ACADEMY JOURNAL OF SCIENCE AND ENGINEERING
Available online @ www.academyjsekad.edu.ng
AJSE 18 (2) 2024
Original Research

COMPARATIVE ANALYSIS OF STANDARD AND ENHANCED SWARM INTELLIGENT DEEP BELIEF NETWORK ON MAIZE DISEASE DETECTION

Adebisi Oluwatosin[1], Kehinde Sotonwa[1], Adenibuyan Micheal[2], Adesesan Adeyemo[3]

[1]Department Computer Science, Lagos State University, Ojo, Nigeria
[1,2]Department Computer Science &Infor. Tech., Bells University of Technology Ota, Nigeria
[3]University of Ibadan, Nigeria
Corresponding Authors E-mail: kehinde.sotonwa@lasu.edu.ng, aeoluwatosin@bellsuniversity.edu.ng,

Abstract

Deep Belief Network (DBN) and some other Artificial Neural Network (ANN) have been used for the detection and classification of diseases in plant but have been known for over-fitting problem. This has often times affected the accuracy of the system with high false positive rate. Particle Swarm Optimization (PSO) technique, has been used to enhance some deep learning techniques but still suffers stagnation in local optima and high computational cost mainly due to the large search space. Hence, there is need for improvement to develop a more efficient model for plant diseases detection and classification using Deep Belief Network with Improved Particle Swarm Optimization (DBN-IPSO). Images of healthy and unhealthy 3852 maize plant samples were acquired from https:\\www.kaggle.com. The acquired data were pre-processed, leaf colour images were converted to grayscale, cropped and contrast-enhanced with local histogram equalization. IPSO was formulated by incorporating modified acceleration coefficient and velocity component into the standard PSO to avoid premature convergence and imbalance between exploration and exploitation stages. The PSO was applied to DBN to select its optimal weight parameters. The system was implemented using a language app designer, trained and tested using k-fold cross-validation method, where k is 10-fold. The performance of the developed DBN-IPSO technique was compared with existing DBN-PSO and DBN using False Positive Rate (FPR), sensitivity, specificity, overall accuracy and computation time. The result shows that the developed technique improves the accuracy and efficiency of the system, making it a promising approach for automated plant disease diagnosis and surveillance in agriculture.

Keywords:
Modified acceleration coefficient, Deep belief network, Particle swarm optimization, Premature convergence, Local optima stagnation.

1. INTRODUCTION

Agriculture adds to around 45% of the GDP of developing countries, with over 75% of their populace depending on agriculture for employment (Awuor *et al.*, 2013a; Awuor *et al.*, 2013b). One of the most significant life forms on earth is the plant. The ratio of oxygen and carbon dioxide in the earth's atmosphere is controlled by plants. The relationship between humans and plants is also very strong. Additionally, there is an intimate relationship between people and plants. Unfortunately, the incredible growth of human civilizations has disturbed this stability to a greater degree than we realized. We all have a responsibility to help protect the plants from danger, whether it is through pushing them out of the way, or even just being there when they need saving. So, the plant community should be restored to its original state and everything should be back to its original balance, (Zhang *et al.*, 2012; Ali et al., 2023; Sushma *et al.*, 2024; Kumar, 2021). The Food and Agriculture Organization (FAO) estimates that pests and diseases cause the loss of 20–40% of worldwide food production, constituting a threat to food security (Bruinsma, 2017; Anitha and Saranya, 2022; Singh and Mukherjee, 2024; Alpyssov, 2023).

Plant identification can be helped by digitization of the image and a plant can be identified from knowledge base through experts, images or illustrations. Plants can be recognized by their flowers, leaves, root and fruits. However, plants may have similar features that may make them hard to differentiate or even identify. The leaves of a plant are considered useful in identification of its species (Malik *et al.*, 2022; Tiwari & Gupta, 2023). Other plant features are also used, ranging from very basic traits like leaf color and shape to extremely complicated ones like tissue and cell structure.

Due to their integration of digital cameras, modern mobile phones may produce images of an extremely high quality (Baihaqie & Wulan, 2021; Xu *et al.*, 2020) . The nature of plant leaves is two dimensional, they possess significant traits that are helpful in classifying plant species. There are abundant examples of relevant numerical properties that should be extracted from the object of interest. This is done in order to classify plants exist everywhere and carry significant life sustaining information for human and environmental growth and development. There are different types of plant species as well as their features. Identifying these different plants requires extensive knowledge base of the plants and the terminology used in the field, with the help of botanists and experts, and can be facilitated with the aid of Information Communication Technology employed in Agriculture (Almodfer *et al.*, 2023; Fan *et al.*, 2020; Zenat *et al.*, 2024).

Several methods have been used to identify plants. One of the most popular and best methods to ensure plat disease identification is the use of determination keys. But its use has become cumbersome and tedious in view of a large number of species to classify. Nomenclature, denominations and use of technical terms does not make the task easier. The second most popular method is an interactive determination through the use of forms on websites that can somehow be viewed as catalogues. It consists of choosing options that will describe the plant in terms of shape or organization of the leaves. The third most popular are the tools to aid in the recognition from images. This method provides a web or mobile platform that integrates an automatic plant classification system from images (Nilsson *et al.*, 2019; Chen *et al.*, 2020; Fan *et al.*, 2021; Roy *et al.*, 2023; Sushma *et al.*, 2024).

The field of computer vision is concerned with the theory underpinning artificial systems that extract information from images. With the use of complex architectures that incorporate several non-linear transformations, a collection of learning approaches known as deep learning attempt to model data. The research on artificial neural networks and multilayer perceptron, which

have additional hidden layers in a deep learning framework, is where the idea of deep learning originated. The elementary bricks of deep learning are the neural networks that are combined to form the deep neural networks, (Ciresan *et al.,* 2011).

Face and speech recognition, computer vision, automated language processing, and text categorization (for example, spam recognition) have all advanced significantly owing to these advances in all aspects of sound and image processing. Before the invention of deep learning for computer vision, learning was based on the extraction of key variables, or features, but these techniques necessitate a high level of expertise for image processing (Huang *et al.,* 2023; Salman *et al.,* 2024).

Restricted Boltzmann machines (RBM) are stochastic and generative neural networks capable of learning internal representations and are able to describe and (given enough time) solve challenging combinatorial problems. It has a two-layered structure, where the visible layer is a set of patterns that can be seen, and the hidden layer is a randomly generated neural network that is used to learn the probability distribution of the input data. RBM is a powerful tool for describing data, has a feature representation and dimensionality reduction technique, and is frequently used for building DBN. RBMs are the building blocks of deep architectures such as DBN. PSO is a computational method that optimize a problem by iteratively trying to improve a solution with regard to a given measure of quality. A DBN fine-tuned by improved PSO was explored in this research to automatically detect and classified plant diseases (Dewi *et al.,* 2020; Wang *et al.,* 2019; Xing *et al.,* 2020)

2. LITERATURE REVIEW

Some challenges of plant disease detection and classification techniques used by previous proposed methods are not able to discover many diseases in one image or multiple cases of the same disease in one image, that is, multi-class classification. Full clarification of background carried out on plant classification and detection by researchers:

Mohanty *et al.,* (2016) proposed the used of image-based plant disease detection using deep learning with AlexNet and GoogLeNet CNN, the study obtained a 99.35% accuracy on an extended test set but it will be necessary to use a more varied collection of training data to increase the accuracy, the work was constrained to the classification of single leaves and computational complexity was not considered. Sladojevic *et al.,* (2016) proposed a deep neural networks-based recognition of plant diseases by leaf image classification, the technique used was deep convolutional networks. Caffe, a deep learning framework developed by Berkley vision and learning centre was used to perform the deep CNN training. It achieved precision between 91% and 98%, for separate class tests, on average 96.3% but the computational complexity of CNNs during the training phase was very high due to the millions of parameters used in the network.

Ha *et al.,* (2017) proposed a Fusarium wilt of radish classification using deep convolutional neural network from unmanned aerial systems. It makes use of VGG-A, the accuracy of CNN's Fusarium wilt of radish detection was 93.3%. Additionally, it achieved 82.9% accuracy, outperforming the industry-standard machine learning technique, but only RGB images were considered. Radish with severe Fusarium wilt was frequently overlooked. The degree of Fusarium wilt of radish was not taken into account, although the segmentation methods used had an impact on detection accuracy.

Zhang *et al.,* (2018) carried out a study on can deep learning identifies tomato leaf disease? The paper applied transfer learning to detect a deep convolutional neural network (CNN) is used to analyze tomato leaf disease. CNN's backbone was made up of AlexNet, GoogleLeNet, and ResNet.

The CA by ResNet gave the best value of 97.28%. But the work was not extended to other plant leaf disease identification problems. Kerkech *et al.,* (2018) proposed the study for the purpose of detecting vine diseases in UAV images, a deep leaning method using colorimetric spaces and vegetation indices is used. LeNet architecture as a convolutional neural network was used in this study. The CA of 95.8% was achieved. There were no vineyard plots or samples of novel grape diseases in the UAV multispectral images database.

Brahimi *et al.,* (2019) carried out a study on deep interpretable architecture for plant carried out a study on deep interpretable architecture for plant diseases classification. CNN architecture was used. The computational cost of this architecture is a little bit high.

Arsenovic *et al.,* (2019)) presented a study that addressed the current drawbacks of deep learning-based techniques for plant disease identification using CNN model.93.67% accuracy was attained by the trained model. However, there was no thorough analysis of many variables of the CNN architecture.

Adekunle, (2020) proposed a study on implementation of improved machine learning techniques for plant disease detection and classification. The study uses random forests and support vector machine (SVM). The experimental results show.

Jasim & Al-Tuwaijari, (2020) proposed a study of employing deep learning and image processing approaches, identify plant leaf diseases. Accuracy of (98.29%) and (98.029%), respectively, was attained using a convolutional neural network (CNN) during training and testing. No in-depth analysis of the CNN architecture's learning parameters. The number of leave types used was on the low side and metrics such as sensitivity, precision and computational time were not taken into consideration. **1.0**

In order to categorized the diseases of coffee and apple plants by extracting semantic data, a CNN model was presented (Fan *et al.,* (2022) to utilizing histogram of gradient and deep learning algorithms independently, separate the various characteristics into parallel extractions that are then combined to separate the regional geographical pattern information. In order to extract the local features, it first normalized before combining all of the features, gamma-correct the image created, it estimated each cell's individual pixel's gradient and amplitude. Obtained detection accuracy regarding the apple and coffee datasets was 99.79% and 97.12%, respectively. This research could be expanded upon by using a bigger collection of pictures taken in real time and analyzing cutting-edge augmentation techniques.

Syed-Ab-Rahman *et al.,* (2022) proposed citrus plant disease diagnosis by the use of deep learning using an equalization method that controls all of the images using a histogram uses numerous data augmentation techniques to combine all intensity ranges into one to increase the sample size, pre-process the dataset. In order to use ResNet101 for feature extraction, the anchor-based RPN technique is employed to train the network from beginning to end which determine the ROI, and to create region suggestions. The average detection accuracy attained by the authors was 94.37%. The system is computationally expensive due to the two-stage approach.

Sushma *et al.,* (2024) outlines a thorough strategy for identifying and categorizing plant leaf diseases by utilizing advanced image processing methods alongside CNNs. Through extensive experimentation and validation, the research demonstrates that the proposed system not only excels in accurately detecting and categorizing diseases but also provides actionable recommendations for agricultural practices to mitigate disease risks and improve overall crop health

3. MATERIALS AND METHODS

In this study, a model for plant disease recognition and detection was developed using a dataset comprising various plant disease images

captured under diverse conditions. Pre-processing steps including noise removal, grayscale conversion, cropping, and contrast enhancement were applied. IPSO technique was devised, modifying standard PSO parameters to enhance exploration and exploitation balance. IPSO was then employed to optimize weight parameters in DBN. Features were extracted using the DBN-IPSO application, facilitating the detection and classification of plant diseases. This integrated approach offers a robust framework for automated plant disease diagnosis, leveraging advanced optimization algorithms and deep learning methodologies.

3.1 Acquisition of Plant Disease Dataset

A wide-ranging maize dataset of different diseases which contain images were acquired in an online public database. Maize images of 3852 disease samples consisting of 1192 sample of maize common rust, 513 samples of maize gray cercospora leaf and 985 samples of maize northern leaf blight and 1162 of healthy and 2690 unhealthy samples were downloaded from online dataset. The maize plant images were downsized without changing the images into an adequate pixel of 200 by 200. All images were trained and tested using k-fold method, where k is 10-fold.

3.2 Formulation of IPSO

There are some parameters in standard PSO algorithm that affect its performance. The basic PSO parameters are number of iterations, velocity components, and acceleration coefficients. In addition, PSO is also influenced by inertia weight, velocity clamping, and velocity constriction. Some of the deficiencies of PSO are premature convergence, high computational complexity, slow convergence, and sensitivity to parameters. Reasons for such challenges PSO typically quickly converges to a local minimum because it is inadequate to manage the interaction between exploitation (local search) and exploration global search (Liang *et al.*, 2015; Zhang *et al.*, 2015). An enhanced PSO from standard PSO selected the best combination of weights at the convolution layers and classification layer of the DBN. The optimal weights selected improved the computational time and accuracy of DBN network during training and classification process and as well overcome the problem of overfitting. However, PSO parameters such as acceleration coefficient and velocity component were modified to produce IPSO which was believed to take care of the interaction between exploitation (local search) and exploration (global search), to avoid being trapped at local minimum quickly and achieving optimal weights.

3.2.1 Formulation of Objectives Function for Assignment of Optimal Weight

An optimal weight determination problem was generally formulated as follows:

$$\min_{Pbest,Gbest,W_f^d} \emptyset(y(W_f^d)) \quad (1)$$

Subject to: C1: $0 \leq (wt, Obj, Pbest, Gbest, W^d) \leq 1\ W^d \in W_e$ (2)

C2: $Obj = \begin{cases} 1 & if Obj \leq \overline{Obj} \\ 0 & otherwise Obj > \overline{Obj} \end{cases}$ (3)

where $wt \in R^n$ is the vectors of randomized weights $\overline{Obj}$ represents the mean square error for Obj. State vector in its full form is represented by $y = [wt]$, where wt contains the collection of weights of DBN. The issues were listed on the weight's horizon $W_e = [W_o^d W_f^d]$. Where W_e made up of original weights of W_o^d of y and chosen final weight W_f^d which was equivalent to optimal weight.

3.2.2 Formation of Acceleration Coefficient and Velocity Component in Standard PSO

The standard PSO has inertia weight ω and the goal of inertia weight was to eliminate the premature convergence but could not eliminate the effect.

$$\omega = \omega_{min} + (\omega_{max} - iter + 1) \times \frac{\omega_{max} - \omega_{min}}{Max_{iter}} \quad (4)$$

where$iter$ is the current iteration, ω_{max} is the final weight, $\boldsymbol{\omega_{min}}$ is the initial weight $\boldsymbol{\omega}$ is the inertia weight employed to overcome the problem of premature convergence**,** $Max_{iter}$is the maximum number of iterations. For this study, modified acceleration coefficientc_1^t, $c_2^t$and constriction factors was introduced to the standard PSO as against the inertial coefficient ω. This is to avoid particle divergence while looking for solutions in the problem space.

$$c_1^t = c_1 - \frac{t}{Max_{iter}}(c_1) \quad (5)$$

$$c_2^t = c_2 - \frac{t}{Max_{iter}}(c_2) \quad (6)$$

Where c_1 is the initial cognitive acceleration coefficient, c_2 is the initial social acceleration coefficient, c_1^t is the modified cognitive acceleration coefficient in the current generation, c_2^t is the modified social acceleration coefficient in the current generation, Max_{iter} is the maximum number of iterations, and t is the current generation.

The constriction factor (γ) was applied to standard PSO to overcome the problem of premature convergence: $\gamma = \frac{2}{(\rho-2)+\sqrt{\rho^2-4\rho)}}$ (7)

where ρ=c_1+$c_2$based on factors such as social and cognitive elements, it is conditioned toρ>4 which ensures the efficiency of the constriction coefficient.

Modified Velocity Component was introduced to reduce the blind spots of the standard PSO by clamping the particle velocity such that it stays within the bounds of (V_{std}, V_{mean}). Using the following expression, the V_{std} andV_{mean} parameters are derived.

$$V_{std,D} = \delta(x_{std,D} - x_{mean,D}) \quad (8)$$

$$V_{mean,D} = \delta(x_{mean,D} - x_{std,D}) \quad (9)$$

where$V_{std,D}$ is the permitted standard deviation velocity of particles, D is the velocity clamping factor, and $x_{std,D}$ and $x_{mean,D}$ are the standard deviation and mean location values of particles at D^{th} dimension. Therefore, the standard deviation velocity limits the velocities in the following way if the velocity update generates a step that is too large:

$$V_{i,D}(t+1) = \begin{cases} V_{std,D}, & if V_{i,D}(t) > V_{std,D} \\ V_{i,D}(t+1), & if V_{i,D}(t) < V_{std,D} \end{cases} \quad (10)$$

The velocity and position of IPSO were given as follows:

$${V_{i,D}}^{t+1} = \gamma.[{V_{i,D}}^{t} + c_1^t \ .r_1\left[P_{best,iD}^t - {x_i}^t\right] + c_2^t.r_2\left[G_{best,D}^t - {x_i}^t\right]] \quad (11)$$

The velocity clamping factor obviously regulates the convergence pace as opposed to acceleration constants, which balance the local and global search. The convergence of the PSO algorithm is essentially determined by the acceleration constants and velocity clamping factor. The velocity and position of IPSO were given:

$${x_i}^{t+1} = {x_i}^t + {V_{i,D}}^{t+1} \quad (12)$$

Where ${V_{i,D}}^{t+1}$ is modified velocity of particle, ${V_{i,D}}^t$ is the current velocity of particle in the D^{th} dimension with (V_{std}, V_{mean}), ${x_i}^{t+1}$ is the modified position of particle, ${x_i}^t$ is the current position of individual particle, γ is inertial weight parameter, c_1^t and c_2^t is the cognitive and social acceleration factor in the current generation, r_1 and r_2 are uniform random number between [0,1], $\boldsymbol{P_{best,iD}^t}$ is best position of individual i in the D^{th} dimension with (V_{std}, V_{mean}) until iteration t, and $\boldsymbol{G_{best,D}^t}$ is the global best position of the group in the D^{th} dimension with (V_{std}, V_{mean}) until iteration t.

Equations (10) and (11) were modified to reflect the acceleration coefficients or factors, which are positive quantities that measure cognitive and social learning. The best DBN weight is finally returned once the number of iterations can no longer be increased due to the global best position (G_{best}). The IPSO technique's computational phases for achieving optimized DBN weight is described in Algorithm Listing below:

Step 1: Select the quantity of particles

Step 2: Initialize population with coordinates$\boldsymbol{x_{ij}^0}$ and velocities $\boldsymbol{v_{ij}^0}$, with cognitive and social parameter$c$1and$c$2, set to the maximum number

of dimensions (D=max) and particles (n=max), respectively. Max_{iter} = maximum no of iteration

Step 3: Set iteration t = 1

Step 4: Set the particle's objective function values as $f(x) = f(x_i^t)$

$$f(x) = \sum_{i=1}^{m}\sum_{j=1}^{n} \Delta(W_{i,j}^{m,n})\left(-(x_j)\right)$$

Where x_i^t represent the s at i=1,2, …, n and k=2,3, …, m, Where $\Delta(W_{i,j}^{m,n})((x_i) - (x_j)$ is the change in weight of input pixel x along the row and column.

Step 5: Find each particle's optimal cognitive position as

$P_{best,iD}^t = x_i^t$ and global best as $G_{best,iD} = min\{P_{best,iD}^t\}$

Step 6: Calculate constriction coefficient $(\gamma) = \frac{2}{(\rho-2)+\sqrt{\rho^2-4\rho)}}$

where ρ=c1+c2 based on the condition, the cognitive, and the social featuresρ>4 assured the efficiency of the constriction coefficient

$c_1^t = c_1 - \frac{t}{Max_{iter}}(c_1)$ $c_2^t = c_2 - \frac{t}{Max_{iter}}(c_2)$

Step 7: Find the new values by updating the velocity and position of particles of the i[th] particle and D[th] dimension

$V_{std,D} = \delta(x_{std,D} - x_{mean,D})$

$V_{mean,D} = \delta(x_{mean,D} - x_{std,D})$

where$V_{std,D}$ is the acceptable standard deviation particle velocity, D is the velocity clamping factor, and $x_{std,D}$and $x_{mean,D}$ are the standard deviation and mean values of particle locations in the D[th] dimension.

$V_{std,D} = \delta(x_{std,D} - x_{mean,D})$

where $V_{std,D}$ is the acceptable standard deviation particle velocity, D is the velocity clamping factor, and $x_{max,D}$ is the standard deviation location values of particles at D[th] dimension.$V_{i,D}(t+1) =$

$$\begin{cases} V_{std,D}, & if V_{i,D}(t) > V_{std,D} \\ V_{i,D}(t+1), & if V_{i,D}(t) < V_{std,D} \end{cases}$$

$V_{i,D}{}^{t+1} = \gamma . V_{i,D}{}^{t} + c_1^t \ . r_1\left[P_{best,iD}^t - x_i{}^t\right] + c_2^t . r_2\left[G_{best,iD}^t - x_i{}^t\right]$

$x_i{}^{t+1} = x_i{}^t + V_{i,D}{}^{t+1}$ where the two random numbers,$r_1$and $r_2$are generated by a uniform distribution $U(0,1)$

Step 8: Find the values of the objective function of:

$x_b^{t+1} as f(x) = f(x_b^{t+1})$ and find the best particle's index b

Step 9: Update $Pbest$ of population

$$P_{best,bD}^{t+1} = \begin{cases} P_{best,iD}^t & iff(x_b^{t+1}) > P_{best,iD}^t \\ x_b^{t+1} & iff(x_b^{t+1}) \le P_{best,iD}^t \end{cases}$$

Step 10: Update $Gbest$ of population

$G_{best,bD} =$

$$\begin{cases} \min\ (P_{best,iD}^t) & iff(x_b^{t+1}) > P_{best,iD}^t \\ \min\ (P_{best,bD}^{t+1}) & iff(x_b^{t+1}) \le P_{best,iD}^t \end{cases}$$

Step 11: If t< Max_{iter} then t= $t+1$and GOTO step 1 else GOTO step l2

Step 12: Output optimum weight selected solution as:

$Gbest_{bD}.Gbest_{bD} = x_b$

3.2.3 Formation of Acceleration Coefficient and Velocity Component in Standard PSO

In order to recognize plant leaves, important features are extracted via feature extraction. The redundant parts of the image are deleted, and a collection of numerical characteristics are employed to represent the leaf images. These characteristics are used to categories the data. This was accomplished using training data that had already been processed. The process flow and the architecture can be seen in Figures 1 and 2 respectively.

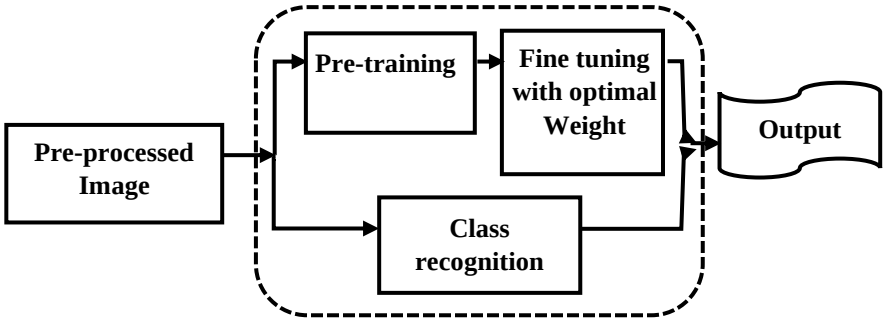


Figure 1: Process flow of DBN-IPSO

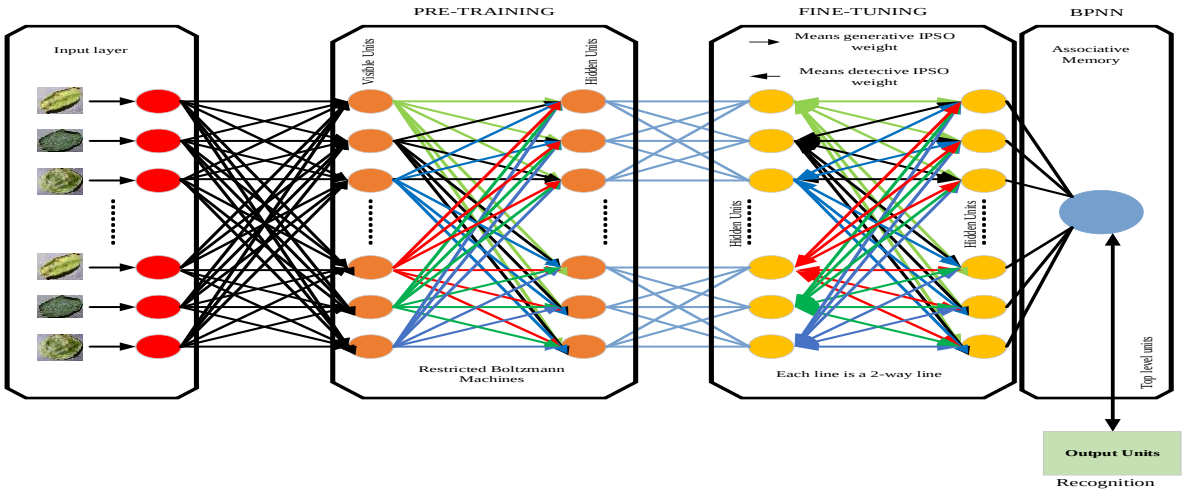


Figure 2: DBN-IPSO architecture

$$T = \{(x_1 y_1), (x_2 y_2), \dots, (x_N y_N)\} \quad (13)$$

Where$(x_1 y_1), i = 1, 2, \dots, N$denotes the sample point, $x_i \in X \subseteq R^n$is data from the sample image while $y_i \in Y$is the label's corresponding tag; The developed system's recognition process involved input data set T into a DBNs network and figuring out how to transfer input X to output Y to produce a generative joint probability distribution model formula.$P(X, Y)$, obtain the result$y_{N+1}$by

$$y_{N+1} = \arg\max_{y_{N+1}} \hat{P}(y_{N+1}|x_{N+1}) \quad (14)$$

wherex_{N+1} is the prediction samplewhich evaluate the classification on the image ofx_{N+1}subject toy_{N+1}.

The following steps was considered:

Step 1: Pre-training: For a specific training set of image data.

$$T = \{x_1, x_2, \dots, x_N\} \quad (15)$$

Step 2: To build a joint distribution model using the energy function for the hidden layer and the visible layer, the combined likelihood greatest probability of the training sample for the given model parameter $\hat{\theta}$was determined by

$$P(v|\hat{\theta}) = \frac{1}{Z(\theta)} \sum_h e^{-E(v,h|\theta)} \quad (16)$$

Step 3: Fine tuning: In deep research, fine-tuning is a standard technique for carrying out supervised learning using labelled sample training set as follow,

$T = \{(x_1^{'}, y_1\ ), (x_2^{'}, y_2\ ), \ldots, (x_N^{'}, y_N\ )\}$ (17)

Step 4: Based on the training set of the statistical classification structure, the multi-RBM network generated sample output, and the top feature vectors corresponding to those vectors were created. Back propagation neural network (BPNN) comes into play, BPNN will apply a Softmax function to a feature vector of a certain dimension.

Step 5: Recognition and detection: Tested sample x_{N+1}network model training exposed network input to feature learning and abstraction to provide an equivalent output.$y_{N+1}$by$y_{N+1} = argmax\hat{P}(y_{N+1}|x_{N+1})$ so it achieved its classification.

3.3 Performance Evaluation Measures

The performance of the developed technique was evaluated using the following criteria: false positive rate (FPR), sensitivity (SEN), specificity (SPEC) and recognition accuracy (ACC). The effectiveness of the performance metrics was evaluated using a confusion matrix. TP, FP, FN and TN are all present.

$$FPT = \frac{\text{FP}}{\text{FP+TN}} * 100 \quad (18)$$

$$Sensitivity = \frac{\text{TP}}{\text{TP+FN}} * 100 \quad (19)$$

$$Specificity = \frac{\text{TN}}{\text{TN+FP}} * 100 \quad (20)$$

$$Precision = \frac{\text{TP}}{\text{TP+FP}} * 100 \quad (21)$$

$$Recognition\ Accuracy = \frac{\text{TP+TN}}{\text{TP+FP+FN+TN}} * 100 \quad (22)$$

Receiver Operating Characteristics (ROC) curve is a statistic for assessing how well a classifier model performs. The ROC curve illustrates the ratio of true positives to FPR, demonstrating how sensitive the classifier model

4. Results and Discussions

The results of the technique are based on the aforementioned categories of maize diseases: Common Rust (CR), Gray Cercospora Leaf (GCL) and Northern Leaf Blight (NLB) dataset. The DBN-IPSO, DBN-PSO and DBN were used as the classifier for the extraction technique. Threshold value had an impact on how well the technique performed. At the threshold value, the best performance obtained 0.40 and above for all technique with respect to aforementioned datasets as shown in Appendices.

4.1 Results using the CR Dataset

Table 1a presents the outcomes of the method DBN, DBN-PSO and DBN-IPSO in relation to the performance indicators of CR datasets. The table reveals that DBN technique approach obtained an ACC of 94.01%, a FPR of 9.04%, a SEN of 96.48%, and a SPEC of 90.96% in 79.03 secs. Similar results were obtained with the DBN-PSO method at 61.58 secs: 8.43% FPR, 97.15% SEN, 91.57% SPEC, and 94.39% ACC. Likewise, in 54.21 secs, the DBN-IPSO technique achieved a FPR of 6.20%, SEN of 97.90%, SPEC of 93.80%, and ACC of 95.88%. The result discloses that the DBN-IPSO technique outperformed DBN-PSO and DBN techniques in terms of FPR, SEN, SPEC and ACC.

Table 1a: Result obtained by the DBN, DBN-PSO and DBN-IPSO techniques with CR datasets

Technique	SEN (%)	SPEC (%)	FPR (%)	ACC (%)	Recognition Time (seconds)
DBN	96.98	90.96	9.04	94.01	79.03
DBN-PSO	97.15	91.57	8.43	94.39	61.58
DBN-IPSO	97.90	93.80	6.20	95.88	54.21

4.2 Results using the GCL Datasets

Table 1b exhibits the products of the method DBN, DBN-PSO and DBN-IPSO in relation to the performance indicators of GCF datasets. The table displays that DBN technique achieved a FPR of 9.04%, SEN of 89.67%, SPEC of 90.96%, and ACC of 90.51% at 49.39 secs. Also, the DBN-PSO technique achieved a FPR of 8.43%, SEN of 90.06%, SPEC of 91.57%, and ACC of 91.10% at 38.49 secs. Correspondingly, the DBN-IPSO technique accomplished a FPR of 6.20%, SEN of 92.59%, SPEC of 93.80%, and ACC of 93.43% at 33.88 secs. The output demonstrates that the DBN-IPSO technique outdone DBN-PSO and DBN technique in terms of FPR, SEN, SPEC and ACC.

Table 1b: Result obtained by the DBN, DBN-PSO and DBN-IPSO techniques with GCL datasets

Technique	SEN (%)	SPEC (%)	FPR (%)	ACC (%)	Recognition Time (seconds)
DBN	89.67	90.96	9.04	90.51	49.39
DBN-PSO	90.06	91.57	8.43	91.10	38.49
DBN-IPSO	92.59	93.80	6.20	93.43	33.88

4.3 Results using the NLB Dataset

Table 1c points out the effects of the method DBN, DBN-PSO and DBN-IPSO in relation to the performance indicators of NLB datasets. The table depicts that DBN technique achieved a FPR of 9.04%, SEN of 98.17%, SEPC of 98.17%, and ACC of 94.27% at 69.15 secs. Similar to this, the DBN-PSO technique attained a FPR of 8.43%, SEN of 98.38%, SPEC of 91.57%, and ACC of 94.69% at 53.88 secs. DBN-IPSO technique attained a FPR of 6.20%, SEN of 98.78%, SPEC of 93.80%, and ACC of 96.09% at 47.44 secs. The result shows that the DBN-IPSO technique outclassed DBN-PSO and DBN technique in terms of FPR, SEN, SPEC an ACC.

Table 1c: Result obtained by the DBN, DBN-PSO and DBN-IPSO techniques with NLB datasets

Technique	SEN (%)	SPEC (%)	FPR (%)	ACC (%)	Recognition Time (seconds)
DBN	98.17	90.96	9.04	94.27	69.15
DBN-PSO	98.38	91.57	8.43	94.69	53.88
DBN-IPSO	98.78	93.80	6.20	96.09	47.44

4.4 Results using the Healthy and Non-Healthy dataset

Table 1d Illustrates the returns of the method DBN, DBN-PSO and DBN-IPSO in relation to the performance indicators used Healthy and Non-Healthy datasets. The table unveils that DBN technique achieved a FPR of 9.04%, SEN of 95.99%, SPEC of 90.96%, and ACC of 94.47% at 180.09 seconds. Similar to this, the DBN-PSO technique attained a FPR of 8.43%, SEN of 96.25%, SPEC of 91.57%, and ACC of 94.83% at 160.02 seconds. DBN-IPSO technique got a FPR of 6.20%, SEN of 97.21%, SPEC of 93.80%, and ACC of 96.18% at 123.58 seconds. The result obtainable from Table 1d uncovers that the DBN-IPSO technique outclassed DBN-PSO and DBN technique in terms of FPR, SEN, SPEC and ACC.

Table 1d: Result obtained by the DBN, DBN-PSO and DBN-IPSO techniques with healthy and non-healthy datasets

Technique	SEN (%)	SPEC (%)	FPR (%)	ACC (%)	Recognition Time (seconds)
DBN	95.99	90.96	9.04	94.47	180.09

DBN-PSO	96.25	91.57	8.43	94.83	160.02
DBN-IPSO	97.21	93.80	6.20	96.18	123.58

4.5 Comparison of total recognition time for maize disease detection and classification system

The SEN, SPEC and ACC of the maize disease detection and classification as seen in Table 2, the results of this study's evaluation of the techniques show that the DBN-PSO methodology increased performance in the recognition rates as shown in Figures 3-7 for all category of dataset used in this study.

Table 2: Combined result obtained by the DBN, DBN-PSO and DBN-IPSO with respect to the datasets

Techniques	CR	GCL	NBL	Healthy/Non-healthy
ACC (%)				
DBN	94.01	90.51	94.27	94.47
DBN-PSO	94.39	91.10	94.69	94.83
DBN-IPSO	95.88	93.43	96.06	96.18
FPR (%)				
DBN	9.04	9.04	9.04	9.04
DBN-PSO	8.43	8.43	8.43	8.43
DBN-IPSO	6.20	6.20	6.20	6.20
SEN (%)				
DBN	96.98	89.47	98.27	95.99
DBN-PSO	97.15	90.06	98.38	96.25
DBN-IPSO	97.90	92.59	98.78	97.21
SPEC (%)				
DBN	90.96	90.96	90.96	90.96
DBN-PSO	91.57	91.57	91.57	91.57
DBN-IPSO	93.80	93.80	93.80	93.80
Recognition time (%)				
DBN	79.03	49.39	69.15	180.09
DBN-PSO	61.58	38.49	53.88	160.02
DBN-IPSO	54.21	33.88	47.44	123.58

It can be concluded from the findings in Table 2 that the DBN-IPSO technique gave an increase of 1.49%, 2.33%, 1.37% and 1.35% ACC, 1.87%, 2.92%, 1.79 and 1.71% ACC for CR, GCL, NBL and healthy/unhealthy datasets using the DBN-PSO and DBN techniques, respectively. The DBN-optimised PSO's features generate more accurate features that help enhance performance, which accounts for the increased ACC. Also, DBN-IPSO technique got an increase of 2.23 %, 2.23 %, 2.23% and 2.23% SPEC, 2.84%, 2.84%, 2.84% and 2.84% SPEC for CR, GCL, NBL and healthy/unhealthy dataset respectively over DBN-PSO and DBN technique. Furthermore, DBN-IPSO technique gave an increase of 0.75%, 2.53%, 0.4% and 0.96% SEN, 0.92%, 3.12%, 0.51% and 1.22% SEN for CR, GCL, NBL and healthy/unhealthy dataset respectively over DBN-PSO and DBN technique.

The improved performance in terms of SEN, SPEC and FPR achieved the DBN-IPSO technique over DBN-PSO and DBN which attributed to the adaptive threshold of the DBN-IPSO. This supported the findings of who discovered that a feature selection might improve the rate of ACC (Devaraj *et al.*, 2020; Pudumalar *et al.*, 2017) discovered that using the IPSO algorithm to optimize features increases classification accuracy rate and that the IPSO gives better accuracy than existing technique. Using feature selection, (Devaraj *et al.*, 2020) obtained highly discriminating features with high classification rates.

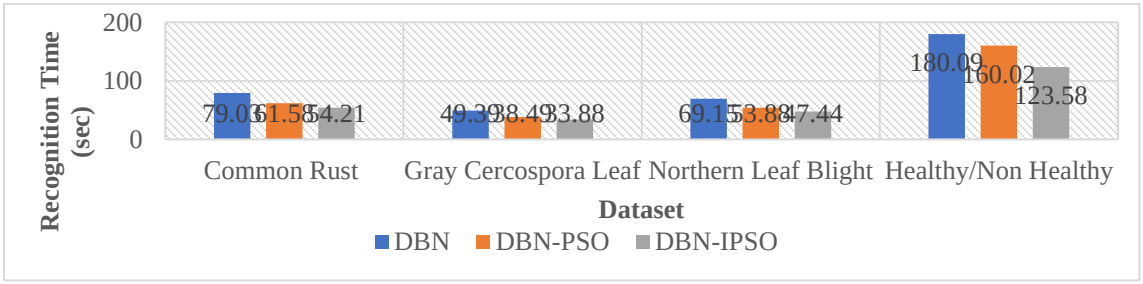


Figure 3: Comparison of total recognition time or maize disease detection and classification system

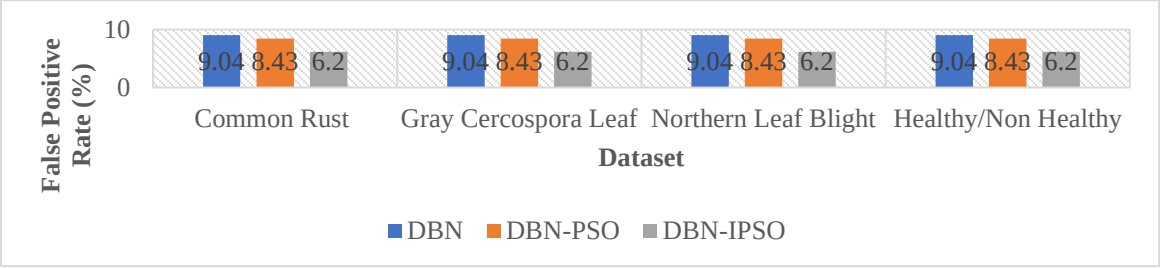


Figure 4: Comparison of FPR for maize disease detection and classification system

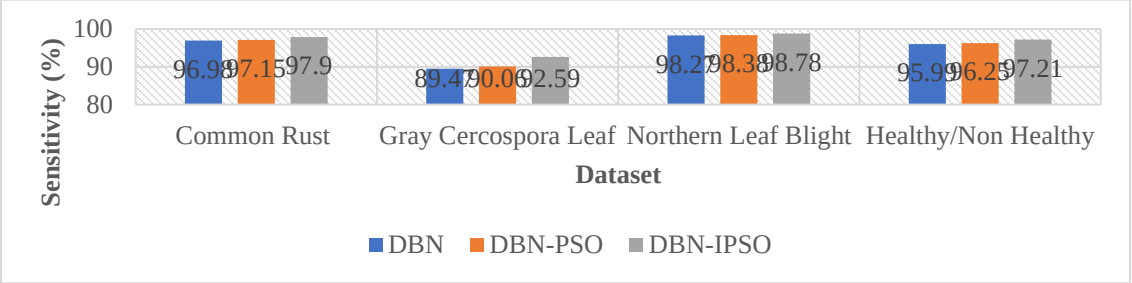


Figure 5: Comparison of sensitivity for maize detection and classification system

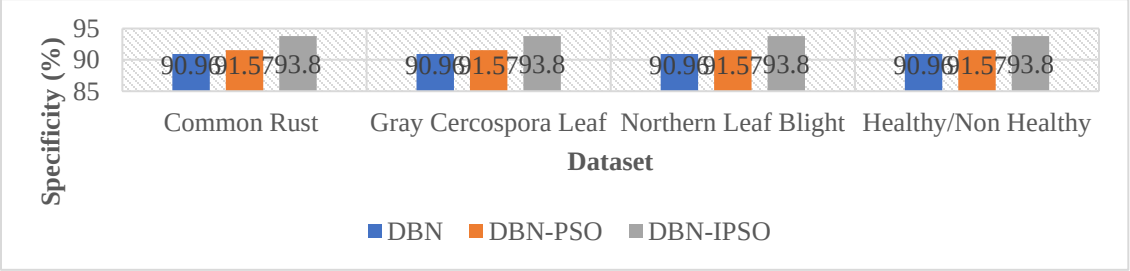


Figure 6: Comparison of specificity or maize disease detection and classification system

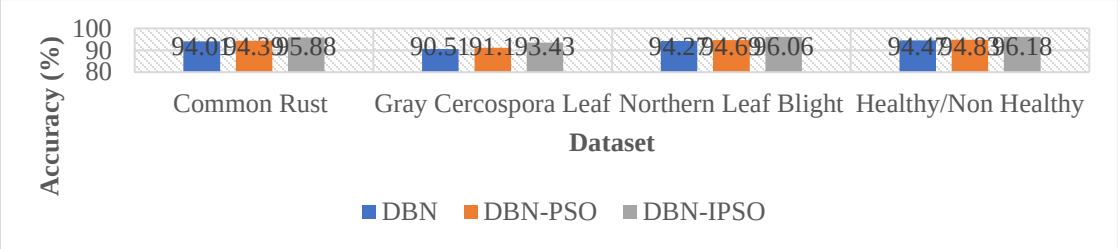


Figure 7: Comparison of recognition accuracy or maize disease detection and classification system

As a result of the aforementioned finding, all datasets used in the study demonstrated improved ACC, SPE, SEN, and FPR when DBN and IPSO approach are combined. This suggests that the DBN-PSO technique was able to produce a solution of higher quality than the DBN strategies already in use. Therefore, when it came to the classification and detection of the maize diseases, DBN-IPSO technique performed better than DBN technique according to the aforementioned criteria. Furthermore, the FPR achieved by the technique further proves its correctness. Therefore, DBN-IPSO technique gave an improved SEN SPEC, ACC and FPR. It is clearly obvious that DBN-IPSO technique is well matched to the other orthodox infected maize detection methods based on its performance in view of SEN, SPEC ACC and FPR.

5. Conclusion

The study proposed a novel model for maize disease detection and classification by combining DBNs with IPSO. The primary objective of the research was to develop an efficient and accurate system that can automatically detect and classify plant diseases from images. The researchers collected a large dataset of plant images with various disease conditions. This dataset serves as the basis for training and evaluating the disease detection and classification model. The IPSO algorithm was employed to optimise the selection of the most discriminative DBN parameters that can represent the different disease patterns effectively. DBNs with IPSO were employed for automatic learning and representation of the extracted features and classification. The model was trained on the collected dataset, where the IPSO selects the most relevant DBN parameter, and the DBN with IPSO learn the hierarchical representations. The performance of the model was then evaluated using standard metrics, such as accuracy, sensitivity, specificity, and recognition time. The research developed an innovative solution for the maize plant disease detection by combining improved PSO and DBNs, with this the integration of the techniques improves the accuracy and efficiency of the system, making it a promising approach for automated plant disease diagnosis and surveillance in agriculture.

In this research, classification and disease detection system for maize, the key features of the DBN-IPSO techniques was explored. Three thousand eight hundred and fifty-two images comprising four categories of datasets namely CR, GCL, NLB, and healthy/unhealthy were employed to assess the developed approach. With different threshold values, the developed DBN-IPSO was used to train and test these images. For future work, the developed model can be utilized to address problems related to maize disease detection and classification system in the prevention of maize disease related illnesses by using other swarm intelligence algorithm such as Cultural Swam Optimization could be hybridized with IPSO to further examine the performance of the system and possibly improve upon the results.

Declaration of Competing Interest

We declare that we have no conflicts of interest regarding this article.

References

Adekunle, I. (2020). Implementation of improved machine learning techniques for plant disease detection and classification. *Int J Res Innov Appl Sci (IJRIAS), 5*(6).

Ali, A., Zia, M. A., Latif, M. A., Zulfqar, S., & Asim, M. (2023). A comparative study of deep learning techniques for boll rot disease detection in cotton crops. *Agricultural Sciences Journal, 5*(1), 58-71.

Almodfer, R., Mudhsh, M., & Zhao, J. (2023). Pyramided and optimized blurred shape model for plant leaf classification. *IET Image Processing, 17*(10), 2838-2854.

Alpyssov, A., Uzakkyzy, N., Talgatbek, A., Moldasheva, R., Bekmagambetova, G., Yessekeyeva, M., Tolstoy, A. (2023). Assessment of plant disease detection by deep learning. *Eastern-European Journal of Enterprise Technologies, 121*(2).

Anitha, J., & Saranya, N. (2022). Cassava leaf disease identification and detection using deep learning approach. *International Journal of Computers Communications & Control, 17*(2).

Arsenovic, M., Karanovic, M., Sladojevic, S., Anderla, A., & Stefanovic, D. (2019). Solving current limitations of deep learning based approaches for plant disease detection. *Symmetry, 11*(7), 939.

Awuor, F., Kimeli, K., Rabah, K., & Rambim, D. (2013). ICT solution architecture for agriculture. Paper presented at the 2013 *Ist-Africa Conference & Exhibition.*

Awuor, F., Rabah, K., & Kimeli, K. (2013). E-agriculture framework: modeling stakeholders' competing and conflicting interests.

Brahimi, M., Mahmoudi, S., Boukhalfa, K., & Moussaoui, A. (2019). Deep interpretable architecture for plant diseases classification. Paper presented at the 2019 *Signal Processing: Algorithms, Architectures, Arrangements, and Applications (SPA)*.

Bruinsma, J. (2017). World agriculture: towards 2015/2030: an FAO study: Routledge.

Chen, J., Chen, J., Zhang, D., Sun, Y., & Nanehkaran, Y. A. (2020). Using deep transfer learning for image-based plant disease identification. *Computers and Electronics in Agriculture, 173*, 105393.

Ciresan, D. C., Meier, U., Masci, J., Gambardella, L. M., & Schmidhuber, J. (2011). Flexible, high performance convolutional neural networks for image classification. Paper presented at the *Twenty-second International Joint Conference on Artificial Intelligence.*

Devaraj, A. F. S., Elhoseny, M., Dhanasekaran, S., Lydia, E. L., & Shankar, K. (2020). Hybridization of firefly and improved multi-objective particle swarm optimization algorithm for energy efficient load balancing in cloud computing environments. *Journal of Parallel and Distributed Computing, 142*, 36-45.

Dewi, C., Chen, R.-C., Hendry, & Hung, H.-T. (2020). Comparative analysis of restricted Boltzmann machine models for image classification. Paper presented at the *Intelligent Information and Database Systems: 12th Asian Conference, ACIIDS* 2020, Phuket, Thailand, March 23–26, 2020, Proceedings, Part II 12.

Fan, F., Roy, T., & Roy, K. (2020). Classification and detection rice leaf diseases using information and communication technology ICT tools. *International Journal of Advanced Engineering Research and Science, 7*(6), 460-470.

Fan, X., Zhou, J., Xu, Y., & Yang, J. (2021). Corn diseases recognition method based on multi-feature fusion and improved deep belief network. *preprint. https://doi. org/10.21203/rs, 3*.

Fan, X., Luo, P., Mu, Y., Zhou, R., Tjahjadi, T., & Ren, Y. (2022). Leaf image based plant disease identification using transfer learning and feature fusion. *Computers and Electronics in Agriculture, 196*, 106892.

Ha, J. G., Moon, H., Kwak, J. T., Hassan, S. I., Dang, M., Lee, O. N., & Park, H. Y. (2017). Deep convolutional neural network for classifying Fusarium wilt of radish from unmanned aerial vehicles. *Journal of Applied Remote Sensing, 11*(4), 042621-042621.

Huang, Z.-Y., Chiang, C.-C., Chen, J.-H., Chen, Y.-C., Chung, H.-L., Cai, Y.-P., & Hsu, H.-C. (2023). A study on computer vision for facial emotion recognition. *Scientific Reports, 13*(1), 8425.

Jasim, M. A., & Al-Tuwaijari, J. M. (2020). Plant leaf diseases detection and classification using image processing and deep learning techniques. Paper presented at the 2020 *International Conference on Computer Science and Software Engineering (CSASE)*.

Kerkech, M., Hafiane, A., & Canals, R. (2018). Deep leaning approach with colorimetric spaces and vegetation indices for vine diseases detection in UAV images. *Computers and Electronics in Agriculture, 155*, 237-243.

Kumar, S., Jain, A., Shukla, A. P., Singh, S., Raja, R., Rani, S., . . . Masud, M. (2021). A comparative analysis of machine learning algorithms for detection of organic and nonorganic cotton diseases. *Mathematical Problems in Engineering, 2021*(1), 1790171.

Liang, X., Li, W., Zhang, Y., & Zhou, M. (2015). An adaptive particle swarm optimization method based on clustering. *Soft Computing, 19*, 431-448.

Malik, O. A., Ismail, N., Hussein, B. R., & Yahya, U. (2022). Automated real-time identification of medicinal plants species in natural environment using deep learning models-a case study from Borneo Region. *Plants, 11*(15), 1952.

Mohanty, S. P., Hughes, D. P., & Salathé, M. (2016). Using deep learning for image-based plant disease detection. *Frontiers in Plant Science, 7*, 215232.

Nilsson, R. H., Larsson, K.-H., Taylor, A. F. S., Bengtsson-Palme, J., Jeppesen, T. S., Schigel, D., Tedersoo, L. (2019). The UNITE database for molecular identification of fungi: handling dark taxa and parallel taxonomic classifications. *Nucleic Acids Research, 47*(D1), D259-D264.

Pudumalar, S., Ramanujam, E., Rajashree, R. H., Kavya, C., Kiruthika, T., & Nisha, J. (2017). Crop recommendation system for precision agriculture. Paper presented at the 2016 *Eighth International Conference on Advanced Computing (Icoac)*.

Roy, K., Chaudhuri, S. S., Frnda, J., Bandopadhyay, S., Ray, I. J., Banerjee, S., & Nedoma, J. (2023). Detection of tomato leaf diseases for agro-based industries using novel PCA DeepNet. *Ieee Access, 11*, 14983-15001.

Salman, Y. E., Daeef, A. Y., Perera, A. G., & Al-Naji, A. (2024). Computer vision for automated facial characteristics detection. *Electrical Engineering Technical Journal, 1*(1), 13-19.

Singh, S., & Mukherjee, S. (2024). Automated detection of plant leaf diseases using image processing techniques.

Sladojevic, S., Arsenovic, M., Anderla, A., Culibrk, D., & Stefanovic, D. (2016). Deep neural networks based recognition of plant diseases by leaf image

classification. *Computational Intelligence and Neuroscience, 2016.*

Sushma, D., Ramya, D., Yashaswini, D., Chaithra, T., & Kavaya, H. (2024). Detecting plant leaf diseases through image processing and cnn with preventive measures. *International Research Journal on Advanced Engineering and Management (IRJAEM), 2*(05), 1710-1713.

Syed-Ab-Rahman, S. F., Hesamian, M. H., & Prasad, M. (2022). Citrus disease detection and classification using end-to-end anchor-based deep learning model. *Applied Intelligence, 52*(1), 927-938.

Tiwari, N., & Gupta, B. K. (2023). Developing a neural network-based model for identifying medicinal plant leaves using image recognition techniques. *SAMRIDDHI: A Journal of Physical Sciences, Engineering and Technology, 15*(03), 301-311.

Wang, G., Qiao, J., Bi, J., Jia, Q.S., & Zhou, M. (2019). An adaptive deep belief network with sparse restricted Boltzmann machines. *IEEE Transactions on Neural Networks and Learning Systems, 31*(10), 4217-4228.

Xing, S., Lei, Y., Wang, S., & Jia, F. (2020). Distribution invariant deep belief network for intelligent fault diagnosis of machines under new working conditions. *IEEE Transactions on Industrial Electronics, 68*(3), 2617-2625.

Xu, H.S., Guo, S.M., Zhu, L., & Xing, J.C. (2020). Growth, physiological and transcriptomic analysis of the perennial ryegrass Lolium perenne in response to saline stress. *Royal Society Open Science, 7*(7), 200637.

Zenat, M., Akther, E., Haque, N. N., Hasan, M. R., Begum, M., Munshi, J. L., Alam, M. A. (2024). Antifungal activity of various plant extracts against aspergillus and penicillium species isolated from Leather-Borne Fungus. *Microbiology Research Journal International, 34*(1), 10-23

Zhang, H., Yanne, P., & Liang, S. (2012). Plant species classification using leaf shape and texture. Paper presented at the 2012 *International Conference on Industrial Control and Electronics Engineering.*

Zhang, K., Wu, Q., Liu, A., & Meng, X. (2018). Can deep learning identify tomato leaf disease? *Advances in Multimedia, 2018.*

Zhang, Y., Wang, S., & Ji, G. (2015). A comprehensive survey on particle swarm optimization algorithm and its applications. *Mathematical Problems in Engineering, 2015*.

Appendices

A: Table showing the performance of DBN-IPSO, DBN-PSO and DBN techniques using CR datasets Images = 1192

DBN

Threshold	TP (CR)	FN (healthy)	FP (CR)	TN (healthy)	SEN (%)	SPEC (%)	FPR (%)	ACC (%)	Time (sec)
0.30	1157	35	112	1050	97.06	90.36	9.64	93.76	78.48
0.34	1157	35	110	1052	97.06	90.53	9.47	93.84	79.42
0.37	1156	36	108	1054	96.98	90.71	9.29	93.88	79.26
0.40	1156	36	105	1057	96.98	90.96	9.04	94.01	79.03

DBN-PSO

Threshold	TP (CR)	FN (healthy)	FP (CR)	TN (healthy)	SEN (%)	SPEC (%)	FPR (%)	ACC (%)	Time (sec)
0.30	1159	33	106	1056	97.23	90.88	9.12	94.10	63.26
0.34	1159	33	103	1059	97.23	91.14	8.86	94.22	63.46
0.37	1159	33	100	1062	97.23	91.39	8.61	94.35	59.14
0.40	1158	34	98	1064	97.15	91.57	8.43	94.39	61.58

DBN-IPSO

Threshold	TP (CR)	FN (healthy)	FP (CR)	TN (healthy)	SEN (%)	SPEC (%)	FPR (%)	ACC (%)	Time (sec)
0.30	1168	24	80	1082	97.99	93.12	6.88	95.58	55.55
0.34	1168	24	77	1085	97.99	93.37	6.63	95.71	54.86
0.37	1167	25	74	1088	97.90	93.63	6.37	95.79	54.64
0.40	1167	25	72	1090	97.90	93.80	6.20	95.88	54.21

B: Table showing the performance of DBN-IPSO, DBN-PSO and DBN techniques using GCL datasets Images = 513

DBN

Threshold	TP (GCL)	FN (healthy)	FP (GCL)	TN (healthy)	SEN (%)	SPEC (%)	FPR (%)	ACC (%)	Time (sec)
0.30	460	53	112	1050	89.67	90.36	9.64	90.15	49.05
0.34	460	53	110	1052	89.67	90.53	9.47	90.27	49.64
0.37	459	54	108	1054	89.47	90.71	9.29	90.33	49.54
0.40	459	54	105	1057	89.47	90.96	9.04	90.51	49.39

DBN-PSO

Threshold	TP (GCL)	FN (healthy)	FP (GCL)	TN (healthy)	SEN (%)	SPEC (%)	FPR (%)	ACC (%)	Time (sec)
0.30	464	49	106	1056	90.45	90.88	9.12	90.75	39.54
0.34	463	50	103	1059	90.25	91.14	8.86	90.87	39.67
0.37	463	50	100	1062	90.25	91.39	8.61	91.04	36.96
0.40	462	51	98	1064	90.06	91.57	8.43	91.10	38.49

DBN-IPSO

Threshold	TP (GCL)	FN (healthy)	FP (GCL)	TN (healthy)	SEN (%)	SPEC (%)	FPR (%)	ACC (%)	Time (sec)
0.30	477	36	80	1082	92.98	93.12	6.88	93.07	34.72
0.34	476	37	77	1085	92.79	93.37	6.63	93.19	34.29
0.37	476	37	74	1088	92.79	93.63	6.37	93.37	34.15
0.40	475	38	72	1090	92.59	93.80	6.20	93.43	33.88

C: Table showing the performance of DBN-IPSO, DBN-PSO and DBN techniques using NLB datasets Images = 985

DBN

Threshold	TP (NLB)	FN (healthy)	FP (NLB)	TN (healthy)	SEN (%)	SPE C(%)	FPR (%)	ACC (%)	Time (sec)
0.30	968	17	112	1050	98.27	90.36	9.64	93.99	68.67
0.34	967	18	110	1052	98.17	90.53	9.47	94.04	69.49
0.37	968	17	108	1054	98.27	90.71	9.29	94.18	69.36
0.40	967	18	105	1057	98.17	90.96	9.04	94.27	69.15

DBN-PSO

Threshold	TP (NLB)	FN (healthy)	FP (NLB)	TN (healthy)	SEN (%)	SPEC (%)	FPR (%)	ACC (%)	Time (sec)
0.30	969	16	106	1056	98.38	90.88	9.12	94.32	55.36
0.34	969	16	103	1059	98.38	91.14	8.86	94.46	55.53
0.37	968	17	100	1062	98.27	91.39	8.61	94.55	51.74
0.40	969	16	98	1064	98.38	91.57	8.43	94.69	53.88

DBN-IPSO

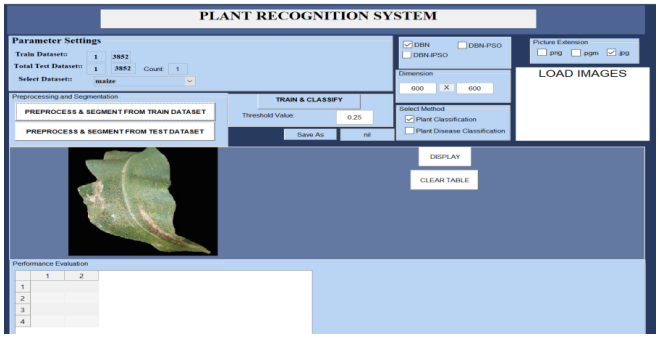
Threshold	TP (NLB)	FN (healthy)	FP (NLB)	TN (healthy)	SEN (%)	SPEC (%)	FPR (%)	ACC (%)	Time (sec)
0.30	973	12	80	1082	92.98	93.12	6.88	93.07	34.72
0.34	973	12	77	1085	92.79	93.37	6.63	93.19	34.29
0.37	973	12	74	1088	92.79	93.63	6.37	93.37	34.15
0.40	973	12	72	1090	92.59	93.80	6.20	93.43	33.88

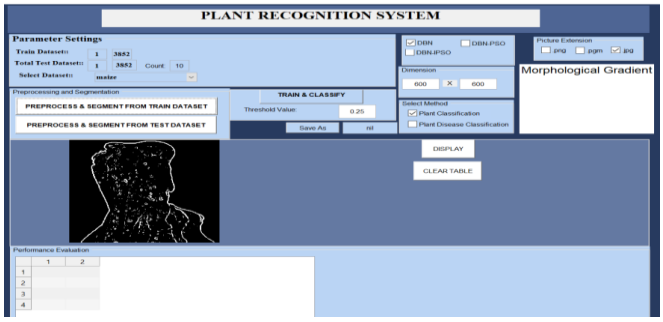
D: Table showing the performance of DBN-IPSO, DBN-PSO and DBN techniques using Healthy/non healthy datasets Images = 520

Threshold	TP	FN	FP	TN	SEN (%)	SPEC (%)	FPR (%)	ACC (%)	Time (sec)	Technique
0.02	410	10	2	98	97.62	98.00	2.00	97.69	164.01	DBN-PSO
0.35	412	8	3	97	98.10	97.00	3.00	97.88	166.33	DBN-PSO
0.45	414	6	4	96	98.57	96.00	4.00	98.08	165.65	DBN-PSO
0.40 and above	417	3	5	95	99.29	95.00	5.00	98.46	167.94	DBN-PSO
Threshold	TP	FN	FP	TN	SEN (%)	SPEC (%)	FPR (%)	ACC (%)	Time (sec)	Technique
0.02	396	24	12	88	94.29	88.00	12.00	93.08	175.32	DBN
0.35	398	22	13	87	94.76	87.00	13.00	93.27	173.71	DBN
0.45	400	20	14	86	95.24	86.00	14.00	93.46	176.91	DBN

0.40 and above	401	19	14	86	95.48	86.00	14.00	93.65	179.43	DBN

GRAPHICAL USER INTERFACE





F: Table showing the performance of DBN-IPSO, DBN-PSO and DBN techniques using CR datasets Images = 1192

DBN

Threshold	TP (CR)	FN (healthy)	FP (CR)	TN (healthy)	SEN (%)	SPEC (%)	FPR (%)	ACC (%)	Time (sec)
0.30	1157	35	112	1050	97.06	90.36	9.64	93.76	78.48
0.34	1157	35	110	1052	97.06	90.53	9.47	93.84	79.42
0.37	1156	36	108	1054	96.98	90.71	9.29	93.88	79.26
0.40	1156	36	105	1057	96.98	90.96	9.04	94.01	79.03

DBN-PSO

Threshold	TP (CR)	FN (healthy)	FP (CR)	TN (healthy)	SEN (%)	SPEC (%)	FPR (%)	ACC (%)	Time (sec)
0.30	1159	33	106	1056	97.23	90.88	9.12	94.10	63.26
0.34	1159	33	103	1059	97.23	91.14	8.86	94.22	63.46
0.37	1159	33	100	1062	97.23	91.39	8.61	94.35	59.14
0.40	1158	34	98	1064	97.15	91.57	8.43	94.39	61.58

DBN-IPSO

Threshold	TP (CR)	FN (healthy)	FP (CR)	TN (healthy)	SEN (%)	SPEC (%)	FPR (%)	ACC (%)	Time (sec)
0.30	1168	24	80	1082	97.99	93.12	6.88	95.58	55.55
0.34	1168	24	77	1085	97.99	93.37	6.63	95.71	54.86
0.37	1167	25	74	1088	97.90	93.63	6.37	95.79	54.64
0.40	1167	25	72	1090	97.90	93.80	6.20	95.88	54.21

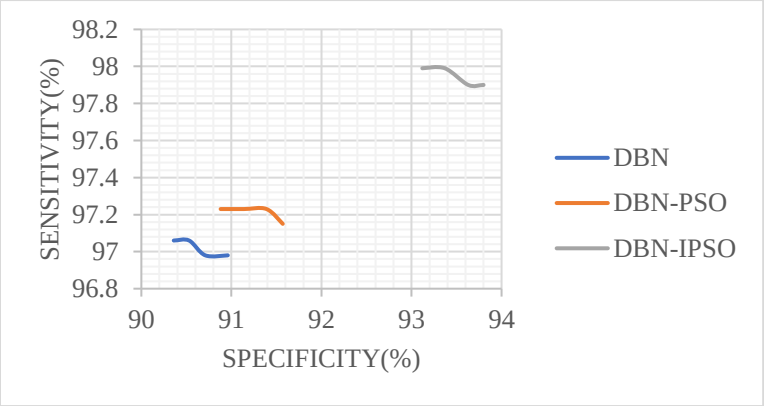


Figure 1.: ROC graph for DBN, DBN-PSO and DBN-IPSO classifiers clearly shows the ROC curves of the three classifiers. The curve shows that DBN-IPSO yields a more excellent result than DBN and DBN-PSO in terms of sensitivity and specificity.

G: Table showing the performance of DBN-IPSO, DBN-PSO and DBN techniques using GCL datasets Images = 513

DBN

Threshold	TP (GCL)	FN (healthy)	FP (GCL)	TN (healthy)	SEN (%)	SPEC (%)	FPR (%)	ACC (%)	Time (sec)
0.30	460	53	112	1050	89.67	90.36	9.64	90.15	49.05
0.34	460	53	110	1052	89.67	90.53	9.47	90.27	49.64
0.37	459	54	108	1054	89.47	90.71	9.29	90.33	49.54
0.40	459	54	105	1057	89.47	90.96	9.04	90.51	49.39

DBN-PSO

Threshold	TP (GCL)	FN (healthy)	FP (GCL)	TN (healthy)	SEN (%)	SPEC (%)	FPR (%)	ACC (%)	Time (sec)
0.30	464	49	106	1056	90.45	90.88	9.12	90.75	39.54
0.34	463	50	103	1059	90.25	91.14	8.86	90.87	39.67
0.37	463	50	100	1062	90.25	91.39	8.61	91.04	36.96

0.40	462	51	98	1064	90.06	91.57	8.43	91.10	38.49

DBN-IPSO

Threshold	TP (GCL)	FN (healthy)	FP (GCL)	TN (healthy)	SEN (%)	SPEC (%)	FPR (%)	ACC (%)	Time (sec)
0.30	477	36	80	1082	92.98	93.12	6.88	93.07	34.72
0.34	476	37	77	1085	92.79	93.37	6.63	93.19	34.29
0.37	476	37	74	1088	92.79	93.63	6.37	93.37	34.15
0.40	475	38	72	1090	92.59	93.80	6.20	93.43	33.88

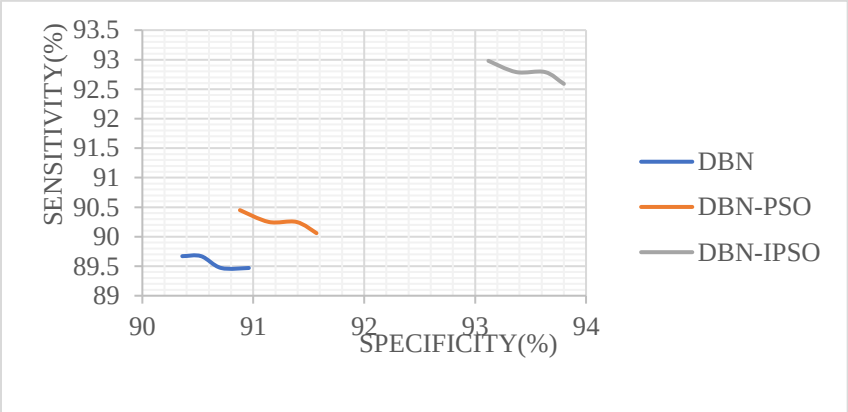


Figure 2: ROC graph for DBN, DBN-PSO and DBN-IPSO classifiers clearly shows the ROC curves of the three classifiers. The curve shows that DBN-IPSO yields a more excellent result than DBN and DBN-PSO in terms of sensitivity and specificity.