International Journal of Electrical and Computer Engineering (IJECE)
Vol. 14, No. 3, June 2024, pp. 3168~3176
ISSN: 2088-8708, DOI: 10.11591/ijece.v14i3.pp3168-3176

An intelligent deep residual learning framework for tomato plant leaf disease classification

Gangadevi Ezhilarasan[1], Shoba Rani Ranganathan[2], Lawanya Shri Mani[3], Seifedien Kadry[4,5]
[1]Department of Computer Science, Dr. MGR Educational and Research Institute, Chennai, India
[2]Department of Computer Science and Engineering, Dr. MGR Educational and Research Institute, Chennai, India
[3]School of Information Technology and Engineering, Vellore Institute of Technology, Tamil Nadu, India
[4]Department of Applied Data Science, Noroff University College, Kristiansand, Norway
[5]MEU Research Unit, Middle East University, Amman, Jordan

Article Info

Article history:

Received Jun 17, 2023
Revised Oct 10, 2023
Accepted Jan 15, 2024

Keywords:

Agriculture
Classification
Deep learning
Jellyfish algorithm
Plant disease

ABSTRACT

Modern agriculture has been fascinated by various advancements in agriculture and the processing of foods and supply management that provision farmers to improve production. The health of plants is essential to improve production and economic growth. Diseases in plants can affect production and create a rigorous impact on the quality and create a hazard to food safety. Hence, detecting and classifying plant leaf diseases is essential to prevent the disease spread across the plants in the agriculture field and to improve productivity. The researchers in existing frameworks utilized artificial intelligence and machine learning techniques to demonstrate noteworthy solutions. However, a few issues exist related to noises in the images, hyperparameter selection problems, and over-fitting problems that influence prediction accuracy. The proposed model jellyfish ResNet (JF-ResNet) works well to achieve a better accuracy level by incorporating jellyfish optimized ResNET for tomato plant leaf disease identification and classification. The performance metrics such as accuracy, specificity, sensitivity, and F1-Score is used to evaluate the performance of the JF-ResNet model. The proposed model achieves 97.3% accuracy, 95.3% sensitivity, 96.1% specificity, 96.9% recall, 96.4% precision and 97.1% F1-Score.

Corresponding Author:

Gangadevi Ezhilarasan
Department of Computer Science, Dr. MGR Educational and Research Institute
Maduravoyal, Chennai, India
Email: smgangadevi@gmail.com

1. INTRODUCTION

Plant leaf disease is the primary cause that creates an impact on quality and production. The demand for productivity is rising due to the current population. High quality and quantity are the major concerns of the farmers to increase economic growth [1]. Plant protection is very important for the farmers which creates a high impact on production. Identification of plant leaf disease is essential to protect and prevent diseases spread across numerous plants [2]. Early detection of plant leaf diseases allows proper treatment and prevention to further spread of the disease across healthy plants [3]. Farmers can take necessary steps to manage by implementing appropriate treatments to cure the disease-affected plants. More than 60% yield failure is reported due to bacterial, virus, and fungal infections during harvesting [4]. Technology and its advancement facilitate researchers to utilize appropriate artificial intelligence (AI) models to detect and classify plant leaf diseases [5]. Machine learning and deep learning algorithms are becoming key tools to diagnose the quality of plants or to identify and classify plant leaf diseases. Early detection of disease-affected plants is necessary to reduce the maintenance cost [6]. Hence plant disease detection and classification model is important which produces better accuracy results [7]. In recent years, artificial intelligence-based classification frameworks were applied by researchers to obtain plant pictures and apply image processing and machine learning algorithms to classify the type of disease the plant gets affected [8]. The AI-based plant leaf disease detection and classification gives prominent results, hence the improvement of accuracy to classify the type of disease is important [9], [10]. Various machine learning models are used for classifying the tomato plant leaf disease; however, few algorithms fail to obtain high-accuracy results due to the size of the datasets and due to hyperparameter problems [11]. To improve the accuracy level and avoid hyperparameter issues, many optimization approaches are used to fine-tune the hyperparameters to get better classification results [12]. Metaheuristic optimization algorithms are mainly used to improve the classification results to improve the convergence rate and to obtain optimal results [13]. The proposed work uses the deep neural network ResNet for the detection and classification of tomato plant leaf disease. The jellyfish optimization algorithm is used to optimize the ResNet to improve classification results and to fine-tune the parameters applied to the algorithm. The preprocessing techniques are used to improve the quality of image segmentation, labeling, and noise removal. The preprocessed images are fed as input to ResNet model for the detection of plant leaf diseases and classify based on the disease type. The main contribution of the proposed work is as follows:

a. Detection and classification of tomato plant leaf disease using JF-ResNet model.
b. Plant village datasets are used to train and test the model for the detection and classification of tomato plant leaf disease.
c. Preprocessing the obtained image is used to improve the quality and enhance the classification result by feature extraction, segmentation and noise removal mechanism.
d. Jellyfish optimization algorithm is used to solve hyperparameter problem and to improve the convergence rate in order to get better accuracy results.
e. The performance metrics like accuracy, sensitivity, specificity and F1-Score to evaluate the JF-ResNet model.

2. LITERATURE REVIEW

Rao *et al.* [14] produced a deep learning method to identify grape leaf and mango leaf with the support of transfer learning. They have used a large dataset of leaf images which are trained to detect the presence of diseases by a deep convolutional model neural network. To acquire the features and classify the type of leaf diseases by pre-trained AlexNet and achieved an accuracy of 99% for grape leaf and 89% for mango leaf. Liu *et al.* [15] followed a deep convolutional kind of neural network for identifying apple leaf diseases. The AlexNet architecture is used in this study to recognize four different categories of apple leaf diseases with 40 epochs and a 0.01 learning rate with the optimization algorithm. This model achieved 97.62% accuracy while testing the images from the datasets.

Arsenovic *et al.* [16] introduced techniques to overcome the drawbacks of deep learning-based techniques for identifying plant diseases. Many augmentation techniques were applied and to rise the size of the datasets generative adversarial networks (GAN) architecture was applied. The proposed model achieved 93.67% accuracy with self-captured images. Khan *et al.* [17] applied a deep learning methodology for analyzing as well as detecting apple leaf diseases. Different layers with corresponding output shapes were applied with suitable parameters for fine-tuning the model. You look only once v4 (YOLO v4), faster region-based convolutional neural network (RCNN), and EfficientDet methods were applied with different hyperparameters and achieved 87.9% accuracy with a precision of 88%, recall of 82%, and F1-Score of 85%. Performance comparison was also made with different classification models.

Chakraborty *et al.* [18] proposed a deep learning structure that followed VGG16, VGG19, MobileNet, and ResNet 50 and fine-tuned VGG16 architectures used in the plant village dataset and attained an accuracy of 97.89%. Training loss with validation loss is also calculated for all the architectures used. Doh *et al.* [19] proposed a deep learning framework using DenseNet for the detection of four varieties of citrus diseases. The validation is carried out using a smaller-sized dataset and divided the leaves of the citrus plant into 4 types with disease variety and one type with healthy leaves. The approach uses transfer learning for validation and showed an accuracy result of 92%, an F1-Score of 95%, and a 98.8% of area under the ROC curve (AUC) score.

For reason that the drawbacks of existing optimization methods, the meta-heuristic optimization approach for attempting combinatorial optimization problems was developed. Meta-heuristic algorithms are focused on the best optimization algorithms, where they take many advantages like resilience, performance reliability, simplicity, ease of implementation, and many more. Meta-heuristic algorithms are separated into several groups, including evolutionary-based algorithms based on evolutionary theory and Swarm-based algorithms based on social behavior and many decision-making of a variety of social groups. The aim for attaining a specific objective in these algorithms is naturally built on bio-community intelligence and cooperative action.

3. METHOD

3.1. Convolutional neural networks

Convolutional neural networks (CNNs) are among the most significant deep learning techniques and are frequently used in a variety of computer vision applications. A CNN typically has three primary layers: the convolution layer, the pooling layer, and the fully connected layer, each of which has a specific function. These three layers work together to create structures that are based on CNNs. The backpropagation method and the loss function with the proper response are used to train the network output [20]. This kind of network is trained using backpropagation, which is then utilized to modify the network's parameters. Below is a description of each layer:

a. Convolution layer: In this layer, the CNN convolves the input picture and the middle feature maps using various kernels to produce various feature maps.
b. Pooling layer: After a convolutional layer, the pooling layer is typically added. It can minimize the size of feature maps and network parameters. The Max-pooling function and the Average-pooling function are the most widely used implementations.
c. Fully connected layer: The fully connected layer enables us to display the network's output as a vector with a predetermined size. Typically, this vector is employed for classification. The fact that this kind of layer has so many parameters is a major issue.

As a result, training comes at a very high processing cost. In this work, three fully linked output nodes are considered for classification, which uses the SoftMax activation function to generate a probability for the output class [21]. The following defines the SoftMax activation function for class I.

$$SM(x)_i = \frac{\exp(x_i)}{\sum_{j=1}^{n} \exp(x_j)} \tag{1}$$

Here, x is an input vector of a soft max function SM which contains n elements. x_i is the i-th element of input vector. exp (x_i) is the exponential function for x_i . $\sum_{j=1}^{n} exp(x_j)$ is a normalization term that verifies the values of output vector $SM(x)_i$. n- is the number of possible outcomes. A loss function that can determine the classification error is necessary for training the model. The categorical cross-entropy loss function is one of the well-known loss functions in classification [22]. A value between 0 and 1 is used to determine how well a classification model is doing.

$$Loss = -\frac{1}{N}\left[\sum_{k=1}^{N}[t_k \log(p_k) + (1 - t_k)\log(1 - p_k)]\right] \tag{2}$$

Here, for total N datapoints, t_k is the truth value considering value 0 or 1 and p_k is the SoftMax probability for the i^{th} data point.

3.2. ResNet

The key innovation of the ResNet architecture is the introduction of "residual connections", which allow for the training of much deeper neural networks than was previously possible. Residual connections involve adding skip connections that bypass one or more layers of the network, allowing information to flow directly from one layer to another [23]. This approach addresses the issue of vanishing gradients that can occur in very deep neural networks, where gradients can become too small to propagate through the layers during training, making it difficult to learn the desired features. With residual connections, the gradients can flow through the network more easily, allowing for deeper and more accurate models to be trained. Figure 1 represents the architecture of the ResNet approach. Without affecting the performance of the model, the ResNet architecture helps to skip the layers of the model.

ResNet architectures typically have many layers, with some of the most commonly used variants having over 100 layers. The original ResNet paper introduced several variants of the architecture, including ResNet-18, ResNet-34, ResNet-50, ResNet-101, and ResNet-152, with each variant having a different number of layers. Residual network is an innovative method which had changed the training of deep convolutional networks for computer related tasks. ResNet uses batch normalization to adjust the input layer to make the network to perform better. It identifies the network to protect from gradient problem.

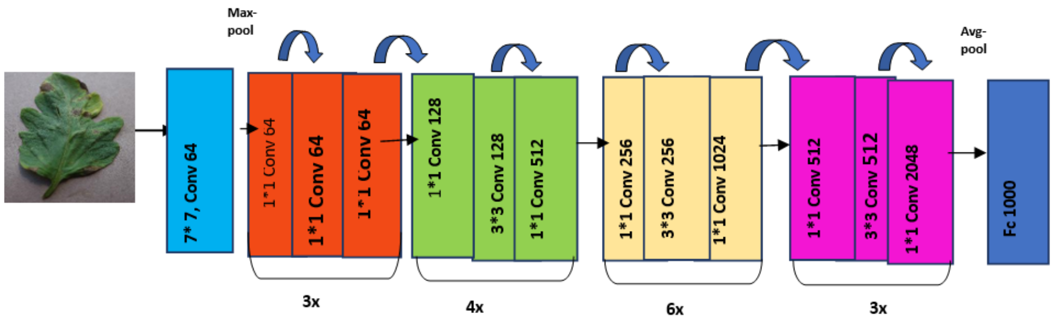

Figure 1. ResNet architecture

3.3. JellyFish algorithm

Chou and Truong in 2020 [24] had recently developed an algorithm known as the jellyfish search optimizer. This strategy was developed in considering the behavior of artificial jellyfish in ocean currents. The jellyfish adjusts to all temperature ranges and the diverse shapes, sizes, and colors found in this planet. Their approaches to prey are also different. Their bottom portion, which resembles an umbrella, aids in this movement by forcing the water outward in order to advance. The jellyfish bloom is another name for the jellyfish mass [25]. When favorable conditions, such as the availability of air, food, and temperature, are present, a swarm will be formed.

The population of jellyfish is given as (3).

$$JF = \{JF_1, JF_{2,..}JF_n\} \tag{3}$$

Here, *n*-is the population size of the jellyfish. Each of the jellyfish is considered as DV-dimensional vector as (4),

$$JF_k = (jf_{k,1}, jf_{k,2}, \dots jf_{k,dv}) \tag{4}$$

Each element $jf_{k,dv}$ is the dv-th dimension of kth jellyfish where DV=1, 2, … dv. To avoid slow convergence, the population of jellyfish is not initialized with random values. The logistic map, a chaotic map is used for initializing the population [26]. It is shown as (5).

$$JF_{k+1} = \omega JF_k\,(1 - JF_k), 0 < JF_0 < 1 \tag{5}$$

where ω is 4.0, JF_k refers the location of the kth jellyfish based on logistic chaotic value and JF_0 is the jellyfish where $JF_0 \,\epsilon\, [0, 1]$ for the initial population. The nutrients present in the ocean current is attracted by the jellyfish. The direction of the ocean current is given by (6), (7).

$$\vec{O}_{dir} = JF^* - A_c \frac{\sum JF_k}{n} \tag{6}$$

$$\vec{O}_{dir} = JF^* - rand(0,\ 1) * \gamma * \mu \tag{7}$$

Jellyfish with current best is represented as JF^*. A_c is for attraction governing feature. Location of jellyfish population is denoted as μ. γ is the distribution co-efficient which is greater than 0 and equal to 3. The new location of jellyfish at time t+1 is given by (8), (9).

$$JF_k^{t+1} = JF_k^t + \text{rand }(0, 1) * \vec{O}_{dir} \tag{8}$$

$$JF_k^{t+1} = JF_k^t + \text{rand }(0, 1) * (JF^* - rand(0, 1) * \gamma * \mu) \tag{9}$$

Active motion or passive motion are performed by jellyfish. They will always start with passive motion which is based on its own location, and they update them with active motion [27]. The updated location is represented as (10).

$$JF_k^{t+1} = JF_k^t + \alpha * \text{rand }(0, 1) * UP_{BSS} - LO_{BSS} \tag{10}$$

Here, UP_{BSS} denoted upper bound search space and LO_{BSS} denotes lower bound search space [12]. α refers to the motion co-efficient which is greater than 0 and found as 0.1.

4. NOVEL JF-RESNET FOR HYPERPARAMETER OPTIMIZATION

The proposed method used the Plant Village dataset to measure the performance and to classify the tomato plant leaf diseases. The total number of leaf images in this dataset are 54,303. Totally, 3,521 images are used for training and testing. MATLAB 2016A is used to implement the JFA-ResNet algorithm.

The JF-ResNet algorithm is applied to optimize the weights and bias of the neural network [28]. The dataset from plant village is taken and splitted as training, testing and validation. After pre-processing the leaf images, ResNet model is initialized with blocks and filters to train them using jellyfish algorithm with loss function as depicted in Algorithm 1. The hyperparameters are initialized to calculate the fitness values to find the current best food and its new location [29]. Overfitting is also prevented by monitoring the performance of the algorithm and fine tune the model to deploy the proposed method to classify the tomato plant leaf diseases [30].

Algorithm 1. JFA-ResNet
1. Collect the dataset from Plant Village
2. Split the dataset as train, test and validate
3. Preprocess the images and normalize the pixel values
4. Initialize ResNet model with number of blocks and number of filters
5. Train ResNet model using JFA and loss function

$$Loss = -\frac{1}{N}\left[\sum_{k=1}^{N}[t_k \log(p_k) + (1-t_k)\log(1-p_k)]\right]$$

5.1 Max iteration, motion co-efficient $\alpha = 0.1$, distribution co-efficient $\gamma = 3$
5.2 Initialize the population of JF and calculate fitness value.
5.3 Objective function (food value) is calculated as JF_k
5.4 Current best food of JF is JF^*, time t=1
5.5 Ocean current direction is calculated using

$$\vec{O}_{dir} = JF^* - A_c\frac{\sum JF_k}{n}$$

5.6 New location of JF is determined by

$$JF_k^{t+1} = JF_k^t + \text{rand}(0,1) * (JF^* - rand(0,1) * \gamma * \mu)$$

5.7 Boundary conditions are verified and food quantity at new location is estimated
6. Prevent overfitting by monitoring validation set performance
7. Evaluate the performance of trained model
8. Fine tune the ResNet model
9. Deploy the ResNet model to classify the tomato plant leaf disease by feeding through network.

The neural network is filled with training data and jellyfish population is also initialized. The hyperparameters like motion co-efficient and distribution co-efficient were initialized [31]. The fitness function of each jellyfish is measured by the neural network, and they are the vector which is treated as a set of weights and biases [32]. So, each of the jellyfish present here helps us to optimize the neural network. With the fixed number of iterations, fitness value is calculated. The framework of JFA-ResNet is explained in Figure 2.

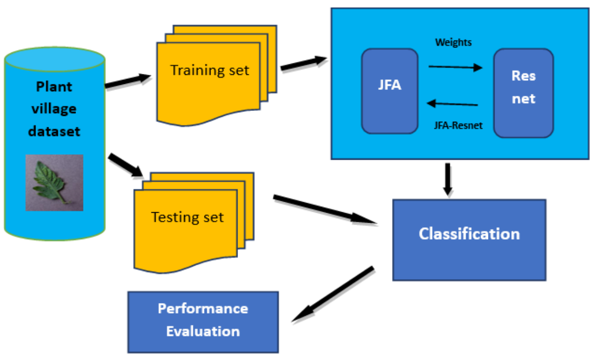

Figure 2. Framework of JFA-ResNet classifier

The search location of the jellyfish is based on the availability of food and its movement is measured based on time control unit that is in ocean current or into the swarm of jellyfish [33]. After which, the jellyfish location is updated through iterations. The best jellyfish measured from the entire iteration is considered as the optimal solution. To test the data, jellyfish is deployed with the neural network and initializes the population to avoid slow convergence and getting trapped into local minima.

5. EXPERIMENTS AND RESULT ANALYSIS

The effectiveness of JF-ResNet was assessed to detect and classify the tomato leaf disease and its types. The model is trained, tested and validated using the plant village dataset that includes the images of healthy and disease infected tomato leaves. The experiments are carried out in Windows 10 OS with Intel Xeon E5 CPU, Radeon Pro W6400, with memory 64 GB Ram, CUDA Toolkit, CUDNN v7.0, 8 GB of video memory, and MATLAB 2016A. TensorFlow is used for implementing the deep learning framework. The plant village dataset is used for classifying of tomato plant leaf disease and its types. The total numbers of 54,303 samples are used for training the proposed deep learning model. The proposed model is evaluated with the following performance metrics. Accuracy, sensitivity, specificity, recall, precision and F1-Score are the metrics used to assess the JF-ResNet model.

$$\text{Accuracy}=\frac{(TrueN+TrueP)}{(TrueN+TrueP+FalseN+FalseP)} \tag{11}$$

$$\text{Sensitivity} = \frac{TrueP}{TrueP+FalseN} \tag{12}$$

$$\text{Specificity} = \frac{TrueN}{TrueN+FalseP} \tag{13}$$

$$\text{Recall}= \frac{TrueP}{(FalseN+TrueP)} \tag{14}$$

$$\text{Precision} = \frac{TrueP}{TrueP+FalseP} \tag{15}$$

$$\text{F1-Score} = \frac{2*precision*recall}{precision+recall} \tag{16}$$

Here, *TrueP*, *TrueN*, *FalseP*, and *FalseN* are true positive, true negative, false positive, and false negative respectively.

6. RESULTS AND DISCUSSION OF JF-RESNET MODEL

The performance metrics such as accuracy, recall, specificity, sensitivity, and F1-Score is used to train and test the JF-ResNet model. Table 1 depicts the results for various performance metrics mentioned for the proposed work. The results proves that the proposed JF-ResNet model performs well in 90% of training and 10% of testing. The JF-ResNet model was trained using 8 batch sizes with 100 epochs. JF-ResNet model achieves 97.3% accuracy, 95.3% sensitivity, 96.1% specificity, 96.9% recall, 96.4% precision and 97.1% F1-Score compared to the existing state of art models.

Table 1. Performance result of JF-ResNet

	Accuracy%	Sensitivity %	Specificity%	Recall %	Precision %	F1-Score %
Proposed approach	97.3	95.3	96.1	96.9	96.4	97.1

Figure 3 depicts the overall performance of the proposed work based on the performance metrics. The work outperforms well with 97.3% of accuracy, 95.3% of sensitivity, 96.1% of specificity, 96.9% of recall, 96.4% of precision and 97.1% of F1-Score. A high sensitivity value represents that the model has a few false negative results [34]. Specificity helps to identify how correctly the model has identified the disease. Table 2 illustrates the comparison result for accuracy performance metrics and in Figure 4 depicts the graphical representation of the accuracy metric.

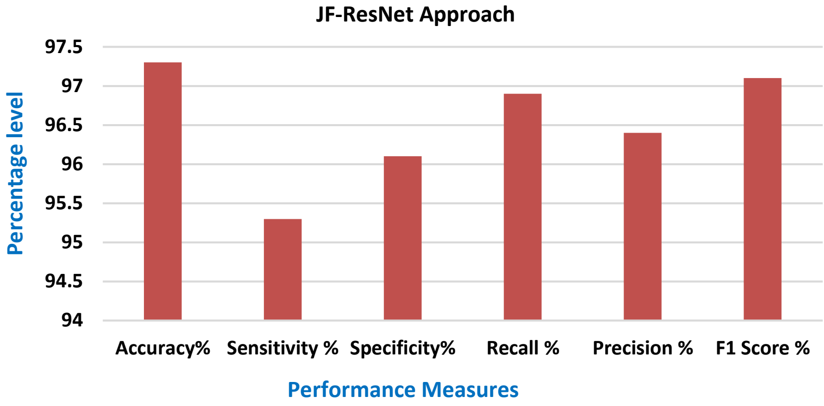

Figure 3. Performance result of JF-ResNet

Table 2. Comparison results for accuracy

Technique	Testing accuracy
Support vector machine (SVM) [35]	87%
Based radial basis function neural network (BRBFNN) [36]	91%
Proposed work	97.3%

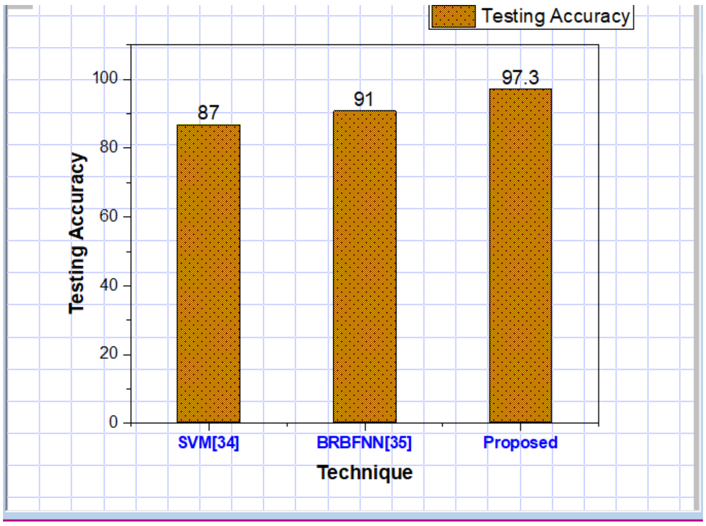

Figure 4. Accuracy result of JF-ResNet

7. CONCLUSION

The proposed model is used to detect and classify the tomato plant leaf disease and its type using jellyfish optimized ResNet. The JF-ResNet model uses plant village dataset to perform training and testing the model. Jellyfish optimized ResNet is used to tune the parameters to achieve better performance results. The proposed work performs well with 97.3% of accuracy, 95.3% of sensitivity, 96.1% of specificity, 96.9% of recall, 96.4% of precision and 97.1% of F1-Score. The comparison of accuracy is performed with the existing well-known algorithms like SVM and BRBFNN. Based on the comparison results the JF-ResNet outperforms better compared to SVM and BRBFNN approaches. In future we can use hybrid optimized ResNet to show better results and to reduce the computational time compared to other existing approaches.

REFERENCES

[1] S. S. Harakannanavar, J. M. Rudagi, V. I. Puranikmath, A. Siddiqua, and R. Pramodhini, “Plant leaf disease detection using computer vision and machine learning algorithms,” *Global Transitions Proceedings*, vol. 3, no. 1, pp. 305–310, Jun. 2022, doi: 10.1016/j.gltp.2022.03.016.

[2] A. Ahmad, D. Saraswat, and A. El Gamal, "A survey on using deep learning techniques for plant disease diagnosis and recommendations for development of appropriate tools," *Smart Agricultural Technology*, vol. 3, Feb. 2023, doi: 10.1016/j.atech.2022.100083.
[3] V. Singh and A. K. Misra, "Detection of plant leaf diseases using image segmentation and soft computing techniques," *Information Processing in Agriculture*, vol. 4, no. 1, pp. 41–49, Mar. 2017, doi: 10.1016/j.inpa.2016.10.005.
[4] K. P. Ferentinos, "Deep learning models for plant disease detection and diagnosis," *Computers and Electronics in Agriculture*, vol. 145, pp. 311–318, Feb. 2018, doi: 10.1016/j.compag.2018.01.009.
[5] V. G. Krishnan, J. Deepa, P. V. Rao, V. Divya, S. Kaviarasan, "An automated segmentation and classification model for banana leaf disease detection," *Journal of Applied Biology & Biotechnology*, Oct. 2022, doi: 10.7324/JABB.2021.100126.
[6] R. U. Khan, K. Khan, W. Albattah, and A. M. Qamar, "Image-based detection of plant diseases: from classical machine learning to deep learning journey," *Wireless Communications and Mobile Computing*, vol. 2021, pp. 1–13, Jun. 2021, doi: 10.1155/2021/5541859.
[7] S. Ishak, M. H. Fazalul Rahiman, S. N. A. Mohd Kanafiah, and H. Saad, "Leaf disease classification using artificial neural network," *Jurnal Teknologi*, vol. 77, no. 17, Nov. 2015, doi: 10.11113/jt.v77.6463.
[8] A. K. Rangarajan, R. Purushothaman, and A. Ramesh, "Tomato crop disease classification using pre-trained deep learning algorithm," *Procedia Computer Science*, vol. 133, pp. 1040–1047, 2018, doi: 10.1016/j.procs.2018.07.070.
[9] Y. Zhang, C. Song, and D. Zhang, "Deep learning-based object detection improvement for tomato disease," *IEEE Access*, vol. 8, pp. 56607–56614, 2020, doi: 10.1109/ACCESS.2020.2982456.
[10] L. Li, S. Zhang, and B. Wang, "Plant disease detection and classification by deep learning-a review," *IEEE Access*, vol. 9, pp. 56683–56698, 2021, doi: 10.1109/ACCESS.2021.3069646.
[11] H. H. Alshammari, A. I. Taloba, and O. R. Shahin, "Identification of olive leaf disease through optimized deep learning approach," *Alexandria Engineering Journal*, vol. 72, pp. 213–224, Jun. 2023, doi: 10.1016/j.aej.2023.03.081.
[12] S. Mirjalili, *Evolutionary algorithms and neural networks*, vol. 780. Germany: Springer: Berlin/Heidelberg, 2019.
[13] S. Sun, Z. Cao, H. Zhu, and J. Zhao, "A survey of optimization methods from a machine learning perspective," *IEEE Transactions on Cybernetics*, vol. 50, no. 8, pp. 3668–3681, Aug. 2020, doi: 10.1109/TCYB.2019.2950779.
[14] U. Sanath Rao *et al.*, "Deep learning precision farming: grapes and mango leaf disease detection by transfer learning," *Global Transitions Proceedings*, vol. 2, no. 2, pp. 535–544, 2021, doi: 10.1016/j.gltp.2021.08.002.
[15] B. Liu, Y. Zhang, D. J. He, and Y. Li, "Identification of apple leaf diseases based on deep convolutional neural networks," *Symmetry*, vol. 10, no. 1, 2018, doi: 10.3390/sym10010011.
[16] M. Arsenovic, M. Karanovic, S. Sladojevic, A. Anderla, and D. Stefanovic, "Solving current limitations of deep learning based approaches for plant disease detection," *Symmetry*, vol. 11, no. 7, 2019, doi: 10.3390/sym11070939.
[17] A. I. Khan, S. M. K. Quadri, S. Banday, and J. Latief Shah, "Deep diagnosis: A real-time apple leaf disease detection system based on deep learning," *Computers and Electronics in Agriculture*, vol. 198, no. May, Art. no. 107093, 2022, doi: 10.1016/j.compag.2022.107093.
[18] K. K. Chakraborty, R. Mukherjee, C. Chakroborty, and K. Bora, "Automated recognition of optical image based potato leaf blight diseases using deep learning," *Physiological and Molecular Plant Pathology*, vol. 117, no. October 2021, Art. no. 101781, 2022, doi: 10.1016/j.pmpp.2021.101781.
[19] B. Doh, D. Zhang, Y. Shen, F. Hussain, R. F. Doh, and K. Ayepah, "Automatic citrus fruit disease detection by phenotyping using machine learning," in *2019 25th International Conference on Automation and Computing (ICAC)*, Sep. 2019, pp. 1–5, doi: 10.23919/IConAC.2019.8895102.
[20] I. Ahmad, M. Hamid, S. Yousaf, S. T. Shah, and M. O. Ahmad, "Optimizing pretrained convolutional neural networks for tomato leaf disease detection," *Complexity*, vol. 2020, pp. 1–6, Sep. 2020, doi: 10.1155/2020/8812019.
[21] A. Afifi, A. Alhumam, and A. Abdelwahab, "Convolutional neural network for automatic identification of plant diseases with limited data," *Plants*, vol. 10, no. 1, p. 28, Dec. 2020, doi: 10.3390/plants10010028.
[22] A. Guerrero-Ibañez and A. Reyes-Muñoz, "Monitoring tomato leaf disease through convolutional neural networks," *Electronics*, vol. 12, no. 1, Jan. 2023, doi: 10.3390/electronics12010229.
[23] R. Jing, L. Xue, M. Li, L. Yu, and J. Luo, "layerUMAP: a tool for visualizing and understanding deep learning models in biological sequence classification using UMAP," *iScience*, vol. 25, no. 12, Dec. 2022, doi: 10.1016/j.isci.2022.105530.
[24] J.-S. Chou and D.-N. Truong, "A novel metaheuristic optimizer inspired by behavior of jellyfish in ocean," *Applied Mathematics and Computation*, vol. 389, Jan. 2021, doi: 10.1016/j.amc.2020.125535.
[25] D. Devarajan *et al.*, "Cervical cancer diagnosis using intelligent living behavior of artificial jellyfish optimized with artificial neural network," *IEEE Access*, vol. 10, pp. 126957–126968, 2022, doi: 10.1109/ACCESS.2022.3221451.
[26] A. Alam *et al.*, "Jellyfish search optimization algorithm for MPP tracking of PV system," *Sustainability*, vol. 13, no. 21, Oct. 2021, doi: 10.3390/su132111736.
[27] M. Farhat, S. Kamel, A. M. Atallah, and B. Khan, "Optimal power flow solution based on jellyfish search optimization considering uncertainty of renewable energy sources," *IEEE Access*, vol. 9, pp. 100911–100933, 2021, doi: 10.1109/ACCESS.2021.3097006.
[28] A. M. Abdu, M. M. M. Mokji, and U. U. U. Sheikh, "Machine learning for plant disease detection: an investigative comparison between support vector machine and deep learning," *IAES International Journal of Artificial Intelligence (IJ-AI)*, vol. 9, no. 4, pp. 670–683, Dec. 2020, doi: 10.11591/ijai.v9.i4.pp670-683.
[29] J.-S. Chou, S. Tjandrakusuma, and C.-Y. Liu, "Jellyfish search-optimized deep learning for compressive strength prediction in images of ready-mixed concrete," *Computational Intelligence and Neuroscience*, vol. 2022, pp. 1–26, Aug. 2022, doi: 10.1155/2022/9541115.
[30] J.-S. Chou and A. Molla, "Recent advances in use of bio-inspired jellyfish search algorithm for solving optimization problems," *Scientific Reports*, vol. 12, no. 1, Nov. 2022, doi: 10.1038/s41598-022-23121-z.
[31] M. Martin-Abadal, A. Ruiz-Frau, H. Hinz, and Y. Gonzalez-Cid, "Jellytoring: real-time jellyfish monitoring based on deep learning object detection," *Sensors*, vol. 20, no. 6, Mar. 2020, doi: 10.3390/s20061708.
[32] Z. Zhou, X. Yang, H. Ji, and Z. Zhu, "Improving the classification accuracy of fishes and invertebrates using residual convolutional neural networks," *ICES Journal of Marine Science*, vol. 80, no. 5, pp. 1256–1266, Jun. 2023, doi: 10.1093/icesjms/fsad041.
[33] D. Yadav, N. Singh, V. S. Bhadoria, N. C. Giri, and M. Cherukuri, "A novel metaheuristic jellyfish optimization algorithm for parameter extraction of solar module," *International Transactions on Electrical Energy Systems*, vol. 2023, pp. 1–21, Jul. 2023, doi: 10.1155/2023/5589859.
[34] K. Sujigarasharma, M. L. Shri, E. Gangadevi, R. K. Dhanaraj, N. C, and B. Balusamy, "Detection and classification of speech disorder using FOA-SCNet," in *2023 3rd International Conference on Computing and Information Technology (ICCIT)*, Sep. 2023, pp. 391–395, doi: 10.1109/ICCIT58132.2023.10273910.

[35] D. Das, M. Singh, S. S. Mohanty, and S. Chakravarty, "Leaf disease detection using support vector machine," in *2020 International Conference on Communication and Signal Processing (ICCSP)*, Jul. 2020, pp. 1036–1040, doi: 10.1109/ICCSP48568.2020.9182128.

[36] S. S. Chouhan, A. Kaul, U. P. Singh, and S. Jain, "Bacterial foraging optimization based radial basis function neural network (BRBFNN) for identification and classification of plant leaf diseases: an automatic approach towards plant pathology," *IEEE Access*, vol. 6, pp. 8852–8863, 2018, doi: 10.1109/ACCESS.2018.2800685.

BIOGRAPHIES OF AUTHORS

Gangadevi Ezhilarasan is an assistant professor in the Department of Computer Science, at Loyola College, Chennai, India. She received her M.Sc, and M.phil from Vellore Institute of Technology, Vellore, India in 2005. She is pursuing Ph.D. at MGR Educational and Research Institute, Chennai, India. She has published five patents, 3 Authored and edited books, and more than 22+ research papers in international journals and many book chapters. She has published a book "DBMS", 3 online Books, and written a Higher secondary computer science book for the State Government of Tamil Nadu. Her areas of research are machine learning, deep learning, IoT, and cloud computing. She can be contacted at email: smgangadevi@gmail.com.

Shoba Rani Ranganathan received her postgraduate degree in computer applications from Bharathidasan University, Tamil Nadu, India in 2002 and her M.Phil from Annamalai University, Tamil Nadu, India in 2004, and her Ph.D. degree from Mother Teresa Women's University, Tamil Nadu, India in Feb 2014. She is the author/co-author of over 36 referred research papers. She is currently working as a Professor and Research guide in the Department of Computer Science and Engineering, at Dr. M.G.R Educational and Research Institute, Chennai, Tamil Nadu, India. She has published research papers in various UGC, Scopus indexed, SCI journals, and patents published at the international level. She has reviewed numerous research papers at National Conferences and International Conferences. Her publications and research areas include image processing, machine learning, and data mining. She can be contacted at email: shobarani.cse@drmgrdu.ac.in.

Lawanya Shri Mani is an associate professor in the School of Information Technology and Engineering at VIT, Vellore, India. She received the M.Tech CSE degree in computer science and engineering at VIT, Vellore, India in 2010 and a Ph.D. degree in the School of Information Technology and Engineering from VIT, Vellore, India in 2017. She has contributed 2 Authored and Edited books, 2 Patents, and 50+ articles and papers in refereed journals and international conferences and contributed chapters to the books. Her research interest includes BlockChain technology, machine learning, cloud computing, and IoT. She serves as an editorial board member in Informatics in Medicine Unlocked (Elsevier) and a reviewer for reputed journals. She is a member of the Computer Science Teacher Association (CSTA); and the International Association of Engineers (IAENG). She can be contacted at email: lawanyaraj@gmail.com.

Seifedien Kadry has a bachelor's degree in 1999 from Lebanese University, MS degree in 2002 from Reims University (France) and EPFL (Lausanne), PhD in 2007 from Blaise Pascal University (France), HDR degree in 2017 from Rouen University (France). At present his research focuses on Data Science, education using technology, system prognostics, stochastic systems, and applied mathematics. He is an ABET program evaluator for computing, and ABET program evaluator for Engineering Tech. He can be contacted at email: skadry@gmail.com.