International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

An Adapted Moth Search with Convolutional Neural Network with Replicator Neuron-Based Leaf Disease Detection

Majed Aborokbah [1]

Submitted: 07/02/2024 **Revised:** 15/03/2024 **Accepted:** 21/03/2024

Abstract: Crop quality and yield can be significantly impacted by plant diseases, and even though plants may be examined for indicators of illness by trained biologists or farmers, this is typically an inexact and labor-intensive process. This study employs IoT and AI-based monitoring strategies to design and develop a smart method for classifying leaf illnesses. So as to measure the effectiveness of these two approaches, simulation results are compared in this work. In the first section, the data of photos of plants from the Plant Village data set augmented using a Hybrid CNN (Convolutional Neural Network) with RNN (Replicator Neural Network) and named as HCRNN, and deep features mined from these images. So as to enhance the precision of the segmentation procedure, the plant images undergo preliminary processing with an adaptive kaun filter. Next, a Glowworm Swarm Optimization based Clustering (GSOC) technique is used to isolate the plant region in the processed image. The HCRNN was then used to classify the plant disease based on the retrieved features. The projected method uses the Adapted Moth Search (AMS) Algorithm to fine-tune the CNN's (Convolutional Neural Network) hyperparameter in order to enhance its classification accuracy. Two branches of the model are used to learn from the T2- and Diffusion-weighted MRI data: one employs a ten-layer CNN After applying HCRNN to classify the relevant characteristics, and then assesses the quality of the classification in terms of precision, recall, and f-score. Extensive field testing indicates that the technique is useful in hot and humid environments and that it is more accurate than other categorization schemes at recognizing classes of disease in leaves.

Keywords: *plant disease, Internet of Things, artificial intelligence, Convolutional Neural Network, Spiking Neuron, Adapted Moth Search*

1. Introduction

There has been an uptick in the petition for agricultural crops in line with the rise in population, and about 75% of farmers still use time-tested methods [1]. Crop yield is negatively impacted by regional climate and soil variances, disease outbreaks, and an ever-increasing global population, making it impossible for these methods to keep up [1]. The belongings of climate and soil types cannot be tracked in the same way using conventional methods. The quantity of fertilizer or pesticide needed for a given crop cannot be determined automatically either; because of this, pesticides and artificial fertilizers may be used in excess. Agricultural prices rise, and the soil and human health are negatively impacted by the chemicals used. In addition, there is currently no automated system for early disease prediction and plant disease classification [2]. For agricultural disease identification using conventional methods, humans are required.

The aforementioned highlights the fact that the main problems with conventional farming are its high prices, dependence on human intervention, inadequate unfortunate crop eminence, yields, and harmful effects of the overuse of fertilizers and pesticides. There is an crucial need to speech the problems above, such as the misuse of managed water systems, the use of harmful pesticides, the lack of effective contamination controls, and the climatic impacts on farming. Any misstep in diagnosis could result in faulty pest management and unnecessary chemical use. One of the major difficulties is accurately spraying a designated area with a sufficient amount of pesticide or fertilizer. Early disease detection is crucial for healthy plant growth [3]. Predicting the growth of plant diseases in a crop field based on environmental parameters collected via the Internet of Things (IoTs) is made possible with the use of Machine Learning (ML).

The agricultural sector has witnessed substantial progress in recent years, attributable to the heightened interest and developments in Internet of Things (IoT) and Deep Learning (DL) [4-6]. The application of IoT enables the contemporaneous acquisition of real-time data. It aids in the economical use of water, fertilizer, and energy [7]. Two branches of the model are used to learn from the T2- and Diffusion-weighted MRI data: one employs a ten-layer CNN IoT device that can effectively monitor the requirement for pesticides and herbicides, as well as the early visible and non-visual symptoms of the disease. When combined with IoT, images may be fused with quantitative and genomic datasets, which is where DL approaches really shine [11]. Deep CNNs (DCNNs) are useful for the automated identification of features and feature selection when extended to plant disease diagnosis [12], weed identification [13], forecasting yields [15], fruit enumeration [14], and presentation of detected fruits, diseases, and weeds [16]. These methods streamline the tedious processes of manually extracting features and recognizing images and boost accuracy [17].

In addition to using ML methods to assess hyperspectral data from wheat leaves, Azadbakht et al. [18] successfully classified 82% of rice leaves as either normal or damaged. Conventional procedure involves the initial classification of plant diseases based on observable symptoms manifesting on the leaves of the

[1] *Faculty of Computers and Information Technology University of Tabuk, Saudi Arabia, m.aborokbah@ut.edu.sa*
ORCID ID : 0000-0001-7376-1458

plant [7, 8]. Traditional machine vision (MV) processes, such as manual categorization, necessitate feature extraction by hand, but CNN does not. Instead, CNN can operate with simply visual data. In contemporary times, the preferred approach for addressing challenges in learning has increasingly centered on the adoption of DL methodologies, with CNNs emerging as a prominent choice [19, 20].

Overfitting and low accuracy are just two of the many challenges that arise when attempting to classify new leaf diseases as they appear in plants. The agriculture sector relies on accurate and error-free analysis to separate good from bad products. To a certain extent, detection and classification tasks are where deep CNNs really shine as an efficient model of autonomous feature extraction. The network is able to properly classify images because of its ability to learn on its own [21]. In recent times, there has been a significant increase in the application of DL methods for image classification in agriculture, particularly for the diagnostic evaluation of plant diseases [22, 23]. Deep CNNs, on the other hand, necessitate extensive amounts of training data, are translatable, and have several parameters that must be set and fine-tuned. To classify a multitude of leaf diseases affecting plants and fruits, this study introduces an efficient framework designed for use in the feature extraction process. This is achieved through the utilization of a dedicated deep transfer learning model. The primary contributions of this proposed work are outlined as follows:

- In the first segment, the data of images of plants from the Plant Village data set are augmented using a HCRNN and deep features extracted from these images.
- Secondly, to improve the accuracy of the segmentation process, plant images undergo pre-processing using an adaptive Kuan filter. Next, a GSO based Clustering (GSOC) technique is used to isolate the plant region in the processed image.
- The third step is to extract the features using the HCRNN and then use it to classify the plant diseases. The hyperparameter of CNN is improved with the use of the AMS Algorithm to boost the classification accuracy of the suggested method.
- The accuracy, sensitivity, f-score, and precision of the resulting classifications are next evaluated. Extensive field studies show that the approach is trustworthy in detecting disease classes of leaves and that it works well in hot and humid environments compared to previous categorization schemes.

The subsequent sections of this project are organized as follows. The second part encompasses a literature review focusing on methodologies for classifying plant diseases. Section 3 underscores the significance of the methodology employed. Section 4 elucidates the outcomes derived from the proposed strategy, and a concise summary is presented in Section 5.

2. Related Work

The current methods used in the IoTs for identifying and categorizing plant diseases have been discussed. Two branches of the model are used to learn from the T2- and Diffusion-weighted MRI data: one employs a ten-layer CNN For the purpose of reducing the severity of this agricultural tragedy, Sowmiya and Krishnaveni [24] introduced a ML model called IQWO- PCA (Improved Quantum Whale Optimization with Principal Component Analysis) to analyze a collection of tomato disease photos. The hyperparameters have undergone systematic optimization in the rigorous analysis, and a publicly accessible dataset on plant diseases has been utilized to construct the network employed in this investigation. The transmission learning-based DNN is built on top of pre-existing prototype networks like Alexnet, VGG16, ResNet50, and DenseNet121. The primary characteristics of the dataset are extracted using a method for optimizing composite construction components. The retrieved information is fed into a deep neural network in order to improve illness classification in tomatoes.

Two branches of the model are used to learn from the T2- and Diffusion-weighted MRI data: one employs a ten-layer CNN Kondekar and Bodhe [25] was proposed a novel OACB (Optimized Attentional Capsule_BiLSTM) model for accurately identifying a plant disease from the available samples. A novel Chaotic Sparrow Search Optimization (CSSO) approach is used to fine-tune the hyperparameters, increasing the effectiveness of the proposed classifier. Simulation results demonstrate that the suggested study when compared to other current methods, achieves a higher detection accuracy value of 97.06% using the MATLAB tool and the Plant Village Dataset.

To improve feature discrimination and processing performance, Saberi Anari [26] was used a model to extract features and then used Modern support vector machine (SVM) models. After settling on a model, the training phase then uses that information to set the RBF's kernel parameters. Six sets of leaf images, encompassing both healthy and damaged leaves from apple, cotton, maize, grape, pepper, and rice plants, were scrutinized using data derived from the PlantVillage and UCI databases. Two branches of the model are used to learn from the T2- and Diffusion-weighted MRI data: one employs a ten-layer CNN. Approximately 90,000 images were generated through the classification process, and preliminary findings from the experimental implementation phase indicate the potential for an effective model in classification processes. This holds promise for various forthcoming applications in the agricultural sector, particularly in the investigation of leaf diseases.

The use of a DNN (deep neural network) was proposed by Al-bayati & Üstünda [27] as a means of diagnosing leaf diseases in apple trees. A new architecture for a plant disease detection system (PDDS) is created, with feature extraction using Speeded up robust feature (SURF), and optimization using the Grasshopper Optimization Algorithm (GOA) to increase recognition and classification accuracy. Classification parameters are calculated to demonstrate the effectiveness of the suggested work, including Recall, Precision, Error, F-measure, and Accuracy.

The authors of [28], who conducted a survey of IoT and DL-based schemes, argued that ambient factors and the lack of crisp borders around the diseased zone pose significant difficulties in ROI segmentation and, by extension, disease prediction. In addition, using IoTs methods for crop monitoring and disease detection presents a difficulty in terms of power consumption. These problems were discussed in [29], where a rider neural network based on the sine cosine process was presented. To mitigate energy consumption during the transmission of images from the basis to the sink, their solution incorporated a routing procedure. Additionally, a median filter was applied to eliminate undesired noise in the images. Despite achieving an accuracy of 91.18% on the Plant Village dataset [30], their algorithm highlights the potential for enhancement in disease forecasting precision.

For the identification of numerous agricultural diseases, the authors of [31] was recommended combining IoT and DL

models. They used MDFC-ResNet (Multidimensional Feature Compensation-Residual Neural Network) and claimed a 93.96% improvement in accuracy during training. Subsequently, the 'FruGar' system, developed by the authors of [32], was offered as a web-based software for disease identification in coffee leaves. They used Social IoT (SIoT) to combine and share disparate data sources, including photos taken with a camera and smartphone with environmental characteristics like temperature, humidity, and pH measured by sensors placed in various locations. Using the MobileNetV2 model, they were able to achieve a classification accuracy of 94.58%. Sensors were utilized to capture parametric data, and then feature extraction, minimizing data, and classification were performed using knowledge-based fusion techniques, as described by the paper's authors [33]. Accuracy in disease detection was improved to 97.5% by combining DCNN with IoT.

Summary: All things considered, the methods mentioned above for leaf disease detection using images of leaves showed promise, but they were only tested on modest datasets and yielded theoretical results. Now that CNNs have revolutionized computer vision, specifically the fields of image classification and object detection, and they are seen as a promising tool to improve these automated systems, allowing for better results, a broader scope of diseases, and the implementation of useful instantaneous plant disease recognition systems.

3. METHODOLOGY

The complete process of the proposed HCRNN-based plant leaf disease detection and classification method is depicted in Fig 1. In-depth explanations of each step taken to create this model for identifying plant diseases with deep CNN are provided. The full procedure is summarized here, beginning with the acquisition of images for classification using deep neural networks. As a result of this research, a four-step method was proposed for detecting diseased leaves: pre-processing, image segmentation, feature extraction, and prediction of disease. At first, AKF is used to do preliminary processing on the gathered raw image in order to clean it up and improve its overall quality. The suggested GSOC is then used to segment the donoised images, and the resulting segments are used as input in the detection phase, where an HCRNN is used to identify the disease. The suggested HCRNN achieved improved performance compared to other plant disease classification schemes, and this is due in large part to the application of AMS to improve the classification accuracy of CRNN by fine-tuning its weights.

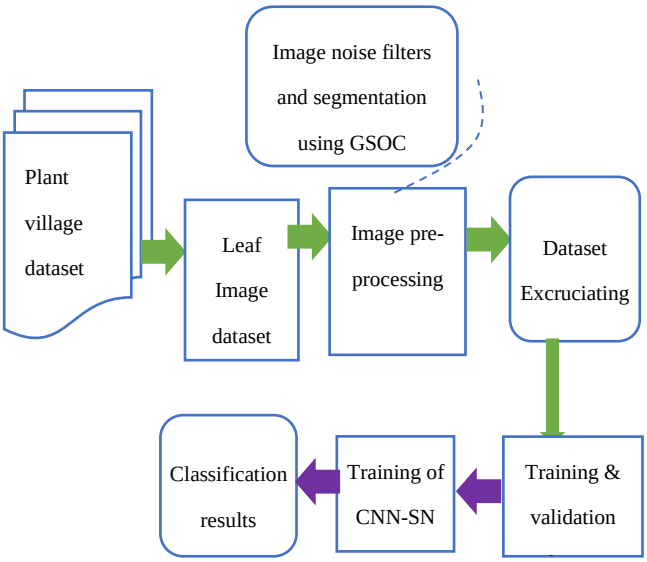

Fig.1. The overall process of Proposed HCRNN Plan Disease Detection

i. Input dataset description

PlantVillage [16], a plant disease dataset produced by Penn State University, has 54,305 RGB photos divided into 38 plant disease classes. There are 14 different plant illustrations inside. Images of healthy and sick leaves, each measuring 256 by 256 pixels, represent at least two distinct classifications for each plant. Fig.2 displays some representative images from the collection. Since this data set became public, numerous studies on the subject of plant disease diagnosis have been done.

Fig.2. Illustrations of some of the 38 different leaf diseases in the Plant Village dataset

ii. Data pre-processing

The utilization of a bilateral filtering technique has been projected as a means to reduce the occurrence of Additive White Gaussian Noise (AWGN) and speckle noise within a dataset of plant village images. The amplification of the noise reduction has enhanced the quality of the image and simplified the process. In this study, the adaptive Kalman filter is employed as a noise-reduction technique for leaf images.

Adapted Kaun filter

In order to mitigate the presence of extraneous noise in plant photos and enhance the accuracy of the village dataset, this method introduces the modified Kuan filter. This filter effectively eliminates noise without compromising the integrity of edges or features within the images. However, the noise is converted from a multiplicative noise model to an image-dependent additive noise model. Subsequently, the model undergoes the application of the least MSE (Mean Square Error) condition in order to estimate the original representation of the image. The pre-processing of the grey-level value P is expressed in the following manner.

$$P_{a,b} = \sum_{a=1}^{r=3,n=3} \sum_{b=1}^{r=3,c=3} C_{a,b} * wf_{a,b} + m_{a,b} * (1 - wf_{a,b}) \quad (1)$$

The rows and columns of the pixel data are represented by the coordinates (r,c), whereas the pixels themselves are represented by the coordinates (a,b) in a picture. The central pixel, or $C_{a,b}$, of the filter window $m_{a,b}$ is the average intensity inside the window. This average is sometimes referred to as the weighting factor $wf_{a,b}$. Weight factor adjustments made with the Random Search Algorithm (RSA) to recover image quality and decrease noise provide the basis for the predicted noise reduction performance. The Adapted Kalman Filter (KF) performs this purpose. An iteration sequence $\{W_k\}$, where k is an index between 0 and k, describes the algorithm under study, which is a general random search algorithm. Previous pixel coordinates and an algorithmic parameter Θ marked by informs these iterations. The mean vector and covariance matrix are used to define the algorithm. Reflecting the probabilistic nature of the RSA algorithm, the current iteration W_k is a set of points, and the iterations are capitalized to denote their status as random variables.

iii. Segmentation using GSOC

Following initial processing, GSOC is used to segment the afflicted area of the plant. The initial number of centroids and premature convergence are two limitations of k-means that are circumvented when swarm intelligence clustering methods are used. As a segmentation solution, the GSOC has been introduced. This method leverages the GSO's multimodal exploration capability to identify optimal centroids, thus incorporating the advantages of GSO for effective segmentation. In addition, the GSOC can figure out how many clusters there are without being given a specific number. The suggested GSOC method is a partitioning-based clustering, with the idea that instances cluster near their centroids, serving as inspiration. As a method for clustering data, K-means uses a weighted average of the instances to determine cluster centroids. If the data set is grouped into regular clusters of the same shape, the weighted average extraction approach may work well; otherwise, it will be inefficient. The suggested method turns the grouping problem into a multimodal optimization problem so that the centres can be found based on how the glowworms move.

Each glowworm in the swarm works to cover more ground in the data set as a whole using the suggested algorithm's clustering technique. Each glowworm, guided by the parameter r_s, also migrates toward other glowworms that cover more data instances and have closer proximity among them in their immediate region. The GSO, which was developed using glowworm behaviour [34], outperforms previous swarm algorithms in terms of coverage rate and works on numerous functions at once. This method chooses the best weights since they have the best performance, and this process typically consists of four stages, including the release of glowworms, an update to the luciferin, some forward motion, and a final decision. The stepwise process is given below.

Step 1: A random point p_j (i.e. cluster fitness) is chosen from the full goal space and used to initialize the glowworms $p_k, k = 1, \ldots, m$. A tent map of chaos n has been incorporated into this phase of GSO for the purpose of improving randomness. Chaos is shown as a non-linear, ubiquitous phenomenon with superior periodicity and randomness. It can enable the glowworms to search for the optimal value accurately and is defined as

$$p_{k+1} = \begin{cases} 2p_k, 0 \le p_k \le 0.5 \\ 2(1 - p_k), 0.5 \le p_k \le 1 \end{cases} \quad (2)$$

Step 2: Each swarm contains the same number of luciferin molecules and a similar range of weights, and the luciferin value has been defined in terms of the objective function value as

$$Lu_i(time + 1) = (1 - \rho)Lu_i(time) + \gamma Ft_i(p_j(time + 1)) \quad (3)$$

Here, ρ represents the luciferin decay constant, taking values within the range of zero to one. γ denotes the luciferin enhancement constant, and $Ft_i(p_j(time + 1))$ represents the objective function value at the position of glowworm i at time $time$. The convergence rate of GSO has been shown to benefit from the introduction of the movement rule. When the number of iterations is greater than 10, the worst-fitting 5% of glowworms are replaced by the average location of all glowworms, allowing for a faster search for the ideal value.

Step 3: Here, a probabilistic process is used to shift the weights in favour of the neighbours whose glowworms have a higher luciferin value. The probability of a glowworm migrating toward a neighbor j, denoted by $p_j(time)$, is defined as

$$pr_j(t) = \frac{(Lu_j(time) * \rho - Lu_i(time) * \rho)}{\sum_{k \epsilon n_i(T)} (Lu_k(time) - Lu_i(time))} \quad (4)$$

Where the luciferin value of glowworm is represented as $lu_i(time)$and the movement of glowworms *i* can be followed as

$$p_i(time+1) = p_i(time) + s \times \left(\frac{\rho * \left(p_j(time) - p_i(time) \right)}{\left\| p_j(time) - p_i(time) \right\|} \right) \quad (5)$$

Where s is the step size.

Step 4: The luciferin rule has been modernized to account for the varying numbers of nearby molecules, and its new definition is as follows:

$$r_d^i(it+1) = min\{r_s, max\{0, r_d^i(it) + \beta(n_{it} - |N_i(it)|)\}\} \quad (6)$$

The local-decision domain of i at the $it+1$ iteration is represented by $r_d^i(it+1)$, where$'\beta'$ is a constant parameter that controls the rate of change of the neighbor domain and n_{it} is a threshold that is used to regulate the number of neighbors. Each image in the plant village data set V has n instances and d dimensions, and these instances are labelled with $p_i, i = 1 \ldots n$, which are then used as inputs to the clustering algorithm. An example of a clustering algorithm is $Cd = \{cd_1; cd_2, \ldots., cd_k\}$ where k is the number of centroids in the cl centroid set, and $Cl = \{cl_1; cl_2, \ldots., cl_k\}$ where a single centroid represents each cluster. The image is segmented to isolate the plant sections in a single shape. Furthermore, the clustering algorithm endeavors to heighten the similarity among examples within the same cluster while diminishing the similarity between instances in distinct clusters. Each cluster must also have at least one instance in it, and the various clusters must be disconnected from one another in a such that $\bigcap_{i,k} cl_i = \{\}$ and $\bigcup_{i,k} cl_i = V$. The Sum Squared Errors (S) is then determined by filling in all of the blanks in each cluster.

$$S = \sum_{j=1}^{k} \sum_{i=1}^{|cl_j|} \left(cd(pv_i, cl_j) \right)^2 \quad (7)$$

Every glowworm in the GSO optimization swarm S is denoted by a vector ($gw_j, j = 1 \ldots m$), with a total of m glowworms in the swarm. The coverage set (cs_j) represents the set of data examples covered by gw_j, and intra-distance (iD_j) signifies the distance between the members of cs_j and the position of gw_j. Each gw_j comprises five components: luciferin level (LL_j), fitness function value (FF_j), d-dimensional position vector (p_j), and cs_j. In its local range, the gw_j should include at least one data instance. All of the glowworms in swarm S have the same radial sensor range; hence, the local range is also r_s. Even if a glowworm has no nearby neighbours or many, it can still move towards the optimal glowworms by maintaining a constant local range during the clustering process. The method above is used to divide up the plant village dataset. Fig. 3 depicts the GSO clustering procedure. Small black crosses represent the initial, unplanned positions of the glowworms, and the red dots represent actual instances of plants in the data set. (ii) How the glowworms moved around when clustering. (iii) Following the clustering procedure with four centroids, the final positions of glowworms (i.e., little squares) are shown, with each cluster in the data set having a different colour (i.e., plant region) based on the smallest distances to the centroid.

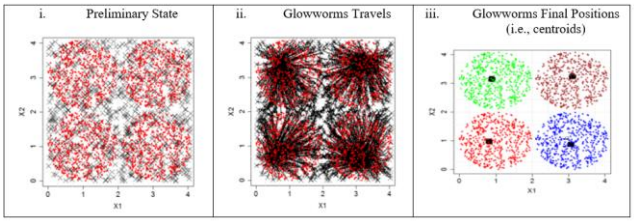

Fig 3: The clustering procedure for the village dataset involves a swarm size of 1, and the maximum number of iterations is set to 200.

iv. Plant disease detection using hybrid CNN and RNN

In this portion, a new automatic hyperparameter selection approach will be proposed for identifying the ideal network configuration, which includes the network topology and hyperparameters for HCRNN by utilizing AMS in conjunction with a leaf gradient descent algorithm. As part of the suggested method, network configurations were encoded as a collection of real-number m-dimensional vectors that the AMS algorithm would use to conduct its search. When using a search agent to train the HCRNN classifier, the AMS algorithm is utilized to discover the optimal settings for the network. In terms of automatic network structure and parameter selection for HCRNN, the suggested method provides an alternate option.

Deep Convolutional neural networks (DCNN)

A CNN has one or more convolutional layers and one or more fully connected layers that connect them. Multiple hidden layers, typically consisting of a succession of convolutional layers, merge the information from the input and output layers. A CNN can spot network traffic anomalies and dangerous software very ease. However, this required a lot of processing time; thus, AMS was used to fine-tune the hyperparameter. For classification, the feature vector learned in the flattened layer in conjunction with softmax activation algorithms employed. Finally, the information gleaned from the network nodes used to place assaults into several categories. The suggested technique uses a CNN because of the CNN's superior performance in feature identification and recognition tasks. DCNN works by chaining together a variety of activation functions, such as. The combination of ReLU and Hard-Swish can produce neural networks with improved performance.

Convolution 1: The array of numerical data that make up the input plant village features can now be fed into a

convolutional layer to undergo further processing. The equation is used to determine the feature matrix F_m.

$$F_m = Ch(ix) \cdot (N_w, \mathfrak{b} \times F_{rw+\mathfrak{w}-1,cl+h-1} + \mathfrak{b}) \quad (8)$$

In this formula, ch signifies the number of channels, rw represents the rows of the matrix, which can take on any value from $1 \leq w \leq l$ and cl represents the columns of the matrix, which can take on any value from $1 \leq h \leq l$. By switching from the ReLU activation function to a non-linear activation function for all activations, the model's convergence rate is stabilized and the vanishing gradient problem is avoided. Here $A(x)$ is the ReLU formula as in Eq. (9):

$$relu(A(ix)) = \begin{cases} 0 & if\ x \leq 0 \\ x & if\ x > 0 \end{cases} \quad (9)$$

In this paper, Hard-Swish (HS), a new and novel activation which is carefully connected to the activation function Swish is introduced. It is defined as in Eq.(10)

$$HS(A(ix)) = 2x \times \max(0, \min(1, (\alpha x \times 0.2 + 0.5))) \quad (10)$$

where α, is a constant or a parameter that can be adjusted through training. After a certain value of $\alpha \to \infty$, the hard-sigmoid part of HS behaves like the ReLU activation function. HS uses a non-linear approach to interpolate between Relu and linear functions smoothly. The level of interpolation can be modified by altering α, a trainable parameter. Consequently, the channel dimension is constructed by stringing together the proposed pair of activation functions (or, more generally, a set of activation functions).

$$Ch(ix) = P([HS(A(ix)), relu(A(x))]) \quad (11)$$

Where P is concentrate.

Max pooling 1: Max pooling can downsample superfluous features to prevent overfitting and lower network parameters. It can be expressed by

$$ix_j^l = A(wt_j^l * d(ix_j^{l-1}) + \mathfrak{b}_j^l) \quad (12)$$

where wt_j^l signifies the weight of the jth feature map in the lth layer and $\mathfrak{b}_j^l$represents the bias in that map. Two branches of the model are used to learn from the T2- and Diffusion-weighted MRI data: one employs a ten-layer CNN Mean, and the greatest, and probabilistic pooling functions are all abbreviations for down-sampling (*) in the field of statistics. The dimension of the final feature map was reduced using max-pooling with shift-invariances.

Convolution 2: After being obtained in the Maxpooling layer, the pooled functional matrix serves as the input for the second convolution layer, which is responsible for the extraction of high-level characteristics from that matrix. The calculation for the second convolution layer is the same as the first (Eq. 8 and 9). The objective of the second max-pooling layer is analogous to that expressed in Equation 13, aiming to reduce the size of the matrix. The outcome of the second max-pooling layer can subsequently serve as a pooled function map for this flattened layer. A pooled plant feature matrix is transformed into a column or feature vector via the flattening layer. Within this layer, the function is restructured in order to transform the features or parts of the pooled feature map Im into feature vectors Iv, as explained below.

$$Q = po.rs\left[\left(relu\left(A(pi)\right) - wt + 1\right) \times \left(HS\left(A(pi)\right) - h + 1\right)\right] \quad (13)$$

Where p is pooling, and rs is reshape.

Classification: Two branches of the model are used to learn from the T2- and Diffusion-weighted MRI data: one employs a ten-layer CNN. Disease categories' probabilities are determined by the use of softmax functions applied to a dense layer of numerous neurons. Consequently, the net output op calculated using (Eq. 14):

$$op_j = \sum_{i=1}^{l} wt_i \cdot pi_i + \mathfrak{b} \quad (14)$$

At the classification layer, the softmax activation functions are used.

Hyperparameter Tuning: The AMS offers an approach that is ideal for searching for parameters in the proposed CNN. Parameters like kernel size, number of steps, number of channels, weight, and bias must be set because it relies on Hyperparameter Tuning. The optimal set of parameters for a model is determined by hyperparameter tuning. Based on the recommended comparison parameters employing AMS and the modified structure referred to as DCNN, Hyperparameter Tuning was chosen for this investigation. The RNN is also a hybrid with the CNN scheme, and its name is HCRNN, to increase the classification accuracy.

The RNN is a three-hidden-layer feed-forward multi-layer perceptron that sits between an input and an output layer. The objective of the RNN training process is to attain a precise reproduction of the input data design at the output layer. The n units in the input layer and the n units in the output layer jointly signify features extracted from the exercise data [35]. The sizes of the three hidden layers are determined in order to minimize the mean rebuilding error transversely all training designs. Specifically, the activation function $A_k(I_{po})$, where I_{po} is the weighted total of the inputs to the unit and is demarcated as the output of unit o in layer p.

$$\mathrm{I}_{po} = \sum_{\mathrm{j}=0}^{\mathrm{L_{p-1}}} \mathrm{w_{poq}} \mathrm{Z_{(p-1)j}} \quad (15)$$

Where Z_{kj} is the productivity from the qth unit of the pth layer. L_p is the number of units in the pth layer and activation function af for the two external hidden layers (p = 2, 4) is then:

$$S_p(I_{po}) = \tanh(a_p I_{po}), p = 2,4 \quad (16)$$

For this experiments, a_p is a tuning parameter set to 1. In the case of the middle-hidden layer (p = 3), the DL is the cascade with N as the number of stages or degrees of activation and a_3 determining the rate of transition between levels.

$$S_3(I_{po}) = \frac{1}{2} + \frac{1}{2(p-1)} \sum_{j=1}^{N-1} \tanh[a_3 (I_{po} - \frac{j}{N})] \quad (17)$$

For the middle-hidden layer, a step-wise af is utilized, which converts the endlessly distributed data points into a set of discrete valued vectors (activation levels): 0, 1/N1, 2N1, etc. This method of data classification is utilized in this work for the purpose of categorizing liver diseases. Data points are organized into groups due to the plotting to the discrete classes in the middle-hidden layer. Further analysis of the RNN's disease predictions can reveal both isolated cases and clusters.

Training the IRNN: The af for the output layer is selected from two possible candidates. The first is linear, and it equals the inputs' weighted total using the formula in Equation 18. This means that $S_5(I_{po}) = I_{po}$. The Sigmoid function comes in at number 2.

$$S_5(I_{po}) = \frac{1}{1 + e^{-a5 I_{po}}} \quad (18)$$

The neural network weights are modified with the use of an adaptive learning rate (l) throughout each training iteration:

$$wt_{oq}^{l+1} = wt_{oq}^{l} + \alpha l + 1 \Delta wt_{oq}^{l+1} \quad (19)$$

The novel learning amount at iteration l + 1, α_{l+1} is specified by:

$$\alpha_{l+1} = \begin{cases} \beta_r \times \alpha l & \text{if } e_{l+1} > 1.01 * Ms_1 \\ \beta_e \times \alpha l & \text{if } Ms_{l+1} < Ms_1 \text{ and } \alpha l < \alpha_{max} \\ \alpha l & \text{otherwise} \end{cases} \quad (20)$$

Where e_l in equation 20 refers to the mean square error (MSE) and calculated as Ms

$$Ms = \frac{1}{mn} \sum_{i=1}^{M} \sum_{j=1}^{n} (x_{oq} - op_{oq}^{l})^2 \quad (21)$$

In Equation 21, the variable "m" denotes the quantity of images within the training dataset, "n" signifies the number of features, and "x_{oq}" represents both the input and the desired output (o = 1, 2.....m, q = 1, 2.....n), and op_{oq}^{l} represents the rate of the RNN's output at the lth iteration, where o and q are the input and target, respectively. Adjustable parameters include the Early Learning Rate, 0, the Extreme Learning Rate, max, the Learning Rate Expansion Factor, β_e, and the Learning Rate Lessening Factor, β_r.

Hyperparameter tunning using AMS

Phototaxis refers to the behaviour of moths, in which they circle and approach a source of artificial light. Various hypotheses have been proposed to explain this behavior, but its precise cause remains a mystery. Using celestial in a transverse orientation when airborne is one of the proposed explanations. Moths, like the moon, will keep a constant angle to the celestial light by flying in a straight line [36]. Moths are characterized in part by their ability to perform Levy's flights, one of the most important flying patterns in their native environments. Levy flights, as seen in insects like the fruit fly Drosophila, resemble a power law distribution over a wide range of scales, with exponents close to 3/2 [37]. Based on the findings presented in reference [38], certain complex flight patterns can be elucidated by adopting an optimal biased scale-free searching strategy to emulate the two primary phases of the MS algorithm, namely exploitation (intensification) and exploration (diversification). In this context, phototaxis and Levy flights inspired by moths in nature were incorporated. Specifically, moths within the population situated in proximity to the most superior moth (or light source) execute Levy flights around the preeminent moth. The ensuing equation encapsulates this behavioral phenomenon:

$$p_i^{t+1} = p_i^t + \alpha LD(s), \quad (22)$$

Two branches of the model are used to learn from the T2- and Diffusion-weighted MRI data: one employs a ten-layer CNN where p_i^{t+1} is the current iteration's moth i's updated location and p_i^t is the original location of moth i in the current generation. The LD(s) notation stands for Levy-drawn steps. The value of the scaling parameter α is a function of the optimization issue. In the unique MS algorithm, α was specified as [36]:

$$\alpha = Smax/t^2, \quad (23)$$

In the equation provided, where $Smax$ represents the maximum step length in the walk, its value is subject to dependence on the specific characteristics of the given problem. LD assumed in Eq. (23) can be articulated as given in Eq. [24]:

$$LD(s) = \frac{(\beta-1)\Gamma(\beta-1)*\sin(\frac{\pi\times(\beta-1)}{2})}{\pi s^{\beta}}, \quad (24)$$

where Γ is gamma function and s is better than 0. Two branches of the model are used to learn from the T2- and Diffusion-weighted MRI data: one employs a ten-

layer CNN The most successful moth in a population, even if it's far from the light, will still fly in a straight line towards it. This process can be described as given eqn

$$p_i^{t+1} = \lambda \times (p_i^t + \varphi \times (p_{best}^t - p_i^t)) \quad (25)$$

where p_{best}^t denotes best moth in generation t and φ and λ are quickening and scale factors, correspondingly. Two branches of the model are used to learn from the T2- and Diffusion-weighted MRI data: one employs a ten-layer CNN The moth can also reach the destination ahead of the best moth in the community by flying toward the source of light. This flight pattern is described as

$$p_i^{t+1} = \lambda \times (p_i^t + \frac{1}{\varphi} \times (p_{best}^t - p_i^t)) \quad (26)$$

The original paper [36] simplifies things by splitting the moth population in half along fitness lines. Moths in the first subpopulation (those with higher fitness) have their positions updated utilizing Levy flights according to eqn.26. In comparison, those in the second subpopulation (those with lesser fitness) have their locations updated according to eqns.27, with a chance of 50%. According to the findings of actual trials and the data presented in [36], the MS algorithm is an effective means of resolving global optimization problems. There were, however, a few problems with the way MS operated.

In later iterations, when the algorithm assumed and has located the proper section of the search space, the current best solution has a large impact on the updated positions of moths in subpopulation 2. However, this often results in inferior mean values and premature convergence in the first few rounds. Subpopulation 2 introduces a third search equation that uses random exploration of the search space to make up for these shortcomings.

$$p_{i,j} = lb_j + rand(0,1) * (ub_j - lb_j), \quad (27)$$

Two branches of the model are used to learn from the T2- and Diffusion-weighted MRI data: one employs a ten-layer CNN where ub_j and lb_j are the upper and lowest limits of the j-th solution variable, respectively, and $p_{i,j}$ is the j-th parameter of the i-th moth in the subpopulation 2. In the second subpopulation, the modified MS (MMS) method can use either eqn (18%), (18%), (20%), or (21%).

4. Experimental Results and Discussions

In this part of the discussion, the performance of the projected HCRNN-based plant disease detection scheme is examined and compared with the performance of existing schemes such as DCNN and OACB. 54,306 pictures; 14 crop species; 26 illnesses (or healthy); 38 classes of healthy and sick leaves are used in this study, all sourced from Plant Village. In this work, the confusion matrix was computed in order to provide a summary of the classification performance of a classifier based on the test dataset. Accuracy, specificity, sensitivity, precision, and the F1 score are only a few of the performance metrics for which it is commonly employed.

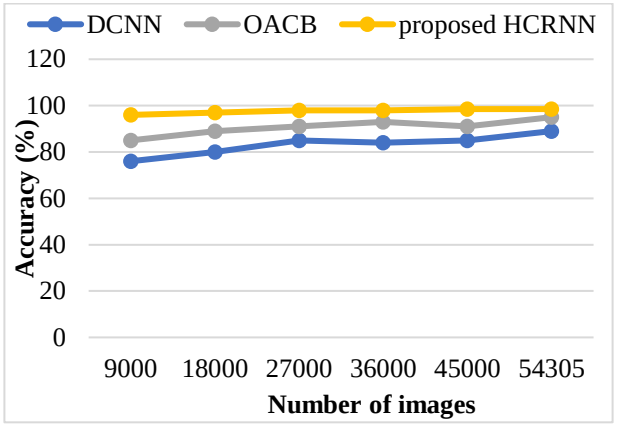

Fig 4: Accuracy performance of proposed HCRNN and other schemes

For a specific number of features in a database, Fig. 4 displays the accuracy of the proposed HCRNN in comparison to the existing models of OACB and DCNN. The HCRNN enhances precision and speeds up computations. Given that the HCRNN does not require a substantial number of derived components during reduction, it attains an accuracy of 98.5%, surpassing all other models in comparison. Two branches of the model are used to learn from the T2- and Diffusion-weighted MRI data: one employs a ten-layer CNN. The HCRNN method proposed in this study outperforms existing methods in terms of favorable validation outcomes for disease prediction. This superiority stems from its enhanced capacity to learn and adapt to illness and leaf image data by incorporating texture features. Moreover, the proposed method successfully mitigates the risk of overfitting in small datasets.

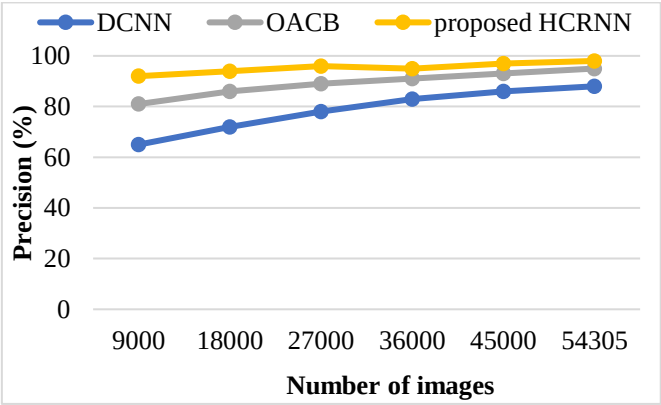

Fig 5: Precision performance of proposed HCRNN and other schemes

The accuracy of HCRNN in comparison to the OACB and DCNN models already in use is shown in Fig. 5. As the number of characteristics grows, so does the accuracy. The HCRNN, in comparison to the DENN and the ECPRC, achieves a recall of 98%. HCRNN decreases the time required computing derived factors, making network fine-tuning easier and more accurate. The most important

details can be zeroed in on and duplicated work can be avoided by employing a unique loss method for each image representation. Therefore, the efficacy of disease classification and prediction is enhanced by this approach.

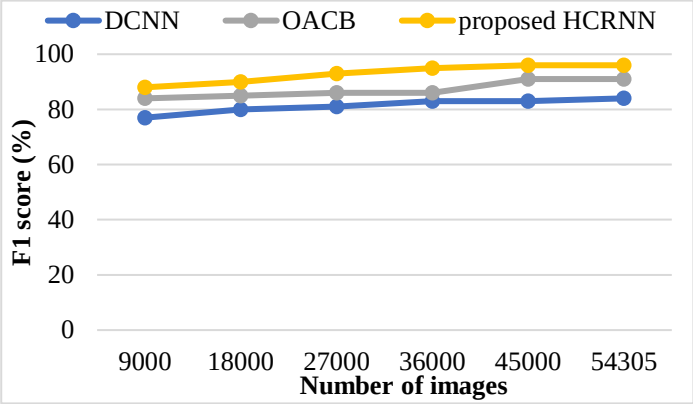

Fig 6: F1-score performance of proposed HCRNN and other schemes

Compared to the existing models of OACB and DCNN, the F1 score of the proposed HCRNN for the number of features in the given databases is shown in Fig. 6. The f-measure is optimized along with the number of characteristics. When compared to other models like OACB and DCNN, for instance, the f-measure provided by the HCRNN is 96%. In terms of accuracy and F1 score, the HCRNN was clearly superior to the alternatives. These processes, when combined with FMSO, greatly enhance the quality of the hyperparameters of AMSs while reducing the computational cost, allowing for a high F1 score to be attained.

Compare the recall of the proposed HCRNN to that of the existing models of OACB and DCNN for a given database size in Fig. 7. The recall improves linearly with the number of characteristics. When compared to the OACB and DCNN, the HCRNN achieves a recall of 98.5%. Existing approaches are underfitting because they rely on simplistic models that perform poorly on high-dimensional data. The suggested method's specificity will be enhanced once the optimal hyper-parameter configuration has been discovered in AMS and applied to the complete training of a CRNN, leading to a much-decreased error rate.

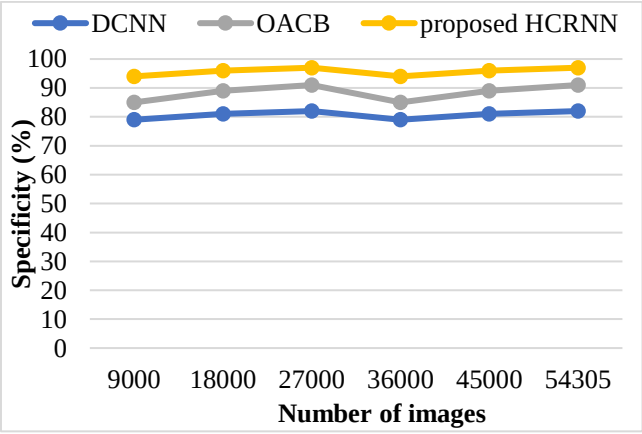

Fig 7: Specificity performance of proposed HCRNN and other schemes

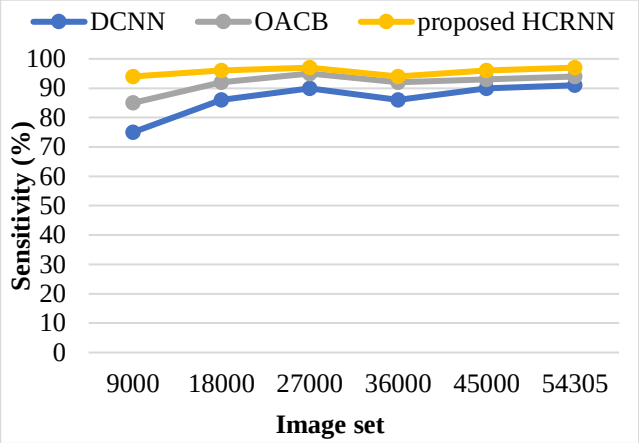

Fig 8: Sensitivity performance of proposed HCRNN and other schemes

Utilizing the identical database and an equivalent number of features, Fig 8 juxtaposes the accuracy of the proposed HCRNN against that of the existing models OACB and DCNN. When compared to the OACB and DCNN, the HCRNN improves accuracy and eventually reaches a rate of 98%. As a result, the suggested method outperforms the state-of-the-art algorithms at validating citrus fruit and leaf health. Since different RNN layers have variable potential in boosting sensitivity, the proposed HCRNN's ability to optimize individual CNN layers is helpful for achieving greater performance at a lower cost.

5. Conclusion

The model utilizes two branches to learn from T2- and Diffusion-weighted MRI data. One branch employs a ten-layer CNN, considering images of both healthy and infected leaves from the leaf infection database. This research introduces a framework for real-time leaf disease detection based on the HCRNN approach. The image pre-processing approaches employed with the AK fileting method to expand the dataset and make the suggested system more resilient. The GSOC segmentation is then used to isolate the damaged area of the leaf picture. The efficiency and generalizability of the model are both improved by this method. The findings of the planned HCRNN study are extremely encouraging for identifying healthy and sick leaf types in the plant village dataset. When compared to other algorithms like OACB and DCNN, the suggested HCRNN performed exceptionally well, with an F1 score of 98% and accuracy, specificity, sensitivity, and precision, all in the 98% range. However, more research is needed to segment the sick parts of the leaf image in the future by reducing background noise. Current leaf disease diagnosis techniques are developed with photos acquired in a controlled environment, such as a laboratory. Even though real-time disease recognition architecture based on photos of leaves taken in the field has been established, the suggested system is not yet fully automated. Additionally, a multi-object deep learning model will be developed to detect plant diseases from a cluster of leaves rather than focusing solely on a single

leaf. This approach aims to address challenges associated with real-time data collection in the context of plant disease detection. Additionally, efforts are being made to put the trained model developed here into a mobile app.

Author contributions

Name1 Surname1: Conceptualization, Methodology, Software, Field study **Name2 Surname2:** Data curation, Writing-Original draft preparation, Software, Validation., Field study **Name3 Surname3:** Visualization, Investigation, Writing-Reviewing and Editing.

Conflicts of interest

The authors declare no conflicts of interest.

References

[1] Vishnoi, V. K., Kumar, K., & Kumar, B. (2021). Plant disease detection using computational intelligence and image processing. *Journal of Plant Diseases and Protection*, *128*, 19-53.

[2] Benjamin.N (2021), Precision agriculture: Where do we stand? A review of the adoption of precision agriculture technologies on field crops farms in developed countries. Agric. Res. 2021;10:515–522

[3] Xiao, Qingxin, et al.. (2019). Occurrence prediction of pests and diseases in cotton on the basis of weather factors by long short term memory network. BMC bioinformatics, 20, 1-15.

[4] Dhaka, Vijaypal Singh, et al. (2021) A survey of deep convolutional neural networks applied for prediction of plant leaf diseases. Sensors. 2021;21:4749.

[5] Nidhi, K., Rani, G and Dhaka. VS. (2020). A comparative analysis of deep learning models applied for disease classification in bell pepper; Proceedings of the 2020 Sixth International Conference on Parallel, Distributed and Grid Computing (PDGC); Waknaghat, India. 6–8 November 2020; pp. 243–247.

[6] Ankita, G., and Rani., G. (2023) Detecting Tomato Crop Diseases with AI: Leaf Segmentation and Analysis; Proceedings of the 2023 7th International Conference on Trends in Electronics and Informatics (ICOEI); Tirunelveli, India. 11–13 April 2023; pp. 902–907.

[7] Savla, D., Dhaka, V. S., Rani, G., & Oza, M. (2022, February). Apple Leaf Disease Detection and Classification Using CNN Models. In *International Conference on Computing in Engineering & Technology* (pp. 277-290). Singapore: Springer Nature Singapore.

[8] García, L., Parra, L., Jimenez, J. M., Lloret, J., & Lorenz, P. (2020). IoT-based smart irrigation systems: An overview on the recent trends on sensors and IoT systems for irrigation in precision agriculture. *Sensors*, *20*(4), 1042.

[9] Chen, C. J., Huang, Y. Y., Li, Y. S., Chang, C. Y., & Huang, Y. M. (2020). An AIoT based smart agricultural system for pests detection. *IEEE Access*, *8*, 180750-180761.

[10] Dankhara, F., Patel, K., & Doshi, N. (2019). Analysis of robust weed detection techniques based on the Internet of Things (IoT). *Procedia Computer Science*, *160*, 696-701.

[11] Karine CD, Salgado AC, and Loscio BF. (2021). A survey on data fusion: What for? in what form? what is next? J. Intell. Inf. Syst. 2021;57:25–50.

[12] Nidhi K, et al. (2021). Iot and interpretable machine learning based framework for disease prediction in pearl millet. Sensors. 2021;21:5386.

[13] Mulham F, et al. (2019). Crop and Weeds Classification for Precision Agriculture Using Context-Independent Pixel-Wise Segmentation; Proceedings of the 3rd IEEE International Conference on Robotic Computing, IRC 2019; Naples, Italy. 25–27 February 2019; pp. 146–152. [CrossRef]

[14] Steven CW., et al. "Counting apples and oranges with deep learning: A data-driven approach." *IEEE Robotics and Automation Letters* 2.2 (2017): 781-788.

[15] Petteri N, Narra N, and Lipping T. (2019) "Crop yield prediction with deep convolutional neural networks." *Computers and electronics in agriculture* 163 (2019): 104859.

[16] Deepak S, et al. "AI-based yield prediction and smart irrigation." *Internet of Things and Analytics for Agriculture, Volume 2* (2020): 155-180.

[17] Andreas K, and Francesc X. P. "Deep learning in agriculture: A survey." *Computers and electronics in agriculture* 147 (2018): 70-90.

[18] Mohsen A, et al. "Wheat leaf rust detection at canopy scale under different LAI levels using machine learning techniques." *Computers and Electronics in Agriculture* 156 (2019): 119-128.

[19] Kasinathan, T., & Uyyala, S. R. (2021). Machine learning ensemble with image processing for pest identification and classification in field crops. Neural Computing and Applications, 33, 7491-7504.

[20] Lee, S. H., Goëau, H., Bonnet, P., & Joly, A. (2020). New perspectives on plant disease characterization based on deep learning. Computers and Electronics in Agriculture, 170, 105220.

[21] 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.

[22] Alibabaei, K., Gaspar, P. D., Lima, T. M., Campos, R. M., Girão, I., Monteiro, J., & Lopes, C. M. (2022). A review of the challenges of using deep learning algorithms to support decision-making in agricultural activities. Remote Sensing, 14(3), 638.

[23] Mourikis, A. I., Kalamatianos, R., Karydis, I., & Avlonitis, M. (2021). A survey on the use of the internet of multimedia things for precision agriculture and the agrifood sector. Engineering Proceedings, 9(1), 32.

[24] Sowmiya, M., & Krishnaveni, S. (2023). IoT enabled prediction of agriculture's plant disease using improvedϖ quantum whale optimization DRDNN approach. Measurement: Sensors, 100812.

[25] Kondekar, V. H., & Bodhe, S. K. (2023). Automation in plant pathology: Optimized Attentional Capsule_BiLSTM optimized with chaotic sparrow algorithm for colour feature-based plant disease detection. Multimedia Tools and Applications, 1-34.

[26] Saberi Anari, M. (2022). A hybrid model for leaf diseases classification based on the modified deep transfer learning and ensemble approach for agricultural aiot-based monitoring. Computational Intelligence and Neuroscience, 2022.

[27] Al-bayati, J. S. H., & Üstündağ, B. B. (2020). Evolutionary feature optimization for plant leaf disease detection by deep neural networks. International Journal of Computational Intelligence Systems, 13(1), 12.

[28] Orchi, H.; Sadik, M.; Khaldoun, M. On using artificial intelligence and the internet of things for crop disease detection: A contemporary survey. Agriculture 2022, 12, 9.

[29] Mishra, M.; Choudhury, P.; Pati, B. Modified ride-NN optimizer for the IoT based plant disease detection. J. Ambient. Intell. Humaniz. Comput. 2021, 12, 691–703.

[30] Dataset, P. PlantVillage dataset, 2018.

[31] Hu, W.J.; Fan, J.; Du, Y.X.; Li, B.S.; Xiong, N.; Bekkering, E. MDFC-ResNet: An Agricultural IoT System to Accurately Recognize Crop Diseases. IEEE Access 2020, 8, 115287–115298.

[32] Garg, D.; Alam, M. Deep learning and IoT for agricultural applications. In Internet of Things (IoT): Concepts and Applications; Springer: Berlin/Heidelberg, Germany, 2020; pp. 273–284.

[33] Zhao, Y.; Liu, L.; Xie, C.; Wang, R.; Wang, F.; Bu, Y.; Zhang, S. An effective automatic system deployed in agricultural Internet of Things using Multi-Context Fusion Network towards crop disease recognition in the wild. Appl. Soft Comput. J. 2020, 89.