep learning technology based night-CNN for nightshade crop leaf disease detection. Int J Intell Syst Appl Eng **11**(1), 215–227 (2023)
34. Pandian, J.A., Kumar, V.D., Geman, O., Hnatiuc, M., Arif, M., Kanchanadevi, K.: Plant disease detection using deep convolutional neural network. Appl. Sci. **12**, 6982 (2022). https://doi.org/10.3390/app12146982
35. Pantazi, X.E., Tamouridou, A.A., Alexandridis, T.K., Lagopodi, A.L., Kontouris, G., Moshou, D.: Detection of Silybum marianum infection with Microbotryum silybum using VNIR field spectroscopy. Comput. Electron. Agric. **137**, 130–137 (2017). https://doi.org/10.1016/j.compag.2017.03.017
36. Li, S., Feng, Z., Yang, B., Li, H., Liao, F., Gao, Y., Liu, S., Tang, J., Yao, Q.: An intelligent monitoring system of diseases and pests on rice canopy. Front. Plant Sci. **13**, 972286 (2022). https://doi.org/10.3389/fpls.2022.972286
37. Pearson, K.: LIII. On lines and planes of closest fit to systems of points in space. London, Edinburgh, Dublin Philosoph Magaz J Sci **2**(11), 559–572 (1901)
38. Hu, X., Chu, L., Pei, J., et al.: Model complexity of deep learning: a survey. Knowl. Inf. Syst. **63**, 2585–2619 (2021). https://doi.org/10.1007/s10115-021-01605-0
39. Hassan, S.M., Maji, A.K., Jasiński, M., Leonowicz, Z., Jasińska, E.: Identification of plant-leaf diseases using cnn and transfer-learning approach. Electronics **10**, 1388 (2021). https://doi.org/10.3390/electronics10121388
40. Singh, V., and Misra, A. K.: Detection of unhealthy region of plant leaves using image processing and genetic algorithm. In: 2015 International Conference on Advances in Computer Engineering and Applications (pp. 1028–1032). IEEE. (2015). https://doi.org/10.1109/ICACEA.2015.7164858.
41. Guadagna, P., Fernandes, M., Chen, F., et al.: Using deep learning for pruning region detection and plant organ segmentation in dormant spur-pruned grapevines. Precision Agric. **24**, 1547–1569 (2023). https://doi.org/10.1007/s11119-023-10006-y
42. Rathod, A. N., Tanawal, B., and Shah, V.: Image processing techniques for detection of leaf disease. Int J Adv Res Comput Sci Softw Eng, 3(11), (2013)
43. Patil, J.K., Kumar, R.: Advances in image processing for detection of plant diseases. J Adv Bioinf Appl Res **2**(2), 135–141 (2011)
44. Camargo, A., Smith, J.S.: An image-processing based algorithm to automatically identify plant disease visual symptoms. Biosyst. Eng. **102**(1), 9–21 (2009). https://doi.org/10.1016/j.biosystemseng
45. Arivazhagan, S., Shebiah, R.N., Ananthi, S., Varthini, S.V.: Detection of unhealthy region of plant leaves and classification of plant leaf diseases using texture features. Agric. Eng. Int. CIGR J. **15**(1), 211–217 (2013)
46. Woebbecke, D.M., Meyer, G.E., Von Bargen, K., Mortensen, D.A.: Color indices for weed identification under various soil, residue, and lighting conditions. Trans. Am. Soc. Agric. Eng. **38**(1), 259–269 (1995)
47. Suykens, J., Vandewalle, J.: Least squares support vector machine classifiers. Neural. Process. Lett. **9**, 293–300 (1999). https://doi.org/10.1023/A:1018628609742
48. Duan, GR.: Analysis and design of descriptor linear systems (Vol. 23). Springer Science and Business Media. (2010)
49. Chen, J., Ma, Z.: Locally linear embedding: a review. Int. J. Pattern Recognit Artif Intell. **25**(07), 985–1008 (2011). https://doi.org/10.1142/S0218001411008993
50. Chai, R.: Otsu's image segmentation algorithm with memory-based fruit fly optimization algorithm. Complexity **2021**, 1–11 (2021). https://doi.org/10.1155/2021/5564690
51. Raju, D., Joseph, P.: Motion detection and optical flow. Int J Comput Sci Inf Technol **5**(4), 5716–5719 (2014)
52. Reshma, S., IIHansa JT: Graph cuts and active contours for texture aware image segmentation. Int J Adv Res Comput Sci Technol **2**(2), 176–180 (2014)
53. Amoda, N., Kulkarni, R.K.: Efficient image segmentation using watershed transform. Int J Comput Sci Technol **4**(2), 214–218 (2013)
54. Samet, R., Hancer, E.: A new approach to the reconstruction of contour lines extracted from topographic maps. J Vis Commun Imag Represent **23**(4), 642–647 (2012)
55. Wang, J., Li, M., Dziatkovskii, A., Hryneuski, U., Krylova, A.: Research on contour feature extraction method of multiple sports images based on nonlinear mechanics. Nonlin Eng **11**(1), 347–354 (2022). https://doi.org/10.1515/nleng-2022-0037
56. Islam, M., Dinh, A., Wahid, K., and Bhowmik, P.: Detection of potato diseases using image segmentation and multiclass support vector machine. In: 2017 IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE) (pp. 1–4). IEEE (2017). https://doi.org/10.1109/CCECE.2017.7946594.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.

Barkha M. Joshi received Master of Engineering in Computer Science and Engineering from the Gujarat Technological University, Ahmedabad, India. She is currently pursuing Ph.D. in Computer Science and Engineering, Faculty of Technology and Engineering, the Maharaja Sayajirao University of Baroda, Vadodara, India. She is currently working as an assistant professor in SVIT, Vasad. She has more than 15+ years of teaching experience in Computer Engineering Department. Her research area is deep learning, machine learning and data mining.

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

Hetal Bhavsar did her master's in Computer Science and Engineering from DD University, India, and Ph.D. in Computer Engineering from CHARUSAT University, India. She has more than 21+ years of teaching experience and is currently working as an assistant professor at the Maharaja Sayajirao University of Baroda. She has published 2 books and 28+ papers in many journals of repute and at international conferences. Her areas of research include database, data mining, deep learning and big data analysis. She is a member of CSI and ISTE professional bodies. She is guiding a number of Ph.D. scholars.

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

Terms and Conditions

Springer Nature journal content, brought to you courtesy of Springer Nature Customer Service Center GmbH ("Springer Nature").

Springer Nature supports a reasonable amount of sharing of research papers by authors, subscribers and authorised users ("Users"), for small-scale personal, non-commercial use provided that all copyright, trade and service marks and other proprietary notices are maintained. By accessing, sharing, receiving or otherwise using the Springer Nature journal content you agree to these terms of use ("Terms"). For these purposes, Springer Nature considers academic use (by researchers and students) to be non-commercial.

These Terms are supplementary and will apply in addition to any applicable website terms and conditions, a relevant site licence or a personal subscription. These Terms will prevail over any conflict or ambiguity with regards to the relevant terms, a site licence or a personal subscription (to the extent of the conflict or ambiguity only). For Creative Commons-licensed articles, the terms of the Creative Commons license used will apply.

We collect and use personal data to provide access to the Springer Nature journal content. We may also use these personal data internally within ResearchGate and Springer Nature and as agreed share it, in an anonymised way, for purposes of tracking, analysis and reporting. We will not otherwise disclose your personal data outside the ResearchGate or the Springer Nature group of companies unless we have your permission as detailed in the Privacy Policy.

While Users may use the Springer Nature journal content for small scale, personal non-commercial use, it is important to note that Users may not:

1. use such content for the purpose of providing other users with access on a regular or large scale basis or as a means to circumvent access control;
2. use such content where to do so would be considered a criminal or statutory offence in any jurisdiction, or gives rise to civil liability, or is otherwise unlawful;
3. falsely or misleadingly imply or suggest endorsement, approval , sponsorship, or association unless explicitly agreed to by Springer Nature in writing;
4. use bots or other automated methods to access the content or redirect messages
5. override any security feature or exclusionary protocol; or
6. share the content in order to create substitute for Springer Nature products or services or a systematic database of Springer Nature journal content.

In line with the restriction against commercial use, Springer Nature does not permit the creation of a product or service that creates revenue, royalties, rent or income from our content or its inclusion as part of a paid for service or for other commercial gain. Springer Nature journal content cannot be used for inter-library loans and librarians may not upload Springer Nature journal content on a large scale into their, or any other, institutional repository.

These terms of use are reviewed regularly and may be amended at any time. Springer Nature is not obligated to publish any information or content on this website and may remove it or features or functionality at our sole discretion, at any time with or without notice. Springer Nature may revoke this licence to you at any time and remove access to any copies of the Springer Nature journal content which have been saved.

To the fullest extent permitted by law, Springer Nature makes no warranties, representations or guarantees to Users, either express or implied with respect to the Springer nature journal content and all parties disclaim and waive any implied warranties or warranties imposed by law, including merchantability or fitness for any particular purpose.

Please note that these rights do not automatically extend to content, data or other material published by Springer Nature that may be licensed from third parties.

If you would like to use or distribute our Springer Nature journal content to a wider audience or on a regular basis or in any other manner not expressly permitted by these Terms, please contact Springer Nature at

onlineservice@springernature.com