ARTICLE IN PRESS

Materials Today: Proceedings xxx (xxxx) xxx

Contents lists available at ScienceDirect

Materials Today: Proceedings

journal homepage: www.elsevier.com/locate/matpr

Automated plant leaf disease detection and classification using optimal MobileNet based convolutional neural networks

S. Ashwinkumar [a,*], S. Rajagopal [a], V. Manimaran [a], B. Jegajothi [b]

[a] Department of Information Technology, National Engineering College, K.R. Nagar, Kovilpatti, Thoothukudi (Dt), Tamilnadu 628503, India
[b] Department of Information Technology, Sri Venkateswara College of Engineering, Sriperumpudur, Chennai, India

ARTICLE INFO

Article history:
Received 27 April 2021
Received in revised form 12 May 2021
Accepted 26 May 2021
Available online xxxx

Keywords:
Plant leaf disease
Deep learning
Image processing
Intelligent models
Computer vision
MobileNet
Parameter tuning

ABSTRACT

Agriculture is the major occupation in India and it loses 35% of the crop productivity annually owing to plant diseases. Earlier plant disease detection is a tedious process because of improper laboratory facilities and expert knowledge. Automated plant disease detection techniques are advantageous for reducing the laborious task of monitoring large crop farms and for identifying disease symptoms early on, i.e., when they appear on plant leaves. Recent advances in computer vision and deep learning (DL) models have demonstrated the value of developing automatic plant disease detection models based on visible symptoms on leaves. With this in mind, this article proposes an automated model for detecting and classifying plant leaf diseases using an optimal mobile network-based convolutional neural network (OMNCNN). The proposed OMNCNN model operates on different stages namely preprocessing, segmentation, feature extraction, and classification. It involves bilateral filtering (BF) based preprocessing and Kapur's thresholding based image segmentation to identify the affected portions of the leaf image. In addition, the MobileNet model is applied as a feature extraction technique in which the hyperparameters are optimized by the use of emperor penguin optimizer (EPO) algorithm to enhance the plant disease detection rate. Finally, extreme learning machine (ELM) based classifier is utilized to allocate proper class labels to the applied plant leaf images. An extensive set of simulations were performed to highlight the superior performance of the OMNCNN model. The experimental outcome has shown promising results of the OMNCNN model over the recent state-of-art methods with the maximum precision of 0.985, recall of 0.9892, accuracy of 0.987, F-score of 0.985, and kappa of 0. 985.

Selection and peer-review under responsibility of the scientific committee of the 1st International Conference on Computations in Materials and Applied Engineering – 2021.

1. Introduction

Annually the Earth's population rises by approximately 1.6%, and therefore the demand for plant products gets increased to a greater extent. The protection of crops from plant diseases plays a significant part to fulfill the increasing requirement of food quantity and quality [1]. From an economical perspective, the occurrence of plant diseases affects the world economy about US$220 billion per annum. Based on the Indian Council of Agricultural Research, more than 35% of crop productivity gets damaged annually owing to pests and diseases. Therefore, food security is endangered by a distressing rise in the quantity of pests and diseases. They have greatly affected the economic, social, and ecological activities. Accurate plant disease detection still a crucial process for agricultural people [2]. They do not comprise several opportunities rather than getting suggestions from other farmers or the Kisan helpline. In addition, in several cases, it is needed to have a laboratory setup to detect the affected leaves. On the other hand, in few countries, the farmers do not possess adequate facilities or even recommendations from expert people. Besides, the expert's consultation is expensive and it consumes more time. Therefore, it is recommended to design a new technique to effectively monitor large crop fields. Automated disease detection with visual signs of the plant leaves is found to be simple and inexpensive.

Advancements in computer vision givens opportunities for expanding and enhancing the idea of accurate plant disease detection and elaborate the computer vision applications in the domains of precision agriculture. The exploitation of general image

* Corresponding author.
E-mail addresses: offgoogly@gmail.com (S. Ashwinkumar), rajatarget@nec.edu.in (S. Rajagopal).

https://doi.org/10.1016/j.matpr.2021.05.584
2214-7853/
Selection and peer-review under responsibility of the scientific committee of the 1st International Conference on Computations in Materials and Applied Engineering – 2021.

processing approaches like color analysis and threshold [3] are employed to detect and classify plant diseases. Distinct techniques are presented employed to detect plant diseases and some of the common methods. These methods can be integrated into distinct techniques of image pre-processing for effective feature extraction. In ML and cognitive science, ANN is a data processing model which is stimulated using the biological nervous system. The brain encompasses several interlinked neurons operating together for solving a particular problem. Prior to the development of deep learning (DL) architectures, several works have concentrated on utilizing image processing/feature extraction to design disease detection models [4,5]. The major issue of this model is the difficulty of defining the symptoms for detection using computers, which has been resolved to utilize DL models where they do not necessitate the features to be defined, however, it learns the features by the use of optimization.

Several works have been used the ability of DL models to effectively achieve distinct levels of accuracy on lab images. When the precise classification models are validated on the data equivalent to the training data, maximum accuracy can be obtained [6] defined several factors influencing the outcome of the DL models for plant leaves disease identification and established that in spite of the considerable maximum success rate of designed model, different reasons which create if still far enough from being a generic tool which can be employed in real-time environment. Many of the existing works have utilized DL models to detect diseases undergone training and testing on the Plant Village dataset that contains a set of minimum variability and identical background. In recent times [7], examined the efficiency of the DL models upon training on individual lesions and spots, utilizing image segmentation and augmentation for increasing the dataset size using considerably lower number of images.

This paper proposes an automated plant leaf disease detection and classification model using optimal mobile network based convolutional neural network (OMNCNN) model. The presented model encompasses bilateral filtering (BF) based preprocessing and Kapur's thresholding based image segmentation to identify the affected portions of the leaf image. Besides, the MobileNet model is utilized to derive feature vectors and then hyper parameter optimization process takes place using emperor penguin optimizer (EPO) algorithm. The hyper parameter optimization actually aims to adjust the hyper parameters of the MNCNN model in such a way that the classification performance gets increased to a maximum extent. Lastly, extreme learning machine (ELM) based classifier is exploited to allot appropriate class labels to the applied plant leaf images. A comprehensive simulation analysis takes place to ensure the better performance of the OMNCNN model.

The rest of the paper is arranged as follows. Section 2 briefs the existing works and section 3 introduces the proposed OMNCNN model. Next, section 4 performs the simulation analysis and section 5 concludes the paper.

2. Literature review

Sladojevic et al. [3] presented a novel plant leaf disease detection and classification model using deep CNN (DCNN). The DCNN model is utilized for training and simple implementation in practice. The enhanced method can detect thirteen distinct kinds of plant diseases based on healthier leaves, with capacity for differentiating plant leaves from environments. Guo et al. [8] established a plant disease recognition and detection depending upon DL that enhances generality, training, and accuracy performance. Initially, the region proposal network (RPN) is used for localizing and recognizing the leaves in a complicated environment. Next, the image segmentation is depending upon the outcomes of the RPN technique, which contains symptom features via Chan–Vese (CV) technique. In Selvaganesan et al. [9], a computer vision architecture is established by framing a method which comprises image acquisition, feature extraction, and classification. The Deep Belief Network (DBN) is utilized for classifying real-time images. Khatoon et al. [10] proposed an end-to-end system for diagnosing basic crop challenges from real time and guaranteeing higher accuracy. This study utilizes several DL methods for recognizing and predicting various diseases affected by nutrition deficiency, pathogens, and pests. Numerous CNN methods are trained for huge dataset of leaves and fruit images from tomato plants.

Chen et al. [11] investigated a TL of DCNN method for detecting plant leaf disease and considered a pre-trained method learned from huge datasets, and later transmit to the certain process trained by individual data. The VGGNet is pre-trained on ImageNet and Inception module is chosen from this technique. Bharali et al. [12] utilize DCNN for detecting plant diseases from images of plant leaves and precisely categorize them as to two types depending upon the absence and presence of diseases. A smaller NN is trained by a smaller dataset of 1400 images that attains 96.6% of accuracy. The network is made by Keras for running on-top of the DL architecture TensorFlow. Khatoon et al. [13] utilize several DL methods for recognizing and predicting various diseases affected by nutrition deficiency, pests, and pathogens.

Xie et al. [14] proposed a real time detector for grape leaf disease using enhanced DCNN method. Initially, this method increases the grape leaf diseases image via image processing technique, which creates grape leaf disease dataset (GLDD). According to GLDD and Faster R-CNN detection method, a DL based Faster DR-IACNN method with high feature extraction ability is introduced for identifying the grape leaf diseases via presenting the stacked encoder (SE) blocks Inception-v1 and Inception-ResNet-v2 modules. Zhang et al. [15] proposed an enhanced Faster RCNN for detecting healthier tomato leaves and 4 diseases such as ToMV, leaf mold fungus, blight, and powdery mildew. Janarthan et al. [16] proposed a fast, lightweight, and precise deep metric learning-based architecture for detecting citrus disease from sparse data. Especially, they proposed a patch based classification network which consists of simple NN classification, cluster prototype, and embedding modules, to identify accurate citrus disease. Barman et al. [17] proposed 2 kinds of CNN frameworks like Self-Structured (SSCNN) and MobileNet classifications for detecting and classifying citrus leaf diseases at the vegetative phase. Both the methods have been tested and trained on a similar citrus dataset.

3. The proposed OMNCNN model

The overall system architecture of the OMNCNN model is illustrated in Fig. 1. The figure demonstrates that the input plant leaf image is initially fed into the BF technique to preprocess the image and enhance the quality. Followed by, Kapur's thresholding model is employed to segment the diseased portions that exist in the preprocessed plant leaf images. Next, the MNCNN method is applied to derive a useful set of feature vectors and the EPO algorithm is utilized to determine the hyper parameters of the MNCNN model. At last, the extracted feature vectors are given as input to the ELM classifier to determine the existence of the diseases with its appropriate class labels. The detailed working of these processes is provided in the succeeding subsections.

3.1. Image preprocessing using BF technique

At the initial stage, the input plant leaf image is given as input to the BF technique for removing the noise that exists in it and

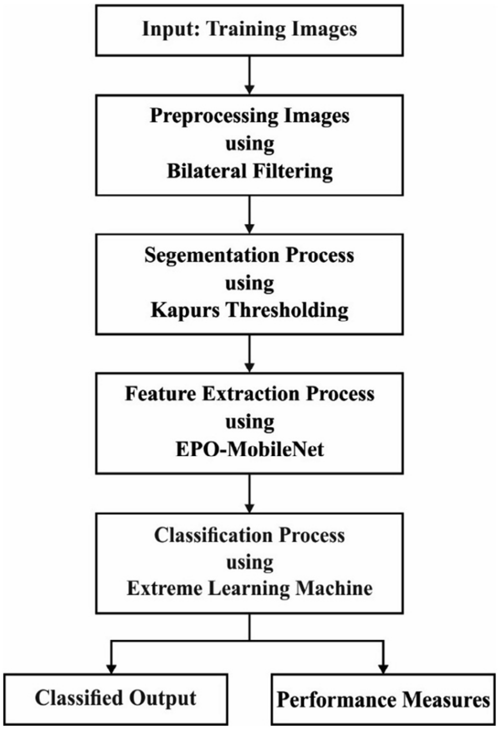

Fig. 1. Working process of OMNCNN model.

thereby enhances the image quality. BF is a non-linear filter which attains the effect of preserving edges and smoothening [18]. The weight of BF assumed the Euclidean distance of pixel with the resemblance among the neighborhood and intermediate pixels. The BF is also a weighted average technique, demonstrating the pixel intensity with weighted average of ambient pixel brightness. The Space weight is commonly utilized as a weighting estimation technique of Gaussian filter by evaluating the distance among2 pixels as given by:

$$g(\alpha,\beta)=\frac{\sum\gamma,\delta f(\gamma,\delta)\omega(\alpha,\beta,\gamma,\delta)}{\sum\gamma,\delta\omega(\alpha,\beta,\gamma,\delta)} \tag{1}$$

The weighting coefficient $\omega(\alpha,\beta,\gamma,\delta)$ is based on the kernel product of range and domain. The domain kernel is given by:

$$d(\alpha,\beta,\gamma,\delta)=\exp\left(-\frac{(\alpha-\gamma)^2+(\beta-\delta)^2}{2\sigma_{d^2}}\right) \tag{2}$$

Besides, the range kernel is given by:

$$r(\alpha,\beta,\gamma,\delta)=\exp\left(-\frac{f(\alpha,\beta)-f(\gamma,\delta)^2}{2\sigma_{d^2}}\right) \tag{3}$$

When the 2 kernels are multiplied, the BF weight functions are created using Eq. (4):

$$\omega(\alpha,\beta,\gamma,\delta)=\exp\left(--\frac{(\alpha-\gamma)^2+(\beta-\delta)^2}{2\sigma_{d^2}}-\frac{f(\alpha,\beta)-f(\gamma,\delta)^2}{2\sigma_{d^2}}\right) \tag{4}$$

3.2. Image segmentation using Kapur's thresholding

During image segmentation, Kapur's thresholding based segmentation technique is executefor determining the infected leaf portions of the image. It is employed for determining optimal thresholds to segment images. It mainly depends upon entropy and probability distribution of the image histogram [19]. It is employed for determining the optimum threshold (th) values which maximize the total entropy. In order to the bi-level instance, the objective function of Kapur's issue is represented using Eq. (5):

$$F_{kapur}(th)=H_1+H_2 \tag{5}$$

where entropy H_1 and H_2 can be defined by:

$$H_1=\sum_{i=1}^{th}\frac{Ph_i}{\omega_0}ln\left(\frac{Ph_i}{\omega_0}\right)andH_2=\sum_{i=th+1}^{L}\frac{Ph_i}{\omega_1}ln\left(\frac{Ph_i}{\omega_1}\right) \tag{6}$$

where Ph_i is the likelihood distribution of the intensity level, ω_0 (th) and ω_1 (th) are probability distribution for the class labels C_1 and $C_2.ln(.)$ is the natural logarithm. The objective function for multi-level thresholding is altered as given in Eq. (7):

$$F_{kapur}(TH)=\sum_{i=1}^{k}H_i \tag{7}$$

where $TH=[thl,th_2,th_(k-1)]$is a vector comprising many threshold values. The entropies are determined individually with the corresponding (th) value, thus Eq. (8) is extended to k entropies.

$$H_k^c=\sum_{i=th_{k+1}}^{L}\frac{Ph_i}{\omega_{k-1}}ln\left(\frac{Ph_i}{\omega_{k-1}}\right) \tag{8}$$

3.3. Feature extraction using MNCNN model

The segmentation leaf images are fed into the MNCNN model and it generates a useful set of feature vectors. MobileNet is an efficient architectural model which exploits depth-wise separable convolution for the construction of lightweight DCNN, which find useful for mobile and computer vision applications. It is compact, easier computation, and achieves maximum performance. Depending upon the depth-wise separable convolution, MobileNet utilizes 2 global hyperparameters for balancing the tradeoff between performance and accuracy. The main concept of MobileNet is to decompose the convolutional kernels. Utilizing the depth-wise separable convolutions, the regular convolution undergoes decomposition to depth-wise convolutions and point-wise convolutions with 1×1 convolutional kernel [20]. The depth-wise convolutional filter carries out convolutions for every channel, and 1×1 convolution is employed for integrating the outcome of the depth-wise convolutional layers. In this manner, N regular convolutional kernels get replaced using M depth-wise convolutional kernels and N point-wise convolution kernels. Fig. 2 illustrates the architecture of MobileNet. A typical convolution filter integrates the inputs to a new set of output, whereas the depth-wise separable convolution partitions the input into two layers (filtering and merging).

3.4. Parameter optimization using EPO algorithm

For tuning the hyper parameters of the MNCNN model, EPO technique is used and thereby enhances the classification performance of the proposed model. The goal of parameter optimization is to actually adjust the hyper parameters of the MNCNN model in such a way that the classification performance gets increased to a maximum extent.EPO algorithm based on the emperor penguins' (EPs) huddling attitude, as original from Antarctica [21]. The EPs generally travel in colonies for foraging. The unusual characteristic of the animals at the time of foraging is the huddling nature. So, the major aim is to determine a proficient mover from the ground in a mathematical way. The distance among the EPs (X_{ep}) are determined subsequent to the temperature profile θ' The effective mover is represented and the position of other EPs is altered for achieving optimal values. The steps involved in EPO are defined

	Type / Stride	Filter Shape	Input Size
	Conv / s2	3 x 3 x 3 x 32	224 x 224 x 3
	Conv dw / s1	3 x 3 x 32 dw	112 x 112 x 32
	Conv / s1	1 x 1 x 32 x 64	112 x 112 x 32
	Conv dw / s2	3 x 3 x 64 dw	112 x 112 x 64
	Conv / s1	1 x 1 x 64 x 128	56 x 56 x 64
	Conv dw / s1	3 x 3 x 128 dw	56 x 56 x 128
	Conv / s1	1 x 1 x 128 x 128	56 x 56 x 128
	Conv dw / s2	3 x 3 x 128 dw	56 x 56 x 128
	Conv / s1	1 x 1 x 128 x 256	28 x 28 x 128
	Conv dw / s2	3 x 3 x 256 dw	28 x 28 x 256
	Conv / s1	1 x 1 x 256 x 256	28 x 28 x 256
	Conv dw / s1	3 x 3 x 256 dw	28 x 28 x 256
	Conv / s1	1 x 1 x 256 x 512	14 x 14 x 256
5 x	Conv dw / s1	3 x 3 x 512 dw	14 x 14 x 512
	Conv / s1	1 x 1 x 512 x 512	14 x 14 x 512
	Conv dw / s2	3 x 3 x 512 dw	14 x 14 x 512
	Conv / s1	1 x 1 x 512 x 1024	7 x 7 x 512
	Conv dw / s2	3 x 3 x 1024 dw	7 x 7 x 1024
	Conv / s1	1 x 1 x 1024 x 1024	7 x 7 x 1024
	Avg Pool / s1	Pool 7 x 7	7 x 7 x 1024
	FC / s1	1024 x 7	1 x 1 x 1024
	PNN / s1	Classifier	1 x 1 x 5

Fig. 2. Architecture of MobileNet.

as follows. The temperature profile of the EPs is represented by:

$$\theta' = \left(\theta - \frac{Iter_{\max}}{C - Iter_{\max}}\right) \tag{9}$$

$$\theta = \begin{cases} 0 & if\ R > 0.5 \\ 1 & if\ R < 0.5 \end{cases} \tag{10}$$

The maximal number of rounds, where C denotes the existing round as defined by *Iter_max* and R is the random number lies in the range of $[0, 1]$.As EPs generally huddle collectively for preserving temperature, cautious precaution needs to be considered for protecting them from neighboring collision. Therefore, a set of 2 vectors $\left(\vec{U}\right)$ and $\left(\vec{V}\right)$ whose values are computed by:

$$\vec{U} = \left\{M \times \left(\theta' + X_{grid}(accuracy)\right) \times Rand0\right\} - \theta \tag{11}$$

$$\vec{V} = Rand() \tag{12}$$

$$X_{grid}(accuracy) = \left|\vec{X} - \vec{X}_{ep}\right| \tag{13}$$

Where M signifies the parameter for movement set as 2, $\overset{?}{x}$ denotes optimal solution, $\overset{?}{x}_{ep}$ specifies the location of other EPs, [0,1] and || is the absolute value for *Rand*.

$$\vec{D} = \left|\left\{S\left(\vec{U}\right) \cdot \vec{X}(x) - \vec{V} \cdot \vec{X}_{ep}(x)\right\}\right| \tag{14}$$

$$S\left(\vec{U}\right) = \sqrt{(fe^{-C/v} - e^{-C})^2} \tag{15}$$

Eqs. (14) and (15) are derived for estimating the distance among the EP and optimal fittest searching agent $\vec{D}$ $S()$ displays the social forces in which the optimum searching agents are followed by EPs and, *e* indicates the exponential function. Based on the optimum agents attained by the use of Eq. (16), the position of the EPs can be upgraded.

$$\vec{X}_{ep}(x+1) = \vec{X}(x) - \vec{U} \cdot \vec{D}_{ep} \tag{16}$$

In EPO, with arbitrarily produced distinct EPs, the EP population is initialized.

3.5. Image classification using ELM model

Finally, the ELM model is used to classify the leaf images into diseased or non-diseased images and allocate proper class labels to the test images. ELM is a single hidden layer feedforward neural network (FFNN) that can be enhanced from the gradient technique. The ELM model does not necessitate parameter update and it only needs arbitrary connection weight initialization among the input and output layers prior to training the data, and bias parameters in the hidden layer. Once the neuron count in the hidden layer is fixed, then optimum solutions can be achieved. It reduces the data computation time and enhances the learning rate [22]. The ELM model includes a serial connection of input, hidden, and output layers. In ELM, there are N arbitrary instances (X_i, t_i), where $X_i = [x_{1i}, x_{2i}, \cdots, x_{ni}]^T \epsilon R^n, t_i = [t_{1i}, t_{2i}, \cdots, t_{qi}, \cdots, t_{mi}]^T \epsilon R^m$. The ELM with L hidden nodes is defined using Eq. (17):

$$\sum_{i=1}^{L} \beta i g(Wi \cdot Xj + b_i) = o_j, j = 1, \cdots, N, \tag{17}$$

where $g(x)$ denotes the activation function of hidden neuron, $W_i = [w_{i,1}, w_{i,2}, \cdots, w_{i,n}]^T$ represents input weights, βi is the output weights, b_i is the i^{th} neuron bias in hidden layer. The goal of ELM is to reduce the error among the actual and required outcomes, as defined in Eq. (18):

$$\sum_{j=1}^{N} \|o_j - t_j\| = 0. \tag{18}$$

$$H\beta = T, \tag{19}$$

where H is the outcome matrix of hidden layer, β and T indicates the actual and expected outcomes. When the input weight W_i and the biases of the hidden layer b_i are arbitrarily found, the output matrix of the hidden layer H is exclusively computed. The ELM can perform both binary and multi-class classification.

Table 1
Dataset Descriptions.

Tomato Dataset	
Disease	Number of Images
Early_Blight	1000
Late_Blight	1909
Leaf_Mold	952
Target Spot	60
Healthy	1591
Total Images	**5512**

4. Experimental validation

This section examines the plant disease detection performance of the proposed OMNCNN model on the benchmark plant leaf disease dataset. The experimental results are investigated interms of different measures. The details related to implementation and results analysis are offered in the subsequent sections.

4.1. Implementation data

The presented OMNCNN mode is experimented with using a PC i5-8600 k, GeForce 1050Ti 4 GB, 16 GB RAM, 1 TB HDD. The simulation tool used is Python – 3.6.5, with some packages namely tensorflow (GPU-CUDA Enabled), keras, numpy, pickle, matplotlib, sklearn, pillow, and opencv-python. The OMNCNN model is tested against benchmark tomato leaf disease dataset [23]. It includes a set of 5452 images of normal and diseased leaves (Early_Blight, Late_Blight, Leaf_Mold, and Target spot). The details related to the dataset are given in Table 1 and sample test images are provided in Fig. 3.

Fig. 3. Sample Images Tomato Dataset.

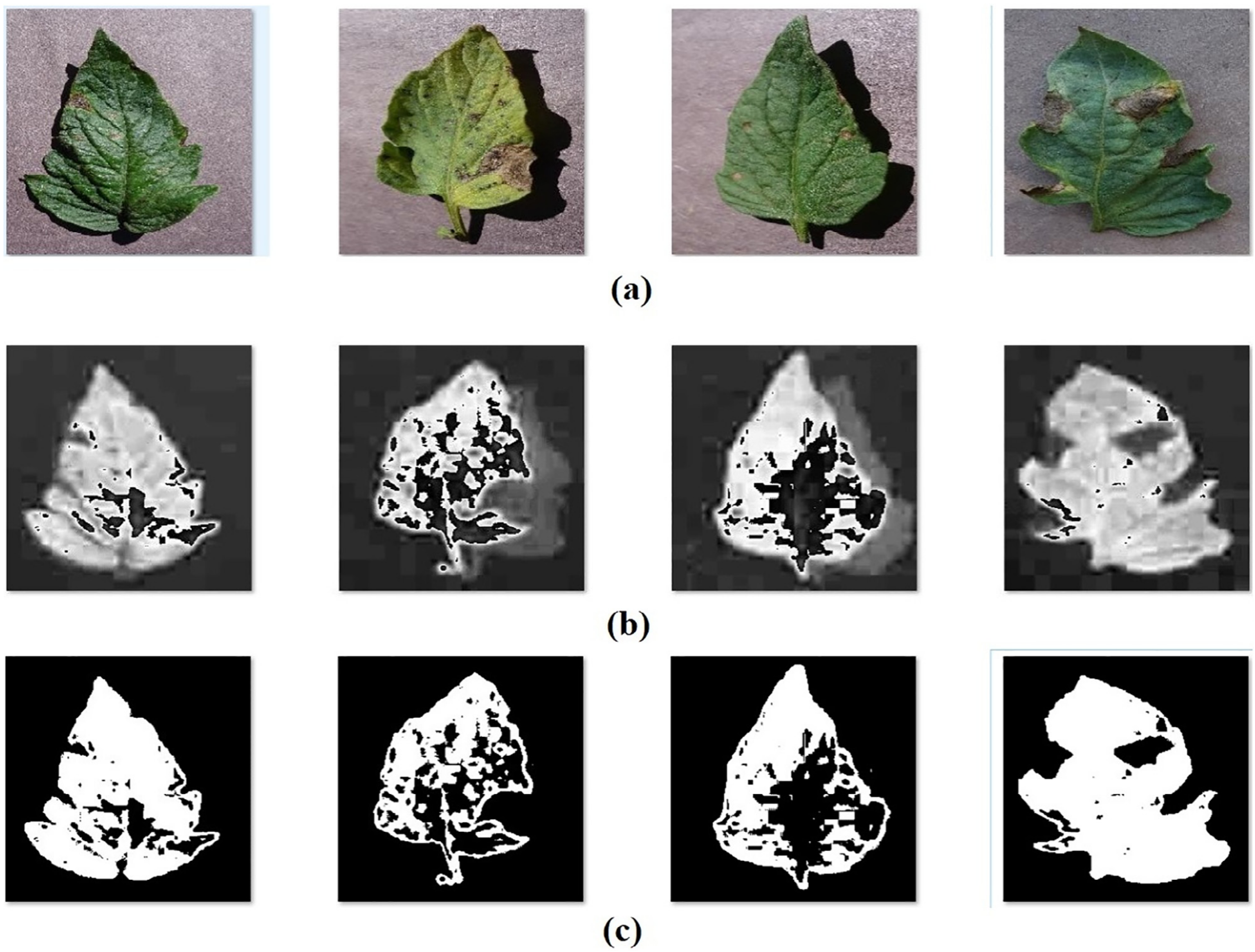

Fig. 4. Sample visualization results of the OMNCNN model.

Table 2
Results of Proposed OMNCNN Method on different classes of Applied Dataset.

Classes	Precision	Recall	Accuracy	F-score	Kappa
Early_Blight	0.979	0.987	0.985	0.975	0.975
Late_Blight	0.985	0.989	0.987	0.989	0.988
Leaf_Mold	0.987	0.986	0.985	0.985	0.986
Target Spot	0.989	0.989	0.989	0.988	0.984
Healthy	0.987	0.992	0.990	0.989	0.990
Average	**0.985**	**0.989**	**0.987**	**0.985**	**0.985**

4.2. Results analysis

Fig. 4 visualizes the otucomes obtained by the presentedOMNCNNmethod on the applied test tomato leaf images. Fig. 4a shows the input test tomato leaf images, Fig. 4b shows the pre-processed tomato leaf images, and its segmented versions are given in Fig. 4c.

The classification results analysis of the OMNCNN model on the tomato leaf image dataset takes place in Table 2 and Figs. 5–6.From the results, it is clear that the OMNCNN method has demonstrated effective outcomes on the applied dataset. For instance, the OMNCNN model classifies the 'Early_Blight' disease with the precision of 0.979, recall of 0.987, accuracy of 0.985, F-score of 0.975, and kappa of 0.975. Eventually, the OMNCNN model classifies the 'Late_Blight' disease with the precision of 0.985, recall of 0.989, accuracy of 0.987, F-score of 0.989, and kappa of 0.988. Meanwhile, the OMNCNN model classifies the 'Leaf_Mold' disease with the precision of 0.987, recall of 0.986, accuracy of 0.985, F-score of 0.985, and kappa of 0.986. In addition, the OMNCNN model classifies the 'Target Spot' disease with the precision of 0.989, recall of 0.989, accuracy of 0.989, F-score of 0.988, and kappa of 0.984. Lastly, the OMNCNN model classifies the 'Healthy' disease with the precision of 0.987, recall of 0.992, accuracy of 0.990, F-score of 0.989, and kappa of 0.990.

A brief comparative results analysis of the OMNCNN model with recent methods takes place in Table 3 and Fig. 7 [24–26]. The results portrayed that the Quadratic-SVM and CNN techniques have showcased worst outcomes with the minimal accuracy of 0.835 and 0.84. At the same time, the CNN-LVQapproach has attained a somewhat increased accuracy value of 0.86. Besides, the Residual CNN, LeNet, and GoogleNet models have portrayed moderate outcomes with the accuracy of 0.95, 0.95, and 0.96 respectively, Moreover, the VGG-16 and AlexNet manners have resulted in near-optimal accuracy values of 0.973 and 0.975. How-

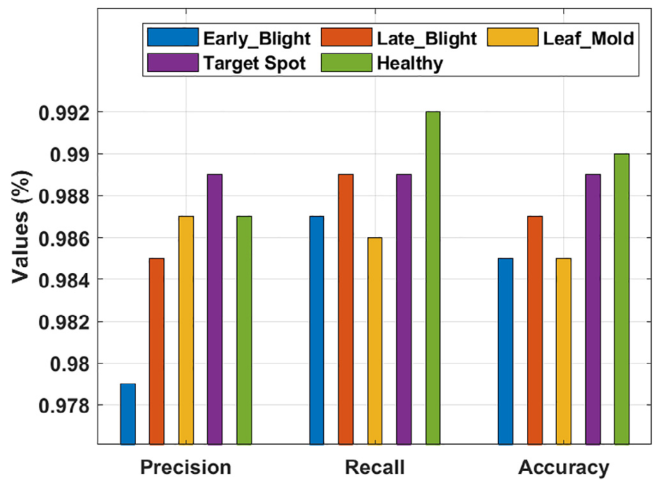

Fig. 5. Result analysis of OMNCNN model with distinct measures.

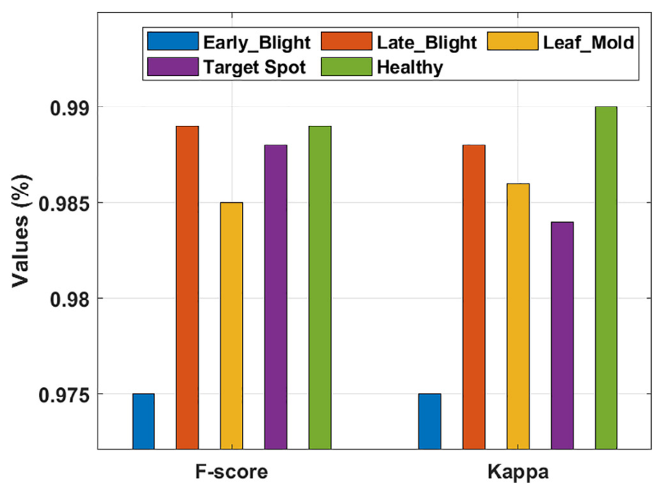

Fig. 6. F-score and kappa analysis of OMNCNN model.

Table 3
Comparative analysis of Proposed OMNCNN Method with respect to Accuracy.

Methods	Accuracy
OMNCNN	0.987
VGG-16	0.973
AlexNet	0.975
CNN	0.840
Residual CNN	0.950
Quadratic SVM	0.835
LeNet	0.950
GoogleNet	0.960
CNN-LVQ	0.860

ever, the proposed OMNCNNmethodology has accomplished maximal performance with a higher accuracy of 0.987.

From the above-mentioned tables and figures, it is evident that the OMNCNN model is found to be an appropriate tool for the detection and classification of plant leaf diseases in real-time environment.

5. Conclusion

This paper has presented an automated plant leaf disease detection and classification using OMNCNN model. The OMNCNN model

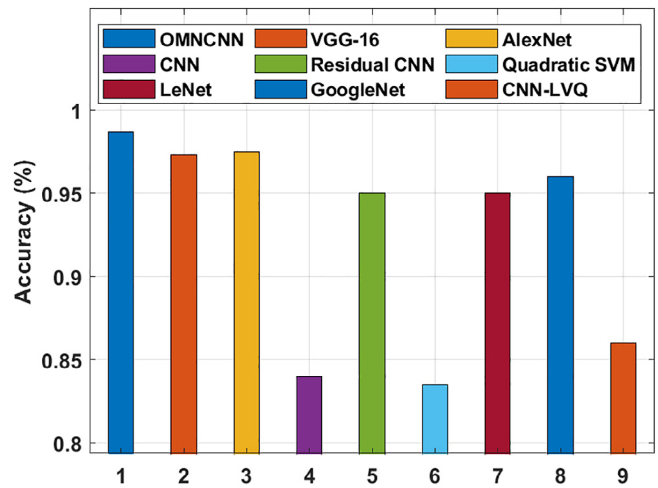

Fig. 7. Comparative analysis of OMNCNN model with existing techniques.

aims to determine the existence of plant diseases by the use of leaf images with maximum detection rate. The proposed model encompasses BF based image preprocessing, Kapur's thresholding based segmentation, MNCNN based feature extraction, EPO based parameter optimization, and ELM based classification. The EPO based hyperparameter optimization actually aims to adjust the hyperparameters of the MNCNN model in such a way that the classification performance gets increased to a maximum extent. A comprehensive simulation analysis takes place to ensure the optimal performance of the OMNCNN model. The experimental outcome has shown promising results of 7 the OMNCNN model over the recent state-of-art methods interms of different measures. In the future, the detection efficiency of the OMNCNN method is improvised by the utilization of advanced DL based image segmentation techniques.

CRediT authorship contribution statement

S. Ashwinkumar: Investigation, Writing - original draft. **S. Rajagopal:** Conceptualization. **V. Manimaran:** Conceptualization, Writing - review & editing, Supervision. **B. Jegajothi:** Investigation, Writing - original draft.

Declaration of Competing 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.

References

[1] R.N. Strange, P.R. Scott, Plant disease: a threat to global food security, Annu. Rev. Phytopathol. 43 (2005) (2005) 83–116.
[2] J. Uthayakumar, T. Vengattaraman, J. Amudhavel, A Simple Lossless Compression Algorithm in Wireless Sensor Networks: An Application of Wind Plant Data, IIOAB J. 8 (2) (2017) 281–288 (ESCI - Web of Science - Thomson Reuters).
[3] S. Sladojevic, M. Arsenovic, A. Anderla, D. Culibrk, D. Stefanovic, Deep neural networks based recognition of plant diseases by leaf image classification, Comput. Intell. Neurosci. (2016).
[4] P. Sharma, Y.P.S. Berwal, W. Ghai, Performance analysis of deep learning CNN models for disease detection in plants using image segmentation, Inf. Process. Agric. 7 (4) (2020) 566–574.
[5] S.K. Lakshmanaprabu, S.N. Mohanty, S. Krishnamoorthy, J. Uthayakumar, K. Shankar, Online clinical decision support system using optimal deep neural networks, Appl. Soft Comput. 81 (2019) 105487, https://doi.org/10.1016/j.asoc.2019.105487.

[6] J.G.A. Barbedo, Impact of dataset size and variety on the effectiveness of deep learning and transfer learning for plant disease classification, Comput. Electron. Agric. 153 (2018) 46–53.
[7] J.G.A. Barbedo, Plant disease identification from individual lesions and spots using deep learning, BiosystEng 180 (2019) 96–107.
[8] Y. Guo, J. Zhang, C. Yin, X. Hu, Y. Zou, Z. Xue, W. Wang, Plant Disease Identification Based on Deep Learning Algorithm in Smart Farming, Discrete Dynamics in Nature and Society, 2020.
[9] S. Selvaganesan, S. Jana, A.R. Begum, Design and Analysis of Pepper Leaf Disease Detection UsingDeep Belief Network, Eur. J. Mol. Clin. Med. 7 (9) (2020) 1724–1731.
[10] S. Khatoon, M.M. Hasan, A. Asif, M. Alshmari, Y.K. Yap, Image-Based Automatic Diagnostic System for Tomato Plants Using Deep Learning.
[11] J. Chen, J. Chen, D. Zhang, Y. Sun, Y.A. Nanehkaran, Using deep transfer learning for image-based plant disease identification, Comput. Electron. Agric. 173 (2020) 105393, https://doi.org/10.1016/j.compag.2020.105393.
[12] P. Bharali, C. Bhuyan, A. Boruah, in: May. Plant disease detection by leaf image classification using convolutional neural network, Springer, Singapore, 2019, pp. 194–205.
[13] S. Khatoon, M.M. Hasan, A. Asif, M. Alshmari, Y.K. Yap, Image-Based Automatic Diagnostic System for Tomato Plants Using Deep Learning, CMC 67 (1) (2021) 595–612, https://doi.org/10.32604/cmc.2021.014580.
[14] X. Xie, Y. Ma, B. Liu, J. He, S. Li, H. Wang, A deep-learning-based real-time detector for grape leaf diseases using improved convolutional neural networks, Front. Plant Sci. 11 (2020).
[15] Yang Zhang, Chenglong Song, Dongwen Zhang, Deep learning-based object detection improvement for tomato disease, IEEE Access 8 (2020) 56607–56614.
[16] Sivasubramaniam Janarthan, Selvarajah Thuseethan, Sutharshan Rajasegarar, Qiang Lyu, Yongqiang Zheng, John Yearwood, Deep metric learning based citrus disease classification with sparse data, IEEE Access 8 (2020) 162588–162600.
[17] Utpal Barman, Ridip Dev Choudhury, Diganto Sahu, Golap Gunjan Barman, Comparison of convolution neural networks for smartphone image based real time classification of citrus leaf disease, Comput. Electron. Agric. 177 (2020) 105661, https://doi.org/10.1016/j.compag.2020.105661.
[18] F. Kang, C. Wang, J. Li, Z. Zong, A multiobjective piglet image segmentation method based on an improved noninteractive GrabCut algorithm, Adv. Multimedia (2018).
[19] Essam H. Houssein, Bahaa El-din Helmy, Diego Oliva, Ahmed A. Elngar, Hassan Shaban, A novel Black Widow Optimization algorithm for multilevel thresholding image segmentation, Expert Syst. Appl. 167 (2021) 114159, https://doi.org/10.1016/j.eswa.2020.114159.
[20] A. G. Howard, M. Zhu, B. Chen et al., Mobilenets: efficient convolutional neural networks for mobile vision applications, 2017, https://arxiv.org/abs/1704.04861.
[21] Gaurav Dhiman, Diego Oliva, Amandeep Kaur, Krishna Kant Singh, S. Vimal, Ashutosh Sharma, Korhan Cengiz, BEPO: a novel binary emperor penguin optimizer for automatic feature selection, Knowl.-Based Syst. 211 (2021) 106560, https://doi.org/10.1016/j.knosys.2020.106560.
[22] D. Xiao, B. Li, Y. Mao, A multiple hidden layers extreme learning machine method and its application, Mathematical Problems in Engineering, 2017.
[23] D. Singh, N. Jain, P. Jain, P. Kayal, S. Kumawat, N. Batra, PlantDoc: a dataset for visual plant disease detection, in: *Proceedings of the 7th ACM IKDDCoDS and 25th COMAD*, 2020, pp. 249–253.
[24] A.K. Rangarajan, R. Purushothaman, A. Ramesh, Tomato crop disease classification using pre-trained deep learning algorithm, Procedia Comput. Sci. 133 (2018) 1040–1047.
[25] R. Karthik, M. Hariharan, S. Anand, P. Mathikshara, A. Johnson, R. Menaka, Attention embedded residual CNN for disease detection in tomato leaves, Appl. Soft Comput. 86 (2020) 105933, https://doi.org/10.1016/j.asoc.2019.105933.
[26] S. Rakesh, P. Sudhakar, Deep transfer learning with optimal kernel extreme learning machine model for plant disease diagnosis and classification, Int. J. Electr. Eng. Technol. (IJEET) 11 (9) (2020) 160–178.