



OPEN

## Leveraging three-tier deep learning model for environmental cleaner plants production

Zahraa Tarek<sup>1</sup>, Mohamed Elhoseny<sup>1,2</sup>, Mohamemd I. Alghamdi<sup>3</sup> & Ibrahim M. EL-Hasnony<sup>1</sup>✉

The world's population is expected to exceed 9 billion people by 2050, necessitating a 70% increase in agricultural output and food production to meet the demand. Due to resource shortages, climate change, the COVID-19 pandemic, and highly harsh socioeconomic predictions, such a demand is challenging to complete without using computation and forecasting methods. Machine learning has grown with big data and high-performance computers technologies to open up new data-intensive scientific opportunities in the multidisciplinary agri-technology area. Throughout the plant's developmental period, diseases and pests are natural disasters, from seed production to seedling growth. This paper introduces an early diagnosis framework for plant diseases based on fog computing and edge environment by IoT sensors measurements and communication technologies. The effectiveness of employing pre-trained CNN architectures as feature extractors in identifying plant illnesses has been studied. As feature extractors, standard pre-trained CNN models, AlexNet are employed. The obtained in-depth features are eliminated by proposing a revised version of the grey wolf optimization (GWO) algorithm that approved its efficiency through experiments. The features subset selected were used to train the SVM classifier. Ten datasets for different plants are utilized to assess the proposed model. According to the findings, the proposed model achieved better outcomes for all used datasets. As an average for all datasets, the accuracy of the proposed model is 93.84 compared to 85.49, 87.89, 87.04 for AlexNet, GoogleNet, and the SVM, respectively.

### Abbreviations

AI	Artificial intelligence
ML	Machine learning
DL	Deep learning
GWO	Grey wolf optimization
IoT	Internet of things
QoS	Quality of service
MGWO	Modified grey wolf optimization
DTL	Data transfer learning
SSD	Single shot detector
RCNN	Recurrent convolutional neural network
MLP	Multi-layer perceptron
FCN	Fully convolutional neural network
KNN	K-nearest neighbor
VGG	Visual geometry group
RF	Random forest
SVM	Support vector machines
CNN	Convolutional neural network
DCNN	Deep convolutional neural network
DAML	Deep adversarial metric learning
CAE	Convolutional autoencoder
GLDD	Grape leaf disease dataset
GPDCNN	Global pooling extended convolutional neural network
EPA	Environmental Protection Agency

<sup>1</sup>Faculty of Computers and Information Science, Mansoura University, Mansoura, Egypt. <sup>2</sup>College of Computing and Informatics, University of Sharjah, Sharjah, United Arab Emirates. <sup>3</sup>Department of Computer Science, Al-Baha University, Al Bahah, Kingdom of Saudi Arabia. ✉email: ibrahimhesin2005@mans.edu.eg

TN	True negatives
FN	False negatives
FP	False positives
TP	True positives
ACO	Ant colony optimization
PSO	Particle swarm optimization
BOA	Butterfly optimization algorithm

In the global economy, agriculture is crucial. The agriculture system will face more strain as the human population grows and the COVID-19 pandemic takes hold. Agri-technology has developed as a novel scientific area that employs data-intensive techniques to boost agricultural output while reducing the environmental impact. In contemporary agricultural activities, data is created by a variety of sensors that give a better understanding of the operational surroundings (weather conditions, dynamic crop, and soil) as well as the operation itself (data from machines), resulting in more precise and quicker decision-making<sup>1</sup>. Conservation agriculture has long been regarded as an effective and ecologically beneficial management strategy to boost agricultural yields. In addition, measuring the total impact of conservation agriculture on crop output amelioration by taking the average of the entire dataset is not unfamiliar. Nevertheless, the influence of conservation agriculture on yielding cleaner output should be examined<sup>2</sup>.

Plant disease and pest identification are critical research areas in the realm of machine learning. To determine whether or not a plant image contains diseases or pests, machine vision equipment is used<sup>3</sup>. Are these types of detection systems needed? A vital food security hazard is plant disease. Agricultural and population growth are affected by plant diseases, as is the economy. Disease control, food safety, and anticipated loss of income need an automated and exact estimation of plant disease gravity. Plant diseases must thus be identified and treated at an early stage. Non-expert farmers, on the other hand, are frequently oblivious to non-native illnesses, necessitating consultation with experts to determine whether there are any strange symptoms or appearances on their crops. A farmer may have to travel vast distances to consult with an expert, which is costly and time-consuming. To automate the process of identifying plant illnesses, these difficulties motivate research and expansion in this area. The essential requirement for a plant disease diagnosis model that can operate in an Internet of Things (IoT) environment with minimal processing capabilities is very important<sup>4</sup>.

Already, machine vision is being used in agriculture to detect plant diseases and pests. While artificial intelligence (AI) is still a long way from being widely deployed, the technology has tremendous development potential and application value<sup>5</sup>. Plant diseases have been classified and identified using machine learning (ML) models. Nonetheless, with improvements in deep learning (DL), this field of research looks to offer enormous promise for greater accuracy. When it comes to planting disease detection, multiple DL structures have been developed or modified, as have many visualization methodologies. Various performance measurements are also used to evaluate these architectures and techniques<sup>6</sup>.

As a result, several researchers have sought to build robust plant disease detection systems that require a high number of disease-infected specimens to be successful. In the past, collecting such a vast number of samples was difficult. Thanks to the Internet of Things, we can now gather and diagnose diseases within the human body! As part of the Plantvillage datasets, there are a lot of photos of corpses with various diseases. Because it is well-labeled and extensively utilized, this dataset has been used in several plant disease detection studies. To maximize their harvests, the farmers also want an easy-to-use detection system that they can use on their phones to identify plant diseases and remove them early. Using image processing methods, Plant disease farmers and researchers may be able to diagnose plant ailments more precisely. Image processing techniques for detecting sickness can also yield satisfactory results, but they require human intervention for other detection and analysis<sup>7-9</sup>.

According to these challenges, we aim to improve the quality of the product and arrive at cleaner production. For plant disease and pest identification using machine vision, the emphasis has switched from standard machine learning and image processing approaches to deep learning techniques in fog environments, which have handled complex previously unsolvable issues. The paper contribution handles four-folds. The first fold is using IoT sensors to generate data and images; there are many sensors used in this field, such as soil moisture, humidity, and temperature, light-dependent resistors, water level, relay module, analog extender, and buzzer ESP 8266. These images were preserved for ten economically and environmentally beneficial plants. Leaf pictures of these plants in ideal and dire circumstances have been collected and dispersed across two categories. In this paper, datasets that focus on plants with significant ecological and economic benefits to their ecosystems are examined. As a result, ten plants, popularly known as Arjun, Mango, Guava, Saptaparni, Jamun, Bael, Sukh Chain, Jatropha, Pomegranate, Basil, Chinar, and Lemon, have been picked. To name just a few, some of these plants have high medicinal value; others are popular for their fruits, and the vast majority are environmentally and economically significant.

Algorithms and models based on deep learning must be successfully integrated with agricultural and plant protection experience to fully exploit AL and ML's potential in the second fold. Three deep learning models applied for plant disease detection are AlexNet, one of the most widely used neural network designs nowadays, GoogleNet, one of the most significant advances in the domain of Neural Networks, notably for CNNs the support vector machine (SVM). These models are utilized to the deep extracted features generated by the pre-trained CNN layers, AlexNet, and we extracted the (fc7) layer as our feature extracting layer. After feeding images into that layer, we can receive features of the images from it. After having all the features, we could use them for training the classifiers. Then, a comparative study among the three models is conducted to show the accuracy of the three models.

The third fold is using a fog environment for computing all necessary tasks of image preprocessing, visualization, monitoring, and local decision support systems for detection and prediction tasks. As a new way of extending and assisting cloud computing, Fog Computing is a rapidly evolving technology. Its proximity to edge users, openness, and mobility, make fog computing platforms ideal for providing services to users quickly and improving the QoS (Quality of Service) of Internet of Things devices. A customer application based on IoT involving real-time activities in agriculture is increasingly reliant on this method<sup>10</sup>.

Lastly, developing a novel version of the grey wolf optimization algorithm (GWO) for selecting the important features to feed to the classifiers. This process is very important to select the relevant features to accelerate the prediction models with fair accuracy. The selected features are fed to the SVM and compared to the standard model, which used all the features from AlexNet.

The remainder of the paper is structured as follows: “[Related work](#)” provides some studies about the recent work. An overview of the basic concepts and methods utilized in this paper is presented in “[Methods and overviews](#)”. “[Proposed methodology](#)” provides the suggested methodology in detail. The experiment setting and results are shown in “[Experimental results and discussion](#)”. “[Conclusion and future work](#)” concludes with a look at what’s next.

## Related work

Abbas et al.<sup>11</sup>, presented a technique based on deep learning for tomato disease diagnosis. To categorize tomato leaf pictures into ten disease categories, the DenseNet121 approach was trained on real and synthetic images using transfer learning. The suggested approach attained an accuracy of 97.11%, 98.65%, and 99.51% for the classification of leaf images into 10 classes, 7 classes, and 5 classes, respectively.

Thenmozhi and Reddy<sup>12</sup>, proposed a powerful CNN approach, and transfer learning is being applied to achieve the best or a desired performance of the pre-training model. Three public insect datasets were used to classify insect species, with accuracy rates of 96.75 percent, 97.47 percent, and 95.97 percent, respectively. Wiesner-Hanks et al.<sup>13</sup>, utilized community data for training a CNN, and nutrition the output into a conditional random field (CRF) to divide pictures into non-lesion and lesion areas with an accuracy of 0.9979 and F1 score of 0.7153.

Too et al.<sup>14</sup>, utilized DenseNets, which have a propensity to always progress in accuracy as the number of iterations increases, with no evidence of performance decay or overfitting. For the classification of plant disease, an accuracy score of 99.75% was achieved. Chen et al.<sup>15</sup>, presented CNN architecture depended on a gliding window to construct a structure for location regression calculation and recognition of pests’ species and plant diseases, feature fusion, characteristics automatic learning, and the identification rate of 38 frequent symptoms was 50–90%. Zhou et al.<sup>16</sup>, demonstrated a rapid approach for the detection of rice diseases founded on the combination of Faster R-CNN and FCM-KM. The sheath blight, bacterial blight, and detection accuracy, and rice blast time were 98.26 percent/0.53 s, 97.53 percent/0.82 s, and 96.71 percent/0.65 s respectively, based on the application results of 3010 images.

Sethy et al.<sup>17</sup> presented 5932 on-field pictures of 4 different kinds of rice leaf illnesses: brown spot, bacterial blight, tungro, and blast. Furthermore, the effectiveness of eleven CNN architectures in the deep feature with SVM and the transfer learning approach was assessed. According to the experimental findings, the deep feature of ResNet50 with SVM outperforms transfer learning equivalent in classification. Deep learning-based methods for identifying illnesses and pests in rice plant pictures have been developed by Rahman et al.<sup>18</sup>. A two-stage tiny CNN design was developed, and it was compared to SqueezeNet, NasNet Mobile, and MobileNet. The simulation findings demonstrated that the suggested framework could attain the necessary accuracy of 93.3%.

Guo et al.<sup>19</sup> presented a mathematical model based on deep learning for the recognition of plant disease and detection. The model was tested for illnesses such as rust diseases, black rot, and bacterial plaque. The results indicated that the accuracy of the model is 83.57%, which is greater than the previous technique, decreasing the impact of illness on agricultural productivity and being beneficial to agriculture’s long-term improvement. Atila et al.<sup>20</sup>, presented the EfficientNet model for the plant leaf disease classification, and the performance of the model was compared to existing previous deep learning techniques. The experimental findings revealed that the B4 and B5 approaches of the EfficientNet attained the greatest rates in the original and enhanced datasets, with the accuracy of 99.91% and 99.97%, and precision of 98.42% and 99.39% respectively. There are many studies for plant disease prediction as in Refs.<sup>7–9,21</sup>.

Table 1 summarizes the role of ML/DL in agriculture for plant diseases classification using accuracy measurement as mentioned by many authors. It is observable that most of the recent works use the PlantVillage dataset and deploying a set of pre-trained CNN models. In this paper, new datasets have been used for testing our proposed architecture for plant disease classification.

The next section handles an overview of the problem statement and the used methods in this paper.

## Methods and overviews

### Overviews

Climate change, population expansion, and food security concerns have pushed the sector to explore more creative ways to agricultural yield protection and improvement. As a result, artificial intelligence (AI) is progressively developing as a component of the industry’s technical growth<sup>31</sup>. Popular applications of traditional machine learning algorithms in agriculture are:

- Recognition/harvesting of vegetables and fruits.
- Plant disease classification/pest detection.
- Crop/weed discernment and classification.

Author	Dataset	Method	Result
<sup>22</sup>	Banana leaf images obtained from banana field	ResNet-152	99.2%
<sup>23</sup>	PlantVillage	VGG-19	98.3%
<sup>24</sup>	PlantVillage	Multi-Scale AlexNet	92.7%
<sup>25</sup>	PlantVillage	ResNet34	99.7%
<sup>26</sup>	PlantVillage	CNN	91.2%,
<sup>27</sup>	Collected data from field	Custom-Net	98.78%
<sup>28</sup>	6 Common cucumber leaf diseases taken in field	GPDCNN	95.18%
<sup>29</sup>	PlantVillage	CAE	86.78%
<sup>30</sup>	PlantVillage	CAE + CNN	98.38%

**Table 1.** ML/DL for plant disease detection.

- Plant/leaves recognition and classification.
- Land cover classification.

In comparison to the defined segmentation, detection, and classification tasks in computer vision, the criteria for detecting pests and plant diseases are quite broad. Its needs may be classified into three categories: what, where, and how. Even though, the fact that the function needs and aims of the 3 phases of plant disease and pest detection are distinct, the three stages are mutually inclusive<sup>32</sup>. The classification job in computer vision is represented by "what" in the first step. Classification defines the image globally using feature expression and then decides if the image contains a certain type of object using the classification process. While structure learning is the primary research path in object detection, feature expression is the primary research path in classification tasks.

Machine learning (ML) has developed alongside high-performance computation and big data technologies to open up new avenues for unraveling, quantifying, and comprehending data-intensive processes in agricultural operational contexts. ML offers machines the capacity to learn without being precisely programmed<sup>33</sup>. Convolutional neural networks (CNNs) are more complex to construct than traditional neural networks, but they are simpler to utilize. It is not required to extract picture characteristics independently in the case of this sort of neural network. In image classification problems, complex and pre-trained CNNs with millions of parameters are frequently utilized. Their complete training is difficult since it is a time-consuming and labor-intensive procedure<sup>34</sup>. With developments in machine learning (ML) principles, significant gains in agricultural activities have been noticed. The capacity to extract features automatically generates an adaptable nature in deep learning (DL), especially CNNs, which achieves human-level accuracy in a variety of agricultural applications, prominent among which are crop/plant recognition, fruit counting, land cover classification, weed/crop discrimination, and plant disease detection and classification<sup>35</sup>.

## Methods

### *Transfer learning.*

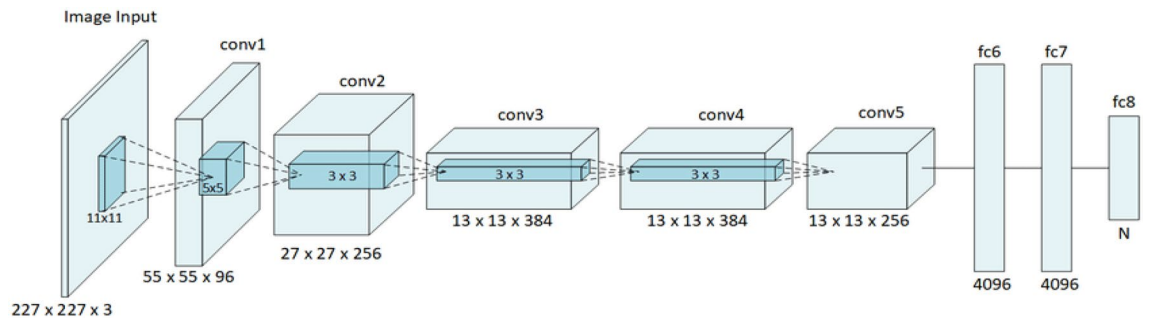
Data Transfer Learning (DTL) is a strategy in which knowledge derived from the data is transferred to solve various but associated assignments to train the CNN, including new data that often comes from a lower population<sup>36</sup>. To initialize the models and pre-train two profound convolutionary neuro-network models, transfer learning was used: AlexNet and GoogleNet.

AlexNet is expected to be the first recommended deep CNN technology due to its remarkable outcomes for the identification and classification functions on image data<sup>37</sup>. In an attempt to improve hardware constraints and obtain the total functionality of deep CNN, AlexNet was trained on two parallel GPUs. In AlexNet, the CNN depth was widened from only five layers in the LetNet CNN to eight layers in the way to produce CNN appropriate to different data sets of images. Dropout, ReLU, and pre-processing are major attributes to attain significant improvement in computer vision applications. The common 8 layers are five convolutional layers, two fully connected hidden layers, and one fully connected output layer, as shown in Fig. 1<sup>38</sup>.

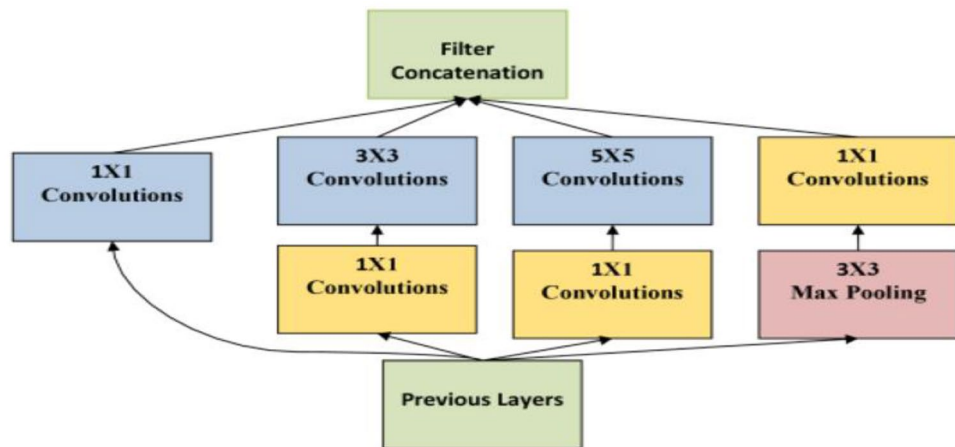
In this study, we replaced the 1000 classes that the original AlexNet had, with only 2 classes which we evaluated in this paper, healthy images and diseased images of 10 different plants as illustrated in the dataset description.

GoogleNet consists of 22 layers deep CNN that is a version of the inception network established by Google researchers. The design of the GoogleNet structure resolved many constraints that appeared for large networks, primarily out of the use of the Inception module. The structure diagram of the GoogleNet network is shown in Fig. 2<sup>39</sup>.

GoogleNet consists of inception modules, so its architecture is complex. GoogleNet is looked like one of the initial CNN architectures to resist successively accumulating convolutions and pooling layers. In addition, GoogleNet plays a vital role in consideration of storage and power, since accumulating all tiers and combining different restrictions would take time for computation and will result in higher costs of memory<sup>40</sup>.



**Figure 1.** An illustration of AlexNet layers.



**Figure 2.** GoogleNet network structure diagram.

*Support vector machine*

The deep feature extraction technique necessitates the training of a classifier method with the extracted features. Vapnik’s SVM was utilized as a classifier in this study<sup>41</sup>. It has been found that the SVM classifier outperforms others in several agricultural image categorization tasks.

The support vector machine is a classifier with a linear or non-linear relationships that is capable of distinguishing between two different types of objects. SVMs are machine learning approaches focused on cambered improvement that operate as stated by the concept of structural risk reduction. These approaches are separate of distribution, as it does not need any details on the common distribution functions<sup>42</sup>. SVM training can be illustrated with algorithm 1<sup>43</sup>.

**Algorithm 1** SVM Training

**Require:**  $y$  and  $X$  loaded with labeled training data,  $\alpha \Leftarrow$  not or partially trained SVM

- 1:  $C \Leftarrow$  some value (for example 10)
- 2: **repeat**
- 3:   **for all**  $\{x_i, y_i\}, \{x_j, y_j\}$  **do**
- 4:     Optimize  $a_i$  and  $a_j$
- 5:   **end for**
- 6: **until** no other restriction criteria are presented or changes in  $\alpha$

**Guarantee:** Keep only the support vectors ( $a_i > 0$ )

While a hyperplane classifier can distinguish between 2 classes, certain categories surpass the highest distance set as the most effective separation hyperplane. The objective of SVM is to construct an ideal hyperplane space by utilizing training sets<sup>40</sup>.

The main idea behind using SVM to solve a classification issue is to find a hyperplane that best separates data from two groups. The formula for a linear SVM’s output is presented in Eq. (1), where  $\vec{w}$  is the hyperplane’s normal vector and  $\vec{x}$  is the input vector. Margin maximization may be thought of as an optimization issue: reduce Eq. (2) subject to Eq. (3) where  $y_i$  and  $\vec{x}$  are the SVM’s correct output and input vector for the  $i$ th training sample, respectively<sup>44</sup>.

$$u = \vec{w} \cdot \vec{x} - b \tag{1}$$

$$1/2 \|\vec{w}\|^2 \tag{2}$$

$$y_i(\vec{w} \cdot \vec{x} - b) \geq 1, \forall i \tag{3}$$

SVM is a binary classifier that can only distinguish between two classes and does not handle multi-class classification issues. One approach to classification of multi-class using SVMs is to build a one-to-one group of classifiers and forecast the class picked by the majority of classifiers<sup>45</sup>. While this allows for the creation of K (K + 1)/2 classifier for the classification issue with K classes, the classifiers training time may be decreased because the training data set for every classifier is lower. In this article, SVM is used to analyze data in addition to CNN techniques such as AlexNet and GoogleNet.

*Grey wolf optimization (GWO) algorithm*

Mirjalili et al. proposed the grey wolf optimizer (GWO) as a novel swarm intelligence method<sup>46</sup>. The GWO method has been successfully utilized and applied in a variety of research. The primary inspiration for the GWO algorithm came from the social pursuit of grey wolves in nature. Figure 3 depicts the social hierarchy as well as an instance of the position update process<sup>47</sup>.

In the GWO algorithm, the first, second, and third best-recommended solutions are alpha (α), beta (β), and delta (δ). Omega is projected to be the outstanding solution. The wolves can be presented in a form that is representable mathematically in Eqs. (4–8) during the hunting process:

$$\vec{D} = \left| \vec{C} \cdot \vec{X}_p - \vec{X}_{(t)} \right| \tag{4}$$

$$\vec{X}_{(t+1)} = \vec{X}_{p(t)} - \vec{A} \cdot \vec{D} \tag{5}$$

$$\vec{A} = 2 \vec{a} \cdot \overrightarrow{rand1} - \vec{a} \tag{6}$$

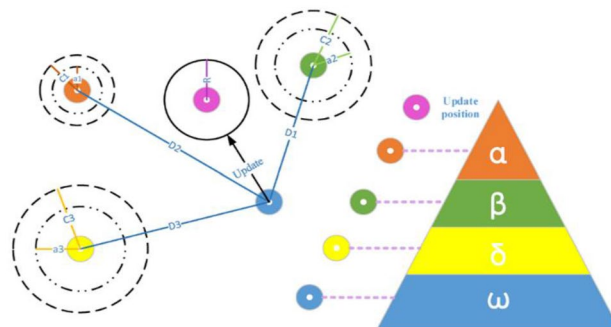
$$a = 2 \times \left( 1 - \frac{t}{t_{max}} \right) \tag{7}$$

$$\vec{C} = 2 \cdot \overrightarrow{rand2} \tag{8}$$

where x is the grey wolf’s vector position, xp is the prey’s vector position, D is the distance between x and xp, t is the current iteration number, and A and C correspond to component-wise multiplication.

The "A" parameter is given in a [- a, a] random value according to the "a" value. Whether a gray wolf attacks or not, the value of A can be determined. As a result of the calculation, the gray wolf is exceptionally close to the hunt and can attack at any time if |A < 1| status is available. The gray wolf leaves a beast in the case of |A > 1|, hoping for a new beast. Another critical parameter of control, C, is recognized as the exploration component of the algorithm and may include random values within the range [0, 2]. This variable leads to a random behavior of the algorithm that prevents an optimization at optimum local values. This condition happens if the random conduct is minimized by |C < 1| or else |C > 1|<sup>47</sup>.

To mimic grey wolf hunting behavior, Eqs. (9–14) show how grey wolves are positions updating of α, β, and δ wolves. It is accepted that the wolves of α, β, and δ are closest to the prey and attract the rest of the wolves to the prey area. The grey wolf population can use the following formulae to determine prey position:



**Figure 3.** GWO’s social structure and position update method.

$$\vec{D}_\alpha = \left| \vec{C}_1 \cdot \vec{X}_\alpha - \vec{X} \right| \quad (9)$$

$$\vec{D}_\beta = \left| \vec{C}_2 \cdot \vec{X}_\beta - \vec{X} \right| \quad (10)$$

$$\vec{D}_\delta = \left| \vec{C}_3 \cdot \vec{X}_\delta - \vec{X} \right| \quad (11)$$

$$\vec{X}_1 = \vec{X}_\alpha(t) - \vec{A}_1 \cdot (\vec{D}_\alpha) \quad (12)$$

$$\vec{X}_2 = \vec{X}_\beta(t) - \vec{A}_2 \cdot (\vec{D}_\beta) \quad (13)$$

$$\vec{X}_3 = \vec{X}_\delta(t) - \vec{A}_3 \cdot (\vec{D}_\delta) \quad (14)$$

The locations are determined from Eqs. (12–14) is utilized to modify the next location of the wolves by Eq. (15):

$$\vec{X}_{(t+1)} = \frac{(\vec{X}_1 + \vec{X}_2 + \vec{X}_3)}{3} \quad (15)$$

where  $x_{t+1}$  is the location of the next iteration. Using Eq. (15) to find a new location for the leading wolves drives the Omega wolves to change their locations to converge with prey.

The GWO algorithm sequence consists of three steps: initialization, fitness calculation, swarm individual position updates, and the best result generation. The optimization process starts with the starting value for all control parameters, and all gray wolves are altered in regular intervals. The fitness function is then calculated based on the initial data, and the best solutions are identified as wolves of alpha, beta, and delta. The next step is to upgrade all gray wolves other than delta, beta, and alpha wolves. The next step is to renew all gray wolves' positions and controller parameter values, followed by Alpha, Beta, and delta wolves. Finally, the alpha wolf returns its optimal position value.

#### *Fog computing and IOT*

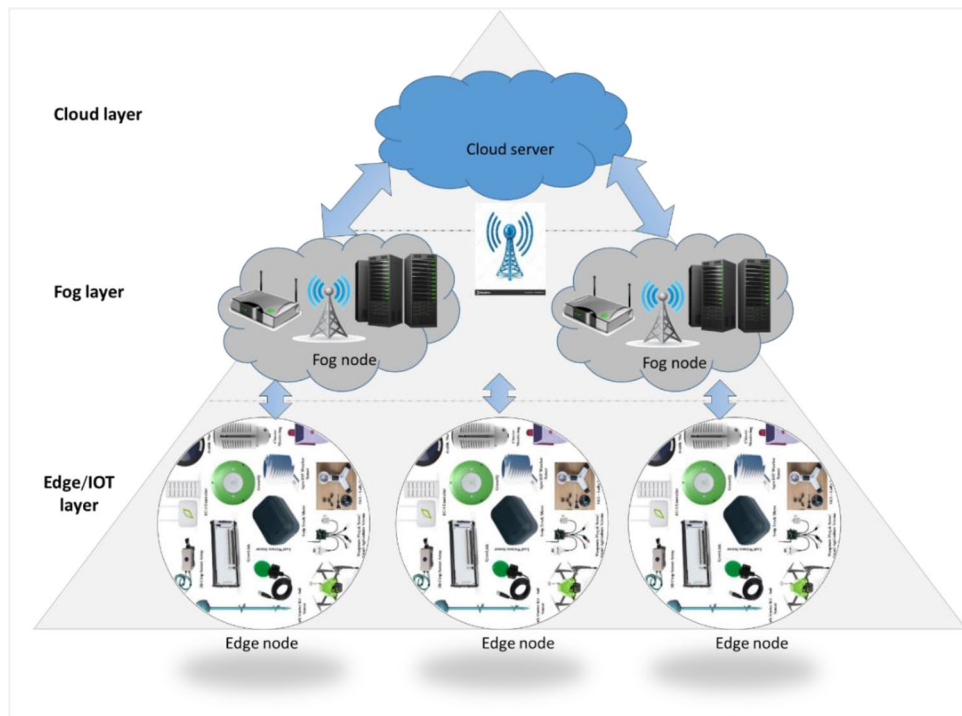
The providers of cloud computing frequently utilize data centers that consider a variety of factors, including energy consumption and user proximity. Thus, the cloud layer, the top layer, often comprises a cloud infrastructure made up of data centers that provide resources and amenities that are dynamically assigned according to the demands of the users. These services could include networking, storage, and server (rendering tools, computational power, and so on) capabilities<sup>48</sup>. Fog Computing attempts to bring processing capabilities closer to end-users, preventing overuse of Cloud resources, further lowering computational burdens, enhancing load balancing, and shortening wait times<sup>49,50</sup>.

The Internet of Things (IoT), which represents the future of communications and computers, is a breakthrough technology. IoT is now used in almost every sector, including intelligent cities, intelligent traffic control, and intelligent homes. The deployment of IoT is wide and may be applied in any field. IoT aids in better resource and crop management, crop monitoring, cost-effective agriculture, and increased quantity and quality. Air temperature sensors, soil moisture, soil pH, water volume, humidity, and other IoT sensors are employed<sup>47</sup>. Figure 4 shows IoT in agriculture using edge computing, fog computing, and cloud computing.

The key benefits of IoT in agriculture are discovered in these points<sup>51</sup>:

- Community agriculture in rural and urban regions, utilizing software and hardware resources as well as vast amounts of data.
- Quality and logistical traceability of food security that allows reduced costs via real-time decision-making data.
- Business strategies established in the agricultural setting that enable direct consumer contact.
- Crop surveillance allows cost savings and machine robbery avoidance.
- Systems of automatic irrigation that operate based on soil moisture levels, and temperature measured by sensors.
- Environmental characteristics are automatically collected via sensor networks for subsequent analysis and processing.
- Large quantities of data are analyzed by decision support systems to increase production and operational efficiency.

At the end of this section, we can summarize this paper in 3 folds; the first is applying DL models (AlexNet, GoogleNet) to extract features from plants. Secondly is using an optimization algorithm called the modified grey wolf optimization algorithm for eliminating the redundant features. The third is the classification of the output images using the support vector machine. The above techniques are divided to be used some processes in Fog and some processes in cloud computing. The next section introduces the architecture of the proposed solution using the deep learning techniques referred to above.



**Figure 4.** Smart agriculture IoT with edge, fog, and cloud computing.

### Proposed methodology

As technology advances, smart agricultural solutions are becoming more prevalent. Since then, technology has returned to agriculture with the latest trends and techniques it has produced. A significant advantage of smart agriculture is connecting to existing 3G and 4G networks using existing hardware and software. For smart agriculture solutions, it speeds up setting up hardware, resulting in the various successful implementation of IoT in agriculture that can run in a fog or cloud environment. There will be an evolution from the existing standard mobile computing scenario of smartphones and their apps to the connection of gadgets around us to help solve a real-world problem<sup>52</sup>. We'll discuss in this section the proposed methodology based on the mentioned transfer learning, pretraining methods, and the optimization algorithm on fog and cloud computing using IoT sensors common in the problem statement of this paper.

Figure 5 shows the block diagram of the proposed IoT smart agriculture network architecture which consists of three layers. The first layer contains the IOT devices that are used for different purposes in agriculture. Many technologies are being used in IoT agricultural solutions which have an important role in modernizing the services of IoT agriculture. Examples of these technologies are cloud and edge computing, machine learning and big data analytics, communication networks and protocols, and robotics. The second layer presents the sequence of work in this paper from collecting the images from IoT sensors then preprocessing these images if they need to resize, or normalize, or removing noise according to the recommended DL algorithms in this paper (CNN, SVM). All processes happened on the images from collecting it till detection of plant diseases are applied on fog environment to facilitate the function of scalability and stability that are advantages of fog computing. The third layer is connecting with cloud computing for henting resources for further and large processing. The other proposed models don't suitable for cloud or fog computing, so we proposed a new model for plant disease detection using machine learning techniques by the Internet of Things (IoT) sensors that can run on fog or cloud environments.

The proposed model depends on deep learning, transfer learning, and shallow machine learning. In deep learning, multi-hidden layers are stacked for learning objects significantly. These layers require a training process including "fine-tuning" for adjusting the weights slightly of DNN discovered during the procedure of backpropagation. In turn, following an efficient training procedure, DL nets can categorize, extract characteristics, and give a decision effectively and accurately. In the proposed model, we use transfer learning to optimize different pre-innovated CNNs architectures to the datasets.

As seen in Fig. 6, the proposed model starts by a data acquisition layer in which images are collected for different plants. This acquisition procedure was entirely wi-fi enabled, which means that the camera and the computer were linked with each other via the internet. In the preprocessing phase, the images are reconstructed and resized since the images are taken from various sources and their dimensions vary. In addition, the model layer of each of these products needs separate image dimensions to be managed. Therefore, the input image size is adjusted to fit the templates that are used in this analysis.

The feature extraction layer comes after image enhancements that represent the layer in which most of the calculations are carried out. The calculations include extracting image data set features and preserve the spatial relationship between image pixels. A pre-trained CNN, AlexNet, was used as feature extraction and we extracted



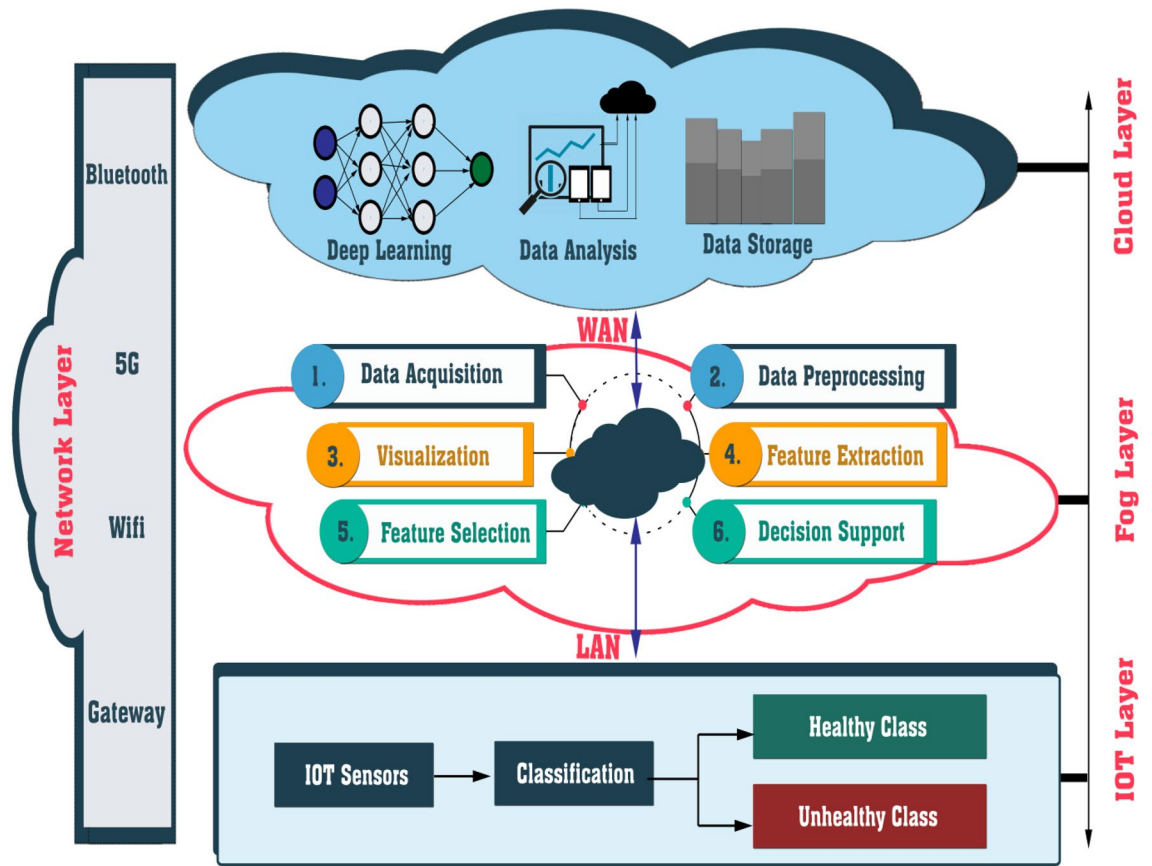


Figure 5. Block diagram of the proposed IoT smart agriculture network architecture.

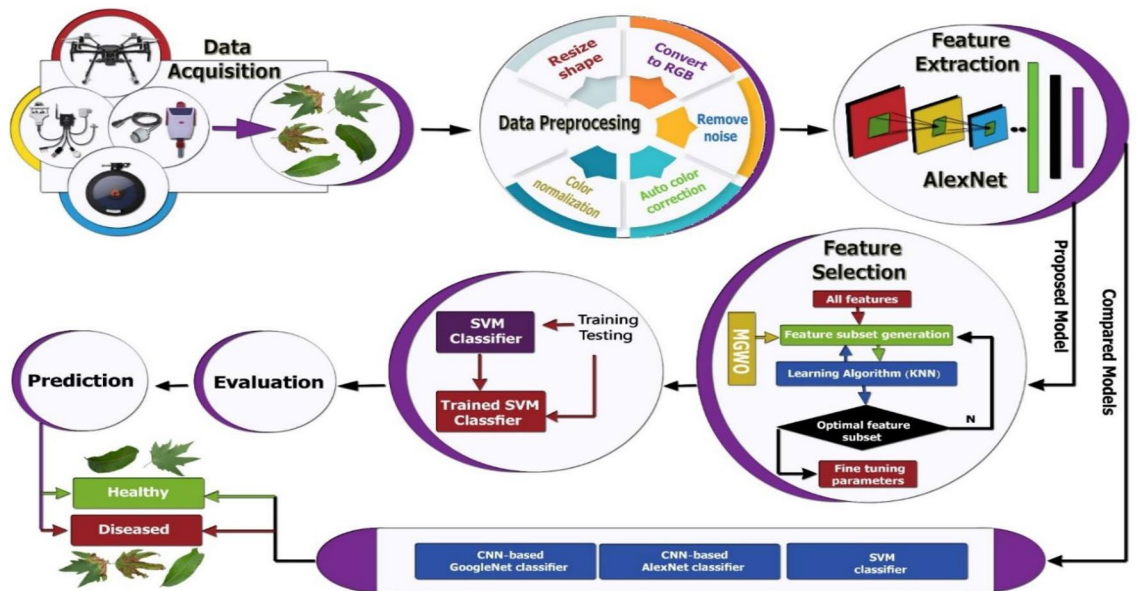


Figure 6. Dataflow diagram of the proposed methodology.

the 'fc7' layer as our feature extracting layer. After extracting the features, it is the role of feature subset selection to reduce the features and eliminate the irrelevant features. The proposed model makes use of a modified version of the grey wolf optimization algorithm. The details of the modified grey wolf optimization algorithm (MGWO) are explained in the next subsection. After that, the generated features set were utilized to train the SVM. Once we get the baseline SVM, we use a validated data set to adjust SVM parameters.

### Modified grey wolf optimization algorithm (MGWO)

Mirjalili showed that The GWO algorithm tends to become stuck at optimal local values because of the small number of control parameters utilized in its simplest form. Because of this, researchers modified GWO by adding additional controls and changing control parameter values. According to their findings, the alpha-wolf was more powerful than the delta—and beta-wolf when searching for food. So, it's possible to acquire better outcomes in tests in this manner. For this reason, there is much research in the literature that has adapted and developed the grey wolf algorithm in various sectors. As a result of this, it produces superior outcomes in tests<sup>47</sup>.

The parameter adjusted equation for the "a" parameter was used in this study to improve the method<sup>50</sup> significantly. However, instead of using the usual GWO Eq. (7), this study uses Eq. (16) to derive the parameter "a" instead.

$$a = 2 \times \left( e^{-\frac{t \times s}{t_{max}}} \right) \quad (16)$$

"s" is only added in Eq. (16) to "a", and it reflects the total number of individuals in the swarm, as seen in Eq. (16). Standard GWO has a linearly decreasing "a" parameter, which prevents the algorithm from settling on local minimum values. Researchers found that as the "a" attribute approaches 0, it not only keeps the algorithm from reaching a locally minimal value but also considerably enhances its strength. Therefore, the method converges on the optimal values faster when this parameter is reduced from 2 to 0. So, the program has sped up and parabolically slowed down from 2 to 0.

Moreover, it can be seen from the governing Eq. (15) that the dominants perform a similar function in the searching procedure; each of the grey wolves converges or flees away from the surroundings with an average weight of the beta, delta, and alpha. Even if the alpha is closest to the victim at first, it may be distant from the eventual result. Only the alpha position should be considered in Eq. (15) at the beginning of the search operation, or its weight should be substantially more significant than that of other dominants. Equation (15)'s average weight, on the other hand, contradicts the grey wolf social hierarchy idea. If the pack's social hierarchy is strictly observed, the alpha is in charge, which could mean that he/she will always be the closest to the prey. This indicates that the alpha wolf's weight in Eq. (15) should never be smaller than that of the delta and beta wolves. As a result, the beta's weight should always be more than the delta's. In light of these concerns, the authors<sup>53</sup> further hypothesize the following:

- (1) The dominants surround a supposed prey when it is being searched for, but they do not surround an actual prey when being hunted. As their social hierarchy dictates, the dominant grey wolves encircle the prey in order of their dominance. The alpha is the closest wolf in the pack, followed by the beta and the delta. Omega wolves play a role in this process, passing on their superior positions to the dominants.
- (2) Alpha controls the search and hunting process, while beta has a minor role, and delta has an even smaller one. A wolf's position changes if an alpha wins out over his/her peers.

Equation (15) should not use the same updating procedure for the positions as the previous hypotheses. Thus, the alpha weight should be near 1.0 at the beginning, whereas delta and beta weights could be close to 0. According to Eq. (15), the delta, beta, and alpha wolves should surround the victim at the final stage. During the entire process of searching, the alpha is always found by the beta, and the beta always finds the delta because he/she is always ranked third. As a result, the beta and delta weights are determined by the total number of iterations. Alpha's weight should be reduced, and beta and delta's weights should rise.

In mathematics, the above ideas could be stated. When adding up the weights, ensure that they're all varying and that the aggregate is capped at 1.0. As a result, Eq. (15) is altered to the following:

$$\vec{X}_{(t+1)} = w_1 \vec{X}_1 + w_2 \vec{X}_2 + w_3 \vec{X}_3 \quad (17)$$

$$w_1 + w_2 + w_3 = 1$$

As a second rule, when calculating the alpha  $w_1$ , beta  $w_2$ , and delta  $w_3$  weights, they should always satisfy  $w_1 > w_2 > w_3$ . Along with the search technique, the weight of the alpha would be adjusted from 1.0 to 1/3. While doing so, we'll boost beta and delta's weights, increasing them from 0.0 to 1/3 in the process.  $w_1$  can be described using a cosine function if we limit the angle range to be between  $[0, \arccos(1/3)]$ . The weights should be adjusted based on the total iterations or "it" as a third point. And we recognize that  $w_2 \cdot w_3 \rightarrow 0$  if it = 0 and  $w_1, w_2, w_3 \rightarrow 1/3$  when it  $\rightarrow \infty$ . As a result, we present an arc-tangent function that changes from 0.0 to  $\pi/2$ . And then, somehow,  $\cos(\pi/4) = \sin(\pi/4) = \frac{1}{\sqrt{2}}$ , so different angular parameter  $\varphi$  was organized as seen below<sup>53</sup>:

$$\varphi = \frac{1}{2} \arctan(it) \quad (18)$$

Given that  $w_2$  would be maximized from 0.0 to 1/3 alongside it, we assume that it includes  $\cos \varphi$  and  $\sin \theta$  and  $\theta \rightarrow \arccos(1/3)$  when it  $\rightarrow \infty$ ; hence,

$$\theta = \frac{2}{\pi} \arccos \left( \frac{1}{3} \cdot \arctan(it) \right) \quad (19)$$

when it  $\rightarrow \infty$ ,  $\theta \rightarrow \arccos(1/3)$ ,  $w_2 = 1/3$ , we can then formulate  $w_2$  in detail. The following is a new updating method for positions with variable weights that are based on these principles:

$$w_1 = \cos \theta, \quad (20)$$

$$w_2 = \frac{1}{2} \sin\theta \cdot \cos\varphi, \tag{21}$$

$$w_3 = 1 - w_1 - w_2 \tag{22}$$

The flowchart of the Modified Gray Wolf Optimization (MGWO) algorithm is shown in Fig. 7. The pseudocode of the MGWO is presented in algorithm 2.

**Algorithm#2 The MGWO**

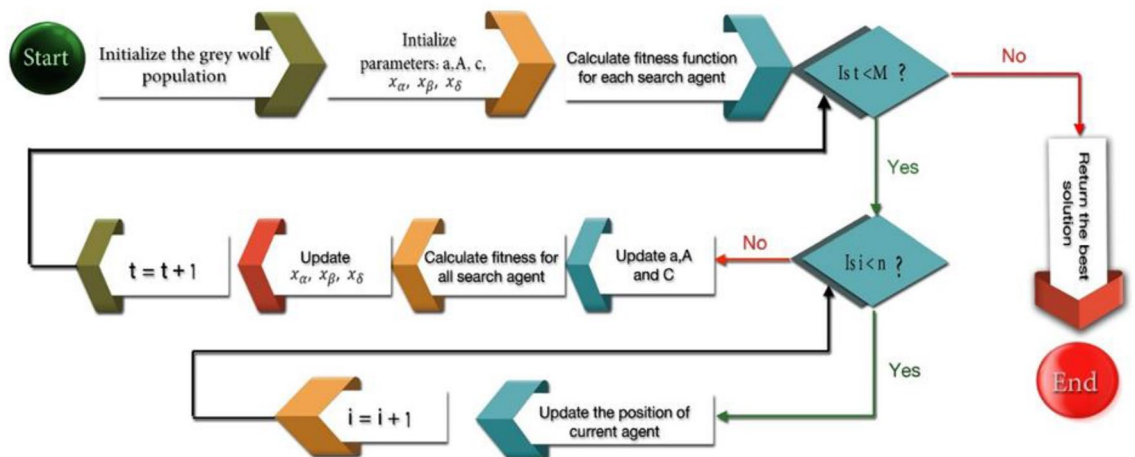
**Input** ← Learning Factors ( $c_1, c_2$ ), Control Parameter ( $a$ ), Swing Factor ( $C$ ), Population Size ( $N$ ), Grey Wolf Population ( $X$ ), iteration number ( $T_{max}$ )

**Output** ← Return the  $\alpha$  (The optimal search space solution)

1. Begin
2.   **For**  $t=1: T_{max}$                                     *// for all iterations*
3.     **For**  $i=1: N$                                       *// for all agents*
4.      **For**  $j=1: d$                                     *// for all agent dimensions*
5.       Generate  $\vec{D}_\alpha, \vec{D}_\beta, \vec{D}_\delta$  according to Eqs. (9,10,11)
6.       Generate  $\vec{x}_1, \vec{x}_2, \vec{x}_3$  with respect to Eqs. (12,13,14)
7.       Generate  $w_1, w_2, w_3$  according to Eqs. (20,21,22)
8.       Position updating using Eq (17)
9.      **End for**
10.     **End for**
11.     Fitness evaluation and modify ( $\alpha, \beta, \delta$ )
12.     Generate the new value for the ( $a$ ) according to Eq (16)
13.     Update  $\vec{A}$  and  $\vec{C}$  according to Eqs. (6,8)
14.     Update  $x_1, x_2, x_3$
15.    **End for**
16. **End**

**Experimental results and discussion**  
**Performance measures**

True positives, true negatives, false negatives, and false positives, and are displayed separately in the table, in two rows and two columns, accordingly (sometimes also referred to as a confusion matrix). In this way, the classification proportion can be studied in greater detail (accuracy). An unbalanced data collection (i.e., when the number of observations in different classes changes dramatically) will lead to inaccurate conclusions. Sensitivity and specificity are also valuable traits to have. As shown in Table 2<sup>54</sup>, the most widely used measures are used to create the confusion matrix (Data science, 2019). Five measurements are utilized in this article to gauge our work performance, these measurements are shown in Table 3.



**Figure 7.** Flowchart of the grey wolf optimization algorithm.

Actual output		Output class	
		Positive	Negative
Real class	Positive	True positives (TP)	False negatives (FN)
	Negative	False positives (FP)	True negatives (TN)

**Table 2.** The confusion matrix.

Metric	Meaning	Calculation	Equation
Sensitivity	(True Positive Rate) assesses correctly detected proportions of positive (i.e., the balance of those correctly detected as having a condition)	Sensitivity (Recall) = $\frac{TP}{TP+FN}$	(23)
Specificity	(The fraction of the negative that is correctly detected (i.e. the ratio of the non-conditional (uninfluenced) erroneously identified as non-condition) (True Negative Rate)	Specificity = $\frac{TN}{TN+FP}$	(24)
Precision	The ratio for all positive examples adequately classified, and the total number of positive models forecast. In positive prediction, it shows the correctness achieved	Precision = $\frac{TP}{TP+FP}$	(25)
Accuracy	The ratio of the entire number of correct forecasts. Precision is the proximity between the measures to a given value in a set of measurements, while precision is the proximity	Accuracy = $\frac{TP+TN}{TP+TN+FP+FN}$	(26)
F1 score	The weighted (sensitivity) and the accurate average of recall. F1, if you are trying to balance precision and reminder, maybe the right choice	F1Score = $2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$	(27)

**Table 3.** Performance measures.

### Experiment 1

We provided the results as two experiments. For the first experiment, a modified grey wolf optimization method (MGWO) for feature selection is being evaluated. When developing a machine-learning model, feature selection is becoming increasingly important. The feature selection process involves deleting irrelevant or redundant characteristics and picking the ideal subset of features that better categorize patterns that belong to different plants. The evaluation is made by using fifteen standard feature selection datasets. The overall properties of these datasets are given in Table 4<sup>55</sup>.

Using a random seeding strategy, a random population of  $n$  wolves or search agents is formed in the first part of the procedure. An ideal solution is found when the number of features "d" equals that of the original dataset features set. When selecting features for purity classification, make sure they enhance accuracy. Identify the appropriate characteristics (one value) and discard the rest (zero). Initially, the binary values (0 and 1) were set in each solution.

A large part of GWO's success depends on the development of initial populations. We use chaotic initialization of maps to increase the global convergence speed of the MGWO optimization process. Instead of a standard map, a chaotic map serves to improve the balance of search-and-exploitation skills. The logistics map is one of the most effective chaos-based approaches. It is represented as follows in Eq. (28)<sup>56</sup>.

$$X_{n+1} = \phi(X_n, \mu) = \mu \times X_n(1 - X_n) \quad (28)$$

ID	Dataset	No. of instances	No. of attributes	No. of classes
1	Breast cancer	699	9	2
2	Climate	540	20	2
3	Diabetic	1151	19	2
4	Ionosphere	351	34	2
5	kc1	2110	21	2
6	Lung Cancer	226	23	2
7	lymphography	148	18	4
8	pc1	1109	21	2
9	Stock	950	9	2
10	Segment	2310	19	7
11	WineEW	178	13	3
12	Tic-tac-toe	958	9	2
13	Vote	300	16	2
14	WDBC	569	30	2
15	Zoo	101	16	7

**Table 4.** Datasets used for evaluating the MGWO.

where  $\mu$  is set to 4, the bifurcation coefficient is also defined.  $x_n$  means the  $n_{th}$  chaotic variable, in other words,  $x_n \in (0, 1)$  in favor of limitations that the initial  $x_0 \in (0, 1)$  of severely static periods  $(0, 0.25, 0.5, 0.75, 1)$ . Figure 8 of the logistic map shows a consistently divided sequence, which prevents it from effectively immersing into minor regular cycles.

Because the problem has more than one objective, it is understood to be a multi-objective problem<sup>57</sup>. Following steps must be taken to solve the multi-objective issue of selecting optimal features. The first is to produce the highest accuracy rate, and the second is to eliminate the features to the lowest range. Taking this into consideration, the fitness function of the resulting solution evaluation is configured to balance the aims as follows:

$$\text{fitness} = \alpha \gamma_R(D) + \beta \frac{|S|}{|D|} \tag{29}$$

Given that  $|S|$  for the length of the selected subset feature cardinality,  $\alpha$  and  $\beta$  are generated as parameters for expressing a weight for the percentage of classification accuracy and the total number of selected features respectively,  $\alpha \in [0, 1]$  and  $\beta = 1 - \alpha$  and have been selected concerning the evaluation function,  $\gamma_R(D)$  denotes the classification error rate.  $|D|$  represents dataset cardinality. So, to find the K neighbors for the KNN classifier, the Euclidean distance<sup>58</sup> must be calculated as follows:

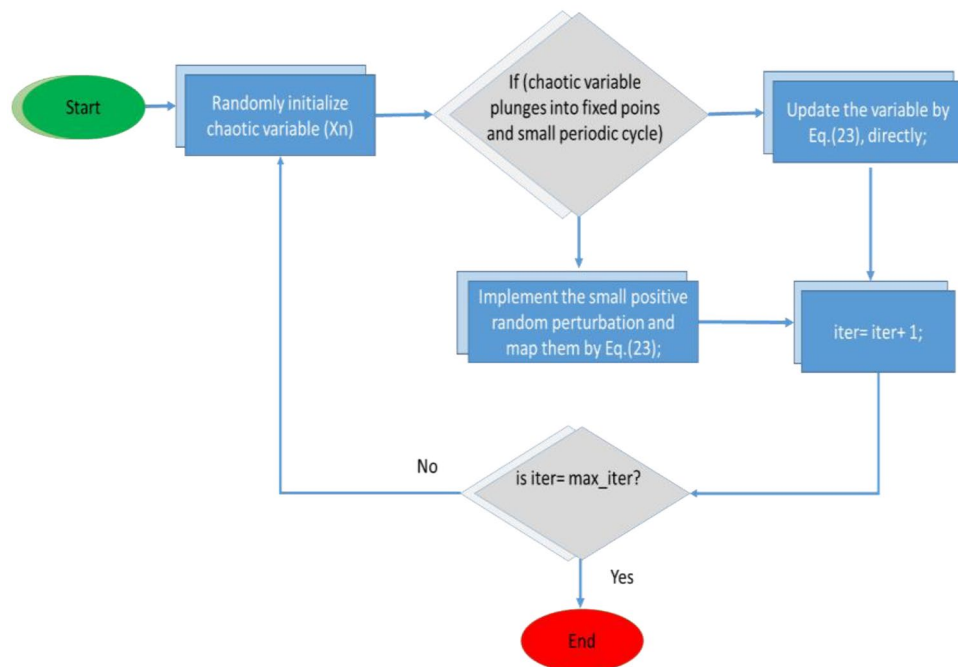
$$\text{EUC}_d(P, Q) = \sqrt{\sum_{i=1}^d (Q_i - P_i)^2} \tag{30}$$

$Q_i$  and  $P_i$  relate to a specific feature in the sample, "i" refer to the number of features in the sample, and d refers to the overall number of features used in the analysis. To reduce overfitting, cross-validation is a popular strategy. Cross-validation with  $K = 10$  is utilized in this paper.

In contrast to binary values, continuous values represent the positions of the search agents formed by the algorithm. A straight application to our situation would be impossible because it is incompatible with the standard binary format for feature selections. Features are selected depending on the problem of feature selection to increase the performance and accuracy of the classification system (0 or 1 in the case of binary features). A transformation function is used to convert a binary search space. The following equations can convert any continuous value into binary with the sigmoid function<sup>57</sup>:

$$x_{s_i} = \frac{1}{1 + e^{-10(x_i - 0.5)}} \tag{31}$$

$$x_{\text{binary}} = \begin{cases} 0 & \text{if } R < x_{s_i} \\ 1 & \text{if } R \geq x_{s_i} \end{cases} \tag{32}$$



**Figure 8.** Flowchart of logistic map for initialization.

where  $i = 1, \dots, d$ , and  $x_{binary}$  parameter identified as 0 or 1 by randomly selected value in range:  $R \in [0,1]$  value compared to  $x_i$ , the value of the parameter  $x_i$ , which defined in the S-shaped search agent is the value identified by the algorithm calculations (continuous). All trials were conducted on a Windows 10 Pro 64-bit operating system with a Core(TM) i7-8550U CPU running at 1.80 GHz and 1.99 GHz, respectively. To implement the algorithms, we use MATLAB (2018a).

The selected values of the algorithms to be its parameters were collected from the literature to make sure that the algorithms are compared on an equal basis<sup>59</sup>. Although the KNN classification unit for feature selection is a frequent wrapper, it can also be thought of as a learning algorithm that is monitored and characterized by simple and quick learning. There are twenty different runs for each algorithm with a random seed. The maximum number of iterations for all subsequent experiments using the standard k-fold cross-validation is 20.

Multiple observational experiments were conducted on a variety of datasets to determine the best literature values for  $\alpha$  and  $\beta$ . Therefore, it has the value of 0.9 for  $\alpha$  and has the value of 0.1 for  $\beta$ . The parameters settings of our experiments are shown in Table 5.

Tables 6 and 7 show the resulted feature and the accuracy respectively. The experimental results are conducted for the standard grey wolf optimization (GWO), the Ant Colony Optimization (ACO), the Butterfly Optimization Algorithm (BOA), the Particle Swarm Optimization (PSO), and the Modified Grey Wolf Optimization (MGWO) algorithms. The experimental results show the superiority of the proposed MGWO in both achieving the least set of features in all the datasets while producing a fair accuracy in most of the utilized datasets. These results are graphically displayed in Figs. 9 and 10.

According to the conclusion of these results, we can say that the MGWO can be used for our plant disease problem.

## Experiment 2

According to the experimental result in the first experiment, the modified grey wolf optimization algorithm (MGWO) can be effective as a wrapper feature selection algorithm. In experiment 2, the core problem of the plant disease classification and prediction is introduced. As discussed in the previous section, the first stage of the proposed model is the feature extraction process where the pre-trained AlexNet CNN is used. This process is performed for ten datasets. The second stage is the feature selection process, in which the MGWO is used as the wrapper feature section method. Lastly, the generated reduced features set were used SVM training. The datasets' details are discussed in the next subsection.

### Datasets description

Plants play a crucial role in climate regulation and erosion reduction. To preserve the environment, ecosystem, and living beings, they are both equally necessary to consider. Deciduous and coniferous trees are the most

Parameters	Value	Parameters	Value
Independent runs	20	A	[0,2]
$\sigma$	0.9	R	[0,1]
$\beta$	0.1	Dataset dimension	d
K-folds cross-validation	10	Iterations	20
K-neighbors	5	Search agents (n)	5

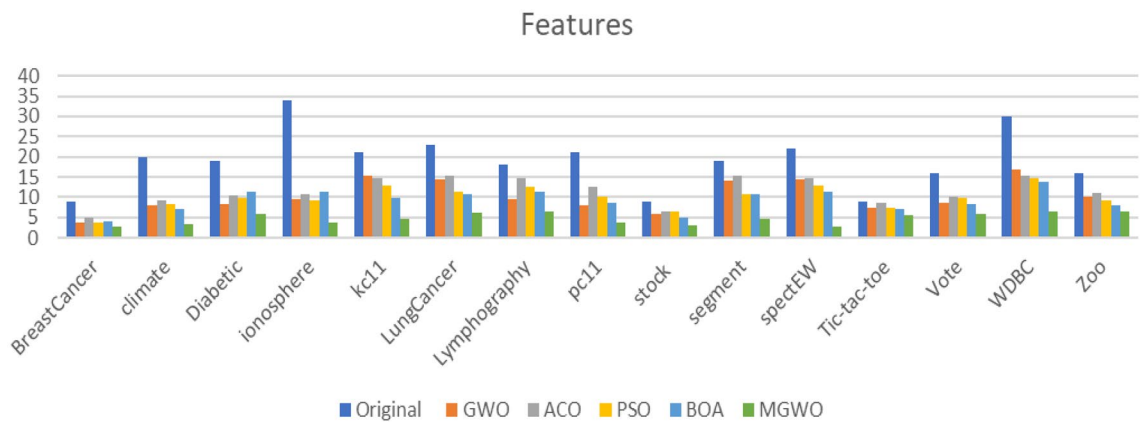
**Table 5.** Parameters settings.

Dataset	Original	GWO	ACO	PSO	BOA	MGWO
Breast cancer	9	3.7	4.8	3.8	3.9	<b>2.7</b>
Climate	20	7.8	9.2	8.4	7.1	<b>3.5</b>
Diabetic	19	8.2	10.5	9.7	11.2	<b>5.7</b>
Ionosphere	34	9.4	10.7	9.11	11.2	<b>3.8</b>
kc11	21	15.2	14.8	12.7	9.8	<b>4.6</b>
Lung cancer	23	14.5	15.2	11.2	10.7	<b>6.1</b>
Lymphographic	18	9.5	14.7	12.4	11.2	<b>6.4</b>
pc11	21	7.9	12.4	10.1	8.6	<b>3.8</b>
Stock	9	5.8	6.4	6.5	4.9	<b>3.1</b>
Segment	19	14.2	15.2	10.8	10.7	<b>4.7</b>
spectEW	22	14.5	14.7	12.8	11.4	<b>2.8</b>
Tic-tac-toe	9	7.4	8.5	7.4	6.9	<b>5.5</b>
Vote	16	8.7	10.1	9.7	8.2	<b>5.7</b>
WDBC	30	16.8	15.4	14.6	13.8	<b>6.4</b>
Zoo	16	10.2	10.9	9.1	7.8	<b>6.3</b>

**Table 6.** The features reduction for different algorithms. Significant values are in bold.

Dataset	Original	GWO	ACO	PSO	BOA	MGWO
Breast cancer	96	95.1	95.55	94.9	93.4	<b>96.2</b>
Climate	90.48	92.8	92.1	90.7	91.4	<b>96.4</b>
Diabetic	61.6	60.7	60.2	61.7	59.8	<b>69.7</b>
Ionosphere	85.1	87.5	86.4	86.5	84.12	<b>94.5</b>
kc11	81.4	80.4	80.7	79.9	82.4	<b>84.4</b>
Lung cancer	87.1	85.9	81.7	84.12	85.8	<b>88.2</b>
Lymphography	82.4	83.2	<b>84.5</b>	83.1	80.4	82.4
pc11	92.5	90.18	90.4	<b>93.5</b>	91.7	93.4
Stock	84	89.18	48.7	90.7	90.8	<b>94.13</b>
Segment	95	90.7	93.4	<b>95.4</b>	92.7	94.9
spectEW	79	79.5	78.4	80.14	81.7	<b>84.9</b>
Tic-tac-toe	77	77.8	75.2	78.4	79.1	<b>80.4</b>
Vote	92.5	94.15	91.5	93.4	92.7	<b>96.2</b>
WDBC	92.5	95.1	93.7	94.5	93.5	<b>95.8</b>
Zoo	89	93.4	90.2	92.9	92.7	<b>94.2</b>

**Table 7.** The classification accuracy for different algorithms. Significant values are in bold.

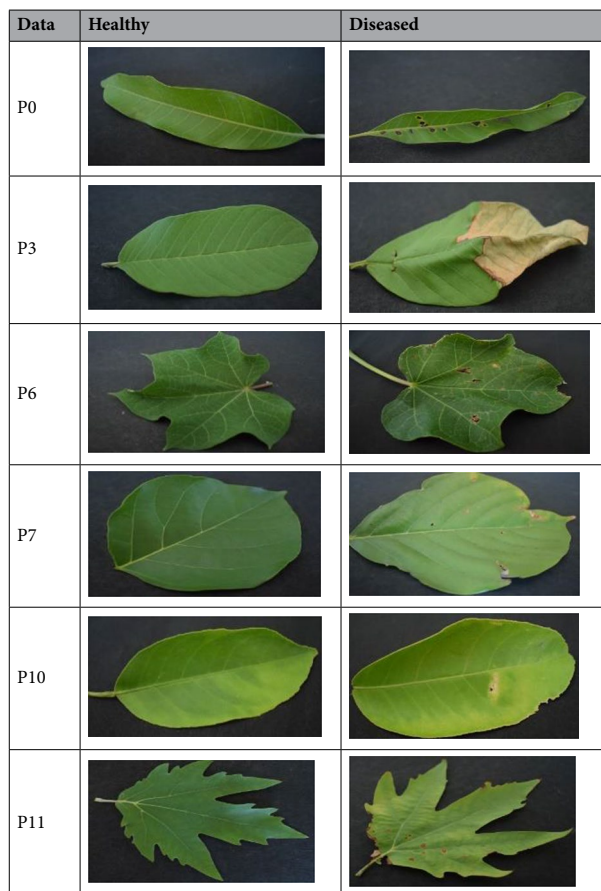


**Figure 9.** The features reduction for different algorithms.



**Figure 10.** The classification accuracy for different algorithm.

common types. Compared to conifers, deciduous trees have broader and bigger leaves. During the fall, their leaves fall off. This is due to the giant leaf, which allows for more photosynthesis to occur. Trees of this type are famous for their high wood production. There is a coniferous tree or evergreen tree green throughout the year.



**Table 8.** Sample of healthy and diseased leaf images of the plant's disease dataset.

ID	Code	Name	Full name	Name of the disease	No. of healthy images	No. of diseased images
1	P0	Mango	Mangifera indica	Anthracnose	170	265
2	P1	Arjun	Terminalia Arjuna	Leaf Spot	220	232
3	P2	Alstonia	Alstonia Scholaris	Foliar Galls	179	254
4	P3	Gauva	Psidium guajava	Fungal disease	277	142
5	P5	Jamun	Syzygium cumini	Fungal disease	279	345
6	P6	Jatropha	Jatropha curcas L	Leaf spot	133	124
7	P7	Sukh chain	Pongamia Pinnata	Chlorotic lesions	322	276
8	P9	Pomegranate	Punica granatum L	Cercospora spot	287	272
9	P10	Lemon	Citrus limon	Citrus Canker	159	77
10	P11	Chinar	Platanus orientalis	Leaf spot	103	120

**Table 9.** Plants diseases datasets description.

Leaves have a triangular form and grow upwards in most cases. Even though they have softer wood, they are pretty durable and resistant to various weather conditions<sup>60</sup>.

The data on <https://data.mendeley.com/datasets/hb74ynkjc/1> focuses on plants that contribute both ecologically and economically. As a result, ten different plants, such as Jamun, Lemon, Sukh Chain, Arjun, Pomegranate, Jatropha, Mango, Saptaparni, Guava, and Chinar, have been selected, as shown in Table 8. Images have been divided into two categories: healthy and diseased images. Table 9 shows the dataset description.

### Results

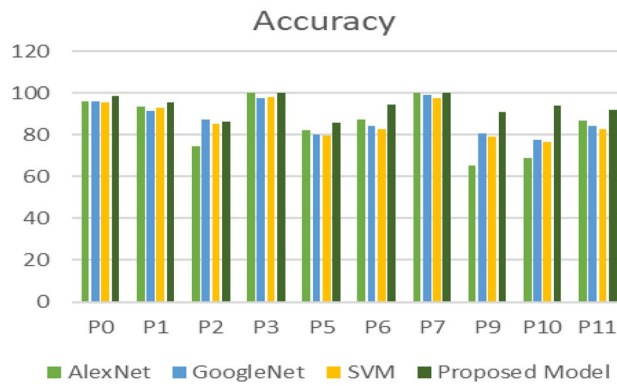
The proposed model (AlexNet as feature extraction, MGWO as a feature selection, and the SVM as a classifier) has achieved better results compared to Alexnet, GoogleNet, and the SVM. The results showed in Table 10 give a comparison among the AlexNet, GoogleNet, SVM, and the proposed model through different metrics such as



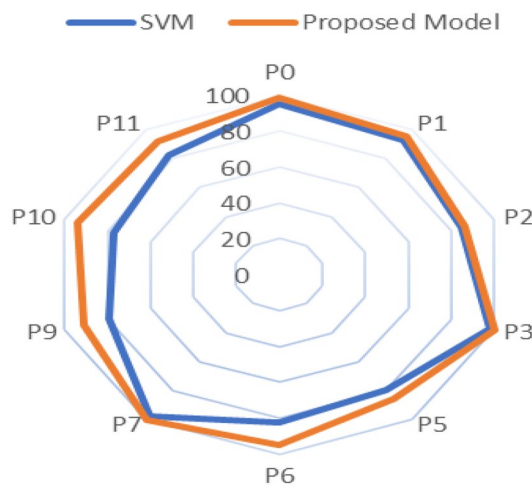
Dataset	Sensitivity	Specificity	Precision	F1-Score	Accuracy
AlexNet					
P0	100	90.6	94.2	97.01	96.3
P1	92.3	95	96	94.11	93.5
P2	97.7	45.7	68.9	80.81	74.4
P3	100	100	100	100	100
P5	93.8	68.5	78.2	85.29	82.4
P6	82.3	93.3	93.3	87.46	87.5
P7	100	100	100	100	100
P9	74.5	56.7	61.2	67.2	65.2
P10	66.7	70	52.6	58.82	68.9
P11	84	90	91.3	87.5	86.7
GoogleNet					
P0	95.45	97.08	97.22	96.33	96.24
P1	98.1	82.5	87.9	92.72	91.3
P2	90.7	82.9	86.7	88.65	87.2
P3	100	96.7	92.6	96.16	97.6
P5	81.3	78.9	76.5	78.83	80
P6	85	84	81	82.95	84.4
P7	100	98.3	98.2	99.09	99.1
P9	89.1	73.3	75.4	81.68	80.9
P10	60	86.7	69.2	64.27	77.8
P11	80	90	90.9	85.1	84.4
SVM					
P0	94.74	96.15	96.43	95.58	95.41
P1	92.11	93.75	94.59	93.33	92.86
P2	86.84	83.33	84.62	85.71	85.14
P3	98.04	98.77	99.01	98.52	98.36
P5	80.68	79.07	79.78	80.23	79.89
P6	82.46	82.57	83.19	82.82	82.51
P7	96.91	97.83	97.92	97.41	97.35
P9	80.34	78.26	78.99	79.66	79.31
P10	77.39	76.19	78.07	77.73	76.82
P11	82.91	82.57	83.62	83.26	82.74
Proposed model					
P0	97.2	100	100	98.6	98.8
P1	94.2	97.5	98	96.06	95.7
P2	88.7	84	86.2	87.43	86.5
P3	100	100	100	100	100
P5	89.5	81.5	85	87.19	85.8
P6	95.1	94.1	95.1	95.1	94.7
P7	100	100	100	100	100
P9	81	100	100	89.5	91
P10	94.4	93.8	94.4	94.4	94.1
P11	87.9	96.4	96.7	92.09	91.8

**Table 10.** Classification results for the four models.

sensitivity, specificity, precision, F1-score, and accuracy. The proposed model achieved the highest accuracy in all datasets except the dataset named p2 in which the GoogleNet achieved the best accuracy. Figure 11 display the comparison among the different model concerning the accuracy metric. A comparison between the SVM which trained for the extracted features directly without feature selection and the SVM which trained to the selected features by the MGWO that extracted by AlexNet showed in Fig. 12. The ROC curve on the test set for the proposed model SVM is introduced in Fig. 13.



**Figure 11.** Classification accuracy for the four models.



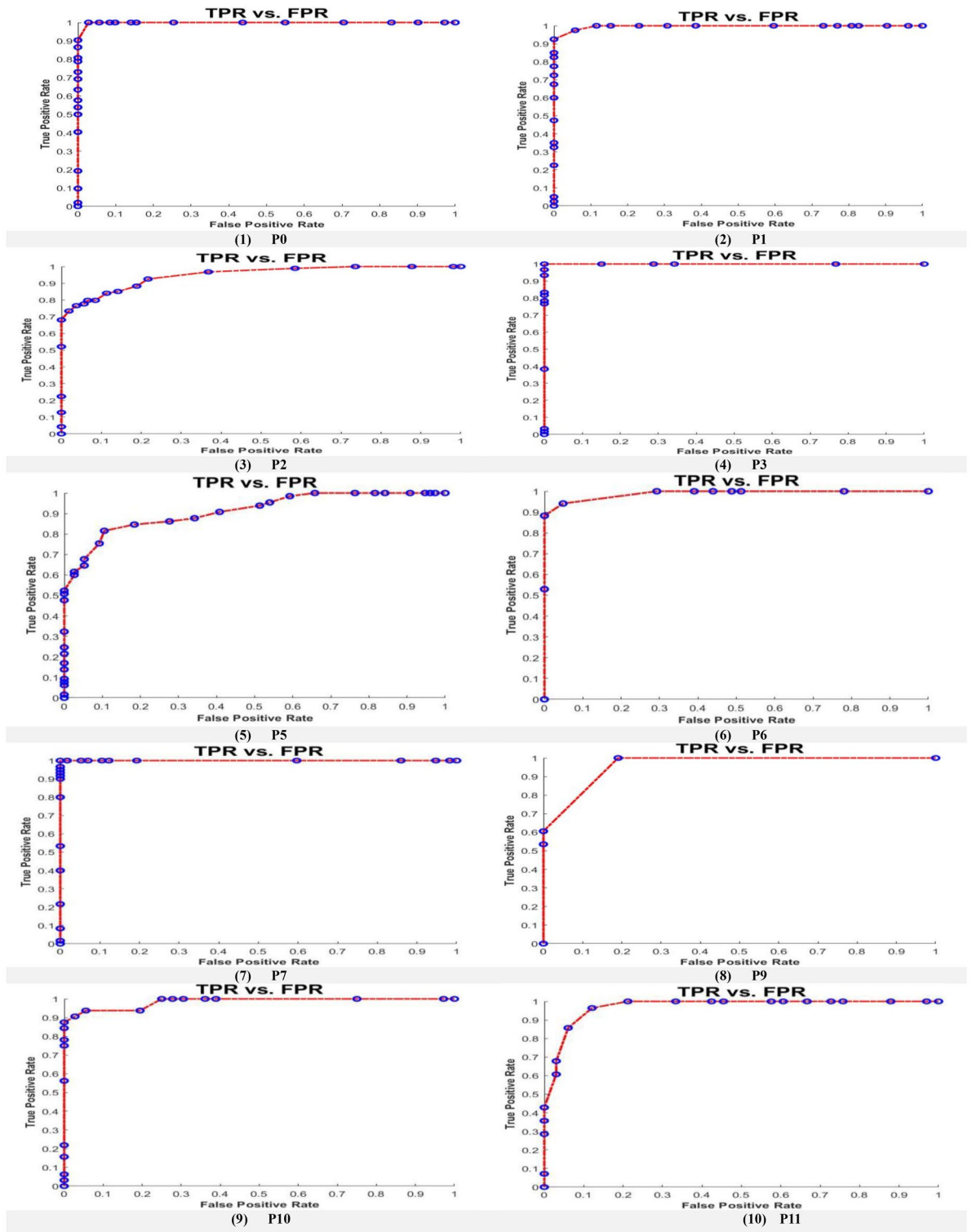
**Figure 12.** Classification accuracy of the standard SVM Vs the proposed model.

### Conclusion and future work

We present in this paper a paradigm for the identification of plant diseases. Initially, a comparison is undertaken using the SVM, AlexNet, and Google Net-based transfer training method, which will be used on the edge servers with increased computational capability, to detect plant diseases. Then, with the AlexNet feature extraction and support vector machines for plant detection and classification diseases, we proposed a hybrid approach based on the modified gray wolf optimization algorithm for eliminating the resulted features from the AlexNet.

The proposed model can operate on Internet of Things (IoT) devices that use a framework that integrates fog and cloud computing with limited resources. Experimental evidence shows that the suggested models can detect plant diseases accurately using the minimum computational resources from real-world datasets. The proposed model worked better on most data sets. In the future, using blockchain technology, we hope to improve the fog environment without impacting the efficiency of features map extraction.

We will also develop apps to detect plant diseases to support smart agriculture with deep learning support.



**Figure 13.** The ROC curve on the test set for the proposed model SVM.

### Data availability

The datasets analyzed for this study available in “<https://data.mendeley.com/datasets/hb74ynkjcjcn/1>” focus on plants that contribute both ecologically and economically. All datasets used are open access data, and we didn’t use any private data. Our research complies with institutional, national, and international guidelines and legislation. We have permissions from our institutional committee for scientific research ethics to do this study and to collect plants data from the open access dataset. Throughout the entirety of the process of generating all the plots and statistics, Microsoft Excel 2016 and MATLAB were utilized. On the other hand, figures were generated

using Microsoft PowerPoint and Adobe Photoshop. All plant samples used in the proposed model and tables are samples from the utilized dataset, which is open access data.

Received: 20 March 2022; Accepted: 24 September 2023

Published online: 09 November 2023

## References

- Liakos, K. G., Busato, P., Moshou, D., Pearson, S. & Bochtis, D. Machine learning in agriculture: A review. *Sensors* **18**(8), 2674 (2018).
- Xiao, L., Zhao, R. & Zhang, X. Crop cleaner production improvement potential under conservation agriculture in China: A meta-analysis. *J. Clean. Prod.* **269**, 122262 (2020).
- Lee, S. H., Chan, C. S., Mayo, S. J. & Remagnino, P. How deep learning extracts and learns leaf features for plant classification. *Pattern Recognit.* **71**, 1–13 (2017).
- Ale, L., Sheta, A., Li, L., Wang, Y., Zhang, N. Deep learning based plant disease detection for smart agriculture. In *2019 IEEE Globecom Workshops (GC Wkshps)*, 1–6 (2019).
- Liu, J. & Wang, X. Plant diseases and pests detection based on deep learning: A review. *Plant Methods* **17**(1), 22 (2021).
- Saleem, M. H., Potgieter, J. & Arif, K. M. Plant disease detection and classification by deep learning. *Plants* **8**(11), 468 (2019).
- 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).
- Mahum, R. *et al.* A novel framework for potato leaf disease detection using an efficient deep learning model. *Hum. Ecol. Risk Assess. Int. J.* **29**(2), 303–326 (2023).
- Gouse, S., Dulhare, U. N. Automation of Rice Leaf Diseases Prediction Using Deep Learning Hybrid Model VVIR. In *Advancements in Smart Computing and Information Security: First International Conference, ASCIS 2022, Rajkot, India, November 24–26, 2022, Revised Selected Papers, Part I*, 133–143 (2023).
- Sarhan, A. Fog computing as solution for IoT-based agricultural applications. In *Smart Agricultural Services Using Deep Learning, Big Data, and IoT*, 46–68 (IGI Global, 2021).
- Abbas, A., Jain, S., Gour, M. & Vankudothu, S. Tomato plant disease detection using transfer learning with C-GAN synthetic images. *Comput. Electron. Agric.* **187**, 106279 (2021).
- Thenmozhi, K. & Reddy, U. S. Crop pest classification based on deep convolutional neural network and transfer learning. *Comput. Electron. Agric.* **164**, 104906 (2019).
- Wiesner-Hanks, T. *et al.* Millimeter-level plant disease detection from aerial photographs via deep learning and crowdsourced data. *Front. Plant Sci.* **10**, 1550 (2019).
- Too, E. C., Yujian, L., Njuki, S. & Yingchun, L. A comparative study of fine-tuning deep learning models for plant disease identification. *Comput. Electron. Agric.* **161**, 272–279 (2019).
- Chen, T. *et al.* Intelligent identification system of disease and insect pests based on deep learning. *China Plant Prot.* **39**(04), 26–34 (2019).
- Zhou, G., Zhang, W., Chen, A., He, M. & Ma, X. Rapid detection of rice disease based on FCM-KM and faster R-CNN fusion. *IEEE Access* **7**, 143190–143206 (2019).
- 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).
- Rahman, C. R. *et al.* Identification and recognition of rice diseases and pests using convolutional neural networks. *Biosyst. Eng.* **194**, 112–120 (2020).
- Guo, Y. *et al.* Plant disease identification based on deep learning algorithm in smart farming. *Discret. Dyn. Nat. Soc.* **2020**, 1–11 (2020).
- Atila, Ü., Uçar, M., Akyol, K. & Uçar, E. Plant leaf disease classification using EfficientNet deep learning model. *Ecol. Inform.* **61**, 101182 (2021).
- Gadekallu, T. R. *et al.* A novel PCA—whale optimization—based deep neural network model for classification of tomato plant diseases using GPU. *J. Real-Time Image Process.* **18**, 1383–1396 (2021).
- Sanga, S., Mero, V., Machuve, D., Mwanganda, D. Mobile-based deep learning models for banana diseases detection. *arXiv Prepr. arXiv:2004.03718* (2020).
- Chohan, M., Khan, A., Chohan, R., Hassan, S. & Mahar, M. Plant disease detection using deep learning. *Int. J. Recent Technol. Eng.* **9**(1), 909–914 (2020).
- Guo, X. Q., Fan, T. J. & Shu, X. Tomato leaf diseases recognition based on improved multi-scale AlexNet. *Trans. Chin. Soc. Agric. Eng.* **35**(13), 162–169 (2019).
- Tan, L., Lu, J. & Jiang, H. Tomato leaf diseases classification based on leaf images: A comparison between classical machine learning and deep learning methods. *AgriEngineering* **3**(3), 542–558 (2021).
- Agarwal, M., Singh, A., Arjaria, S., Sinha, A. & Gupta, S. ToLeD: Tomato leaf disease detection using convolution neural network. *Proc. Comput. Sci.* **167**, 293–301 (2020).
- Kundu, N. *et al.* IoT and interpretable machine learning based framework for disease prediction in pearl millet. *Sensors* **21**(16), 5386 (2021).
- Zhang, S., Zhang, S., Zhang, C., Wang, X. & Shi, Y. Cucumber leaf disease identification with global pooling dilated convolutional neural network. *Comput. Electron. Agric.* **162**, 422–430 (2019).
- Khamparia, A. *et al.* Seasonal crops disease prediction and classification using deep convolutional encoder network. *Circuits Syst. Signal Process.* **39**(2), 818–836 (2020).
- Bedi, P. & Gole, P. Plant disease detection using hybrid model based on convolutional autoencoder and convolutional neural network. *Artif. Intell. Agric.* **5**, 90–101 (2021).
- Faggella, D. AI in agriculture—present applications and impact. *Emerj Artif. Intell. Res. Insight.* **18**, 2020 (2020).
- Boulent, J., Foucher, S., Théau, J. & St-Charles, P.-L. Convolutional neural networks for the automatic identification of plant diseases. *Front. Plant Sci.* **10**, 941 (2019).
- Samuel, A. L. Some studies in machine learning using the game of checkers. *IBM J. Res. Dev.* **3**(3), 210–229 (1959).
- Kujawa, S., Mazurkiewicz, J. & Czekala, W. Using convolutional neural networks to classify the maturity of compost based on sewage sludge and rapeseed straw. *J. Clean. Prod.* **258**, 120814 (2020).
- Saleem, M. H., Potgieter, J. & Arif, K. M. Automation in agriculture by machine and deep learning techniques: A review of recent developments. *Precis. Agric.* **22**, 1–39 (2021).
- Weiss, K., Khoshgoftaar, T. M. & Wang, D. A survey of transfer learning. *J. Big Data* **3**(1), 1–40 (2016).
- Krizhevsky, A., Sutskever, I. & Hinton, G. E. Imagenet classification with deep convolutional neural networks. *Adv. Neural Inf. Process. Syst.* **25**, 1097–1105 (2012).
- Hemmer, M., Van Khang, H., Robbersmyr, K. G., Waag, T. I. & Meyer, T. J. J. Fault classification of axial and radial roller bearings using transfer learning through a pretrained convolutional neural network. *Designs* **2**(4), 56 (2018).

39. Dhillon, A. & Verma, G. K. Convolutional neural network: A review of models, methodologies and applications to object detection. *Prog. Artif. Intell.* **9**(2), 85–112 (2020).
40. Özyurt, F. Efficient deep feature selection for remote sensing image recognition with fused deep learning architectures. *J. Supercomput.* **76**, 1–19 (2019).
41. Vapnik, V. *The Nature of Statistical Learning Theory* (Springer Science & Business Media, 2013).
42. Soman, K. P., Loganathan, R. & Ajay, V. *Machine Learning with SVM and Other Kernel Methods* (PHI Learning Pvt. Ltd., 2009).
43. Pedersen, R., Schoeberl, M. An embedded support vector machine. In *2006 International Workshop on Intelligent Solutions in Embedded Systems*, 1–11 (2006).
44. Altuntacs, Y. & Kocamaz, F. Deep feature extraction for detection of tomato plant diseases and pests based on leaf images. *Celal Bayar Univ. J. Sci.* **17**(2), 145–157 (2021).
45. Bishop, C. M. Pattern recognition. *Mach. Learn.* **128**(9) (2006).
46. Mirjalili, S., Mirjalili, S. M. & Lewis, A. Grey wolf optimizer. *Adv. Eng. Softw.* **69**, 46–61 (2014).
47. Dereli, S. A new modified grey wolf optimization algorithm proposal for a fundamental engineering problem in robotics. *Neural Comput. Appl.* **33**, 1–13 (2021).
48. Tsipis, A. *et al.* An alertness-adjustable cloud/fog IoT solution for timely environmental monitoring based on wildfire risk forecasting. *Energies* **13**(14), 3693 (2020).
49. Guardo, E., Di Stefano, A., La Corte, A., Sapienza, M. & Scatà, M. A fog computing-based iot framework for precision agriculture. *J. Internet Technol.* **19**(5), 1401–1411 (2018).
50. Goundar, S., Bhushan, S. B. & Rayani, P. K. *Architecture and Security Issues in Fog Computing Applications* (IGI Global, 2019).
51. Gómez-Chabla, R., Real-Avilés, K., Morán, C., Grijalva, P., Recalde, T. IoT applications in agriculture: A systematic literature review. In *2nd International Conference on ICTs in Agronomy and Environment*, 68–76 (2019).
52. Srinivasan, G., Vishnu Kumar, N., Shafeer Ahamed, Y. & Jagadeesan, S. Providing smart agricultural solution to farmers for better yielding using IoT. *Int. J. Adv. Sci. Eng. Res.* **2**(1), 2017 (2017).
53. Gao, Z.-M. & Zhao, J. An improved grey wolf optimization algorithm with variable weights. *Comput. Intell. Neurosci.* **2019**, 1–18 (2019).
54. What is Confusion Matrix and Advanced Classification Metrics? *Data Science and Machine Learning-blogger*. manisha-sirsat.blogspot.com (2019).
55. Dheeru, E. D., Taniskidou, K. {UCI} Machine Learning Repository. (2017).
56. Dwivedi, S., Vardhan, M. & Tripathi, S. An effect of chaos grasshopper optimization algorithm for protection of network infrastructure. *Comput. Netw.* **176**, 107251 (2020).
57. Abdel-Basset, M., El-Shahat, D., El-henawy, I., de Albuquerque, V. H. C. & Mirjalili, S. A new fusion of grey wolf optimizer algorithm with a two-phase mutation for feature selection. *Expert Syst. Appl.* **139**, 112824 (2020).
58. Gou, J. *et al.* A generalized mean distance-based k-nearest neighbor classifier. *Expert Syst. Appl.* **115**, 356–372 (2019).
59. El-Hasnony, I. M., Elhoseny, M. & Tarek, Z. A hybrid feature selection model based on butterfly optimization algorithm: COVID-19 as a case study. *Expert Syst.* **39**, e12786 (2022).
60. Chouhan, S. S., Singh, U. P., Kaul, A., Jain, S. A data repository of leaf images: Practice towards plant conservation with plant pathology. In *2019 4th International Conference on Information Systems and Computer Networks (ISCON)*, 700–707 (2019).

## Author contributions

All authors have equal contributions.

## Competing interests

The authors declare no competing interests.

## Additional information

**Correspondence** and requests for materials should be addressed to I.M.E.-H.

**Reprints and permissions information** is available at [www.nature.com/reprints](http://www.nature.com/reprints).

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



**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

© The Author(s) 2023