ISEIS

Journal of Environmental Informatics 42(1), 25-38 (2023)

Journal of Environmental Informatics

www.iseis.org/jei

Rice Plant Leaf Disease Detection and Classification Using Optimization Enabled Deep Learning

T. Daniya[1, 2 *] and S. Vigneshwari[1]

[1] *Department of Computer Science and Engineering, Sathyabama Institute of Science and Technology, Chennai 600119, Tamilnadu, India*
[2] *Department of Information Technology, GMR Institute of Technology, Rajam 532127, Andhra Pradesh, India*

Received 27 November 2021; revised 18 May 2022; accepted 27 June 2022; published online 15 February 2023

ABSTRACT. An automatic identification and classification of rice diseases are very important in the domain of agriculture. Deep learning (DL) is an effective research area in the identification of agriculture pattern identification where it can effectively resolve the issues of diseases identification. In this paper, a hybrid optimization algorithm is developed to categorize the plant diseases. The pre-processing is made using Region of Interest (ROI) extraction and the input image is created by combining the Rice plant dataset, and Rice disease dataset. The segmentation is accomplished using Deep fuzzy clustering. The features, like statistical features, entropy, Convolutional Neural Network (CNN) features, Local Optimal-Oriented Pattern (LOOP), and Local Gabor XOR Pattern (LGXP) is considered for extracting the appropriate features for further processing. The data augmentation is employed to enlarge the volume of extracted features. Then, the first level classification is made by deep neuro-fuzzy network (DNFN), which is trained using Rider Henry Gas Solubility Optimization (RHGSO) that categories into healthy and unhealthy plants. The RHGSO is the integration of Rider Optimization Algorithm (ROA) and Henry gas solubility optimization (HGSO). After that, second-level classification is made by a Deep residual network (DRN) that is tuned by RHGSO. Thus, the RHGSO-based DRN categorizes the unhealthy plants into Bacterial Leaf Blight (BLB), Blast, and Brown spot. Thus, the implementation of the proposed RHGSO-based deep learning approach offered better accuracy, sensitivity, specificity, and F1-score of 0.9304, 0.9459, 0.8383, and 0.9142.

Keywords: deep residual network, texture features, neural network, henry gas solubility optimization, local optimal-oriented pattern

1. Introduction

Rice is one among the significant food in all over the world, but the diseases in rice plants degrade the yield production (Liang et al., 2019). Moreover, plant diseases are a major cause of food security, which affects the food production (Brahimi et al., 2017; Barbedo, 2018; Miao et al., 2019; Li et al., 2020). With the tremendous improvement of computer vision and application, farmers can consult the plant pathologists using the image data-base of rice plants for predicting the diseases of rice plants (Dong and Wang, 2013; Khairnar and Dagade, 2014; Zhang and Yang, 2014; Lu et al., 2017).

1.1. Significance

The revealing and recognition of rice malady in its starting stage can preserve the growth of the plant that produced high yield with better quality of rice. In advanced agriculture, it is very vital to manage plant diseases with less damage to the plants using effective methods (Abed-Ashtiani et al., 2018; Liu et al., 2018). Recently, advanced plant monitoring techniques are developed by combining computer-aided methods with crop image to monitor plant growth (Karmokar et al., 2015; Sabrol and Satish, 2016; Liu et al., 2018). In order to enhance the effectiveness of plant diseases prediction, an automatic prediction and categorization of diseases depending on the DL method provides a better solution. Some of the automatic diseases prediction approaches are the pattern identification method (Phadikar et al., 2013), SVM (Niu and Suen, 2012), computer vision (LeCun et al., 2015), and image processing methods (Khush, 2005). An automatic rice diseases identification model provides the information for diseases control, which reduces the economic loss, usage of pesticides and improves the quality of yield (Bojja and Ambati, 2020; Bojja et al., 2020). In the automatic diseases prediction model, the classification of diseases type is challenging due to the complexity (Babu and Rao, 2015; Liang et al., 2019). Recently, DL approaches are mostly employed for plant diseases prediction and classification, due to its effective performance. DL algorithms produce an output through the multiple layers by extracting the complicated features of leaf images (Bhagawati et al., 2015; Lu et al., 2017; Anandkumar, 2020).

1.2. Previous Studies

The research papers are taken based on rice leaf disease classification in recent years. Rahman et al. (2020) developed the Deep CNN for the recognition of rice diseases. In this meth-

* Corresponding author. Tel.: +91 9095051376; fax: 08941-251591.
E-mail address: daniinfotech@gmail.com (T. Daniya).

ISSN: 1726-2135 print/1684-8799 online
 doi:10.3808/jei.202300492.

od, the InceptionV3 and VGG16 were utilized in the large-scale identification of rice plant infection and two-stage small CNN was used in the small-scale rice plant detection. This method offered good precision and high accuracy for both large and small datasets. However, the VGG16 has more parameters, which may affect the performance of the system. Lu et al. (2017) developed the Deep CNN model for the recognition of diseases in the rice. This method was efficiently classified ten various rice plant diseases and offered an accuracy of 95.48%. The major advantages of this method were good training performance, recognition ability, and convergence rate. The analysis of this method was done with 500 natural images of rice leaves, but for effective performance, it required a large dataset. Chen et al. (2020) developed the Deep transfer learning model for identifying the diseases in the rice plants. This method had the capability to classify multiple kinds of diseases. This method integrates the merits of both ImageNet and DenseNet, which improved the feature extraction capability and minimizes the computational complexity. Liang et al. (2019) developed the Deep CNN for the recognition of rise blast disease. In this method, a dataset was established with the help of plant protection experts, which contains 2902 negative and 2906 positive images. It had good reliability and accuracy. Shrivastava et al. (2019) developed an image-based machine learning approach for classifying leaf disease of rice plants. Here, the analysis was done with 619 images. The deep CNN was used as a feature extraction and the classification is made by SVM. This method was capable of classifying the disease at an early stage. Sethy et al. (2020) developed the SVM model for identifying the leaf disease of rice plants. In this method, the analysis was done with 5932 images and it was provided better performance by combining with the transfer learning technique. However, the F1-score of this method was low. Krishnamoorthy and Parameswari (2018) developed the combination of transfer learning and Deep CNN for an early detection of leaf disease of rice plants. This method identified multiple kinds of diseases from rice plant. This method reduced the time of training and boosts the neural networks' functional capabilities. Jiang et al. (2020) developed the DL and SVM approaches to categorize the four-leaf diseases. This method was processed with thousands of diseases in the rice plant and offered good accuracy. However, the selection of the optimal neurons and layers was difficult in this method.

1.3. Shortcomings

Rice is a significant food crop, which is damaged due to the attack of various pests and diseases. Thus, the revealing and recognition of rice malady in its starting stage can preserve the growth of the plant that produced high yield with better quality of rice. Some of the issues faced by the conventional approaches are listed below:

- Some traditional approaches have high computational cost, time-consuming, and needs the expert's advice.
- Some methods failed to segment the affected portion accurately with heterogeneous background images.
- Some important features are not considered for the evaluation, so that the accuracy of the model gets affected.

These shortcomings in the conventional approaches are considered as a motivation, a novel technique named RHGSO-based DRN is developed for rice plant diseases detection and classification.

1.4. Research Objectives

The research objectives of the implemented plant leaf disease detection technique are discussed below:

- To develop two-level plant diseases classification approaches, namely RHGSO-based DNFN and RHGSO-based DRN.
- To perform the first level classification process using DNFN in order to classify the healthy and unhealthy plants.
- To perform the second level classification using DRN, which classify the unhealthy images as either BLB disease or Blast disease, or Brown spot disease.
- To train the DNFN and DRN classifier using the RHGSO algorithm that is designed by the incorporation of the HGSO algorithm and ROA.

2. Developed Approach for Rice Leaf Diseases Detection and Classification

This section depicts the developed optimization-based DL method to find and classify the rice leaf diseases. Initially, the pre-processed is performed in the input images and forwarded to the segmentation phase. Here, the segmentation is made by deep fuzzy clustering to perform the segmentation of the diseased area from images. Then, the statistical features, entropy features, CNN features, LOOP features, and LGXP features are taken from the segmented output. After the mining of features, then the size of extracted features is enlarged using the data augmentation phase. After that, the first level classification process is performed using RHGSO-based DNFN in order to classify the plant is either healthy or unhealthy. Subsequently, the unhealthy plants are further classified using DRN for classifying the leaf diseases, like BLB, blast, and brown spot. Figure 1 shows the block diagram of devised RHGSO-based DL.

2.1. Getting Input Data

In this paper, the input data is gathered from two datasets, namely rice plant dataset (https://www.kaggle.com/rajkumar898/rice-plant-dataset) and rice disease dataset (https://github.com/aldrin233/RiceDiseasesDataSet). The Rice plant dataset contains 1006 images of rice seeds with healthy and unhealthy images. In this dataset, the count of healthy images is 501, and the unhealthy images count is 506. The rice disease dataset contains images of three diseases, like BLB, brown spot, and blast. This paper considered these two datasets and then merged them into a single dataset for performing disease classification.

The rice plant dataset D contains n images:

$$D = \{D_1, D_2, \ldots, D_x, \ldots, D_n\} \tag{1}$$

where $D_x \in N^{u \times v}$, $x = \{1, 2, \ldots, n\}$, n signifies all images and $[u \times v]$ denotes the dimension of every image.

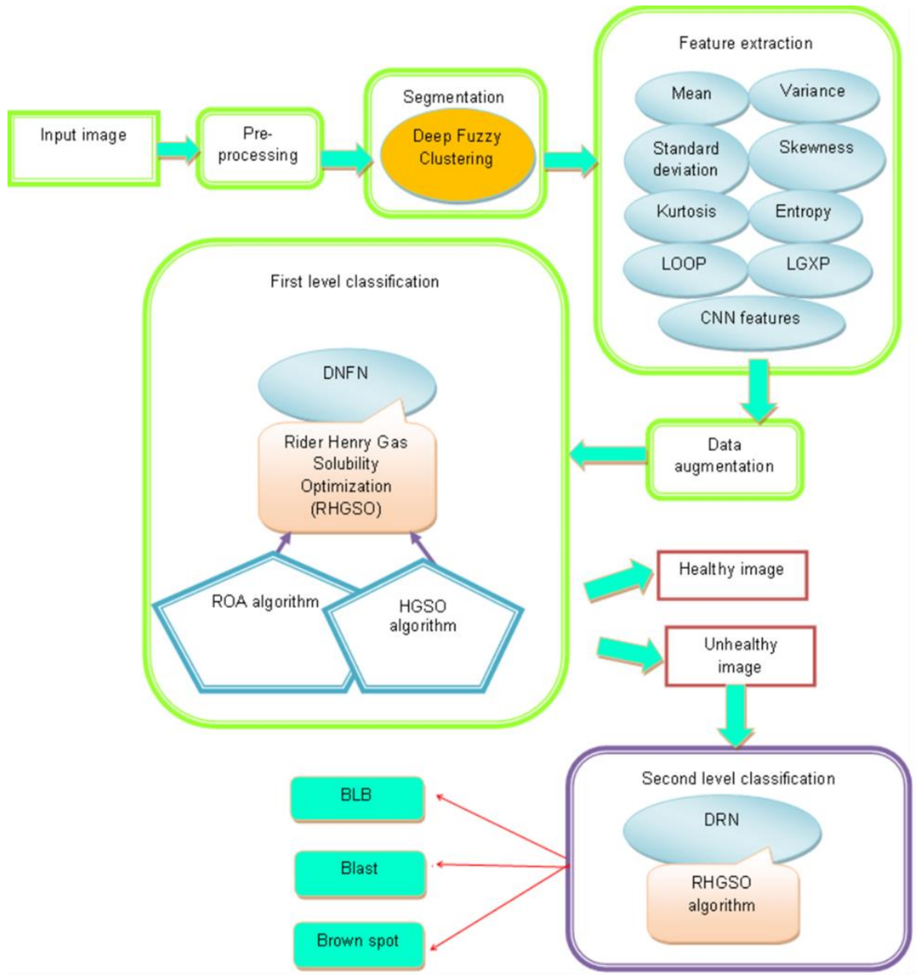

Figure 1. Proposed RHGSO-based Deep learning algorithm for rice plant leaf diseases classification.

The rice disease dataset B contains z count of input images, which is expressed as:

$$B = \{B_1, B_2, \ldots, B_y, \ldots, B_z\} \tag{2}$$

Here, $y = \{1, 2, \ldots, z\}$ and z depicts the total quantity of images with the dimension of $[a \times b]$, respectively.

After the gathering of two datasets, it is required to be combined into a single database, which contains multiple images, and its mathematical depiction is expressed as:

$$M = \{D, B\} \tag{3}$$

$$M = \{D_1, D_2, \ldots, D_x, \ldots, D_n\}\{B_1, B_2, \ldots, B_y, \ldots, B_z\} \tag{4}$$

where M depicts the database of combining two datasets.

2.2. Pre-Processing

The selected M database is forwarded to the pre-processing module for removing the unwanted noise from the images that exists in the dataset using the ROI extraction process. The purpose of ROI extraction method is to create a noise-free image for enhancing performance. Thus, the pre-processed noise-free dataset is depicted as f. The next step is the segmentation process, which segments the affected portion from a noise-free image f using deep fuzzy clustering.

2.3. Segmentation Using Deep Fuzzy Clustering

After pre-processing, the noise-free dataset f is fed into the input of deep fuzzy clustering (Feng et al., 2020) to segment the affected area from rice leaf images. The deep fuzzy clustering effectively handled the complex latent high-dimensional data distribution. This process can help to ease the process of further processing.

Deep fuzzy clustering: The input is $f = \{f_1, f_2, \ldots, f_y, \ldots, f_z\}$, count of clusters U, batch size z_t, and maximum iteration S_{max}, correspondingly. The auto-encoder's objective function is given as:

$$F(f, \delta) = \frac{1}{z}\sum_{y=1}^{z}\left\|ERR_{P,U}(f_y) - f_y\right\|^2 + \gamma \cdot reg(P) \tag{5}$$

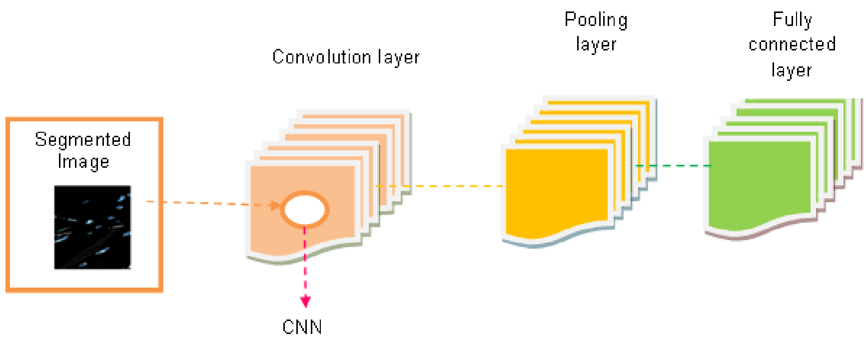

Figure 2. Extraction of CNN features.

where $\| \|$ signifies Euclidean norm, $reg(P)$ represents term of regularization for minimizing the overfitting.

Let us calculate FM_h fuzzy memberships, TR_h target, and the objective function using KL-divergence. The graph regularization is identified using A_f and H_h:

$$\min \gamma_u = \min \sum_{y,\tau=1}^{z} \left\| u_y - u_\tau \right\|^2 A_{f\ y,\tau} \tag{6}$$

where affinity in-between data f_y, and f_τ is signified as A_f. The clustering loss function is signified as:

$$L_f = \sum_{y=1}^{z} \left\| ERR_{P,U}\left(f_y\right) - f_y \right\|_2^2 + \mu_1 \sum_{y=1}^{z} \sum_{f=1}^{cc} J_{if} \log \frac{J_{if}}{K_{if}} + \mu_2 \sum_{y,\tau=1}^{z} \left\| u_y - u_\tau \right\|^2 A_{f\ y,\tau} \tag{7}$$

Finally, the segmented region is signified as:

$$s = \left\{ s_1, s_2, \ldots, s_z \right\} \tag{8}$$

where P represents the weight of autoencoder, U represents the bias of autoencoder, μ_1 and μ_2 represents the hyperparameter, f_y be the input data, u_y and u_τ depicts the hidden features, J_{if} and K_{if} depicts the fuzzy membership, s denotes a segmented region, and z denotes the total number of segments, respectively.

2.4. Extraction of Features

After the segmentation is done, the feature mining process is performed for deriving the powerful features, like statistical features, CNN features, LOOP, and LGXP features. The segmented output s_z is given for mining the relevant features.

2.4.1. Statistical Features

The statistical features are variance, mean, kurtosis, skewness, and standard deviation. Mean is computed by taking mean image pixels, which is given by:

$$C_s = \frac{1}{\left| c\left(s_z\right) \right|} \times \sum_{z=1}^{\left| c\left(s_z\right) \right|} c\left(s_z\right) \tag{9}$$

where z indicates the total segments, $c(s_z)$ indicates every segment pixel value, and all pixels within a division is denoted by $\left| c\left(s_z\right) \right|$, and C_s depicts the value of the mean feature.

Variance is employed to remove the inappropriate features. It is measured by the value of mean feature:

$$V_s = \frac{\sum_{z=1}^{\left| c\left(s_z\right) \right|} \left| s_z - C_s \right|}{c\left(s_z\right) - 1} \tag{10}$$

where V_s specified the variance feature.

Standard deviation denotes the square root of the mean value, which is termed as S_s. Skewness illustrates the object shape with regards to mathematical value with its relative symmetry. It is indicated as K_s. Kurtosis is the peak value irregularity and is expressed as the relative peak value of probability density function, and it is denoted as N_m. The statistical features are given as $E_f = \{C_s, V_s, S_s, K_s, N_s\}$ with dimension of [5 × 1].

2.4.2. Entropy

Entropy discovers the data content that exists in an image. The edge pixels details are selected for calculating entropy measures and are indicated as E_s.

2.4.3. CNN Feature

CNN feature is mined from the segmented region to perform the classification procedure. The CNN feature is represented as Q_s. The feature mining is given in Figure 2.

2.4.4. Loop

The non-linear combination of LBP and LDP is called as LOOP that overwhelms the disadvantages of these two while conserving the strengths of each. Thus, it encrypts rotation invariance to the significant formulation. The expression for the LOOP feature (Chakraborti et al., 2018) is expressed as follows:

$$LOOP(i_s, j_s) = \sum_{n-1}^{N} L(x_n - x_s) \cdot 2^{r_n} \tag{11}$$

where x_n depicts the intensity values, r_n depicts the exponential value and x_s indicates the cluster center, and $L(i)$ is expressed as:

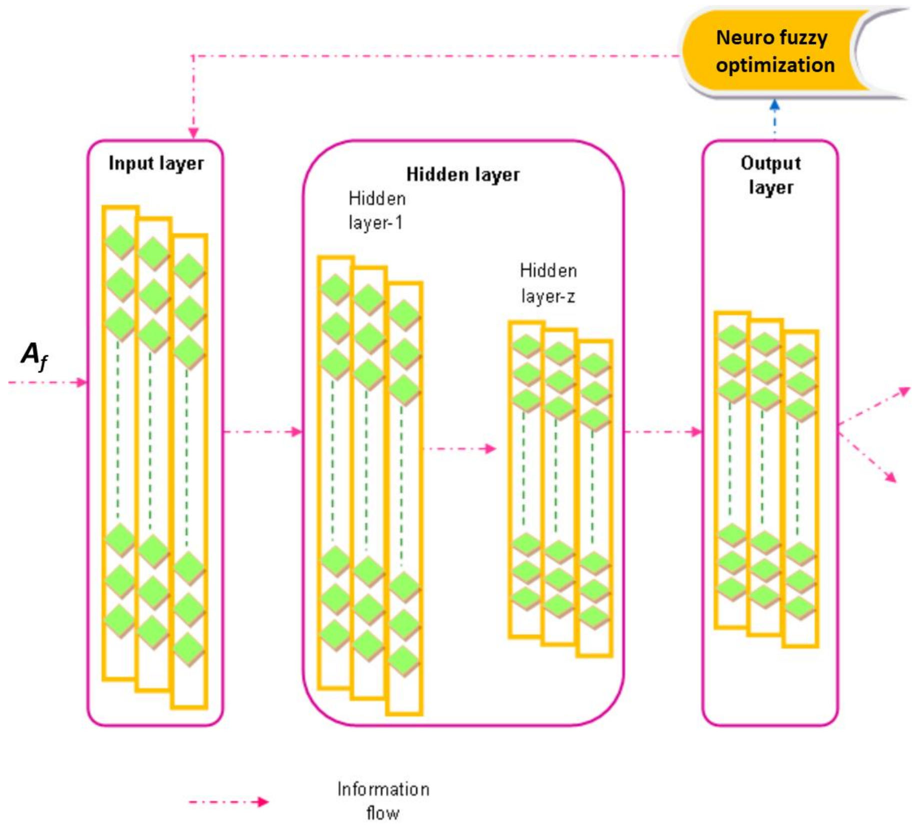

Figure 3. Structural design of DNFN classifier.

$$L(i) = \begin{cases} 1 & \text{if } i \geq 0 \\ 0 & \text{otherwise} \end{cases} \tag{12}$$

The extracted LOOP feature is represented as L_s.

2.4.5. LGXP Feature

The LGXP feature is the combination of the XOR pattern or LXP and Gabor filter coefficient. The LGXP is formed by subjecting LXP to the phase of the Gabor filter coefficient. The LXP is employed to encode the Gabor phase and is represented as X_s.

Finally, the feature vector made using the mined features is given as E_f and given in Equation (13):

$$D_f = \{E_f, E_s, Q_s, L_s, X_s\} \tag{13}$$

Thus, D_f is the all feature vector given to the data augmentation phase.

2.5. Data Augmentation

The extracted feature D_f is passed this phase to increase the size of extracted features. Data augmentation is an integral approach in DL since the DL process needs a large size of features for effective processing. The output of the data augmentation process is represented as A_f.

2.6. Rice Plant Leaf Diseases Classification

Once the data augmentation is done, then the leaf diseases classification process is initialized. The first level and second level classification algorithms and its training process are explained in the following section.

2.6.1. First Level Classification

In this, the augmented feature A_f is given to the input of the DNFN classifier to categorize the given leaf image as either healthy or unhealthy. The structural design of DNFN is explained as follows.

2.6.1.1. DNFN Architecture

The DNFN (Javaid et al., 2019) is the hybridized model of fuzzy logic and Deep Neural Network. The basic parameters involved in the DNFN are premises and consequents. The premises rely on membership function and the consequents are lies on the defuzzification process. In this classifier, the high-dimensional data is handled with better accuracy. The structural design of DNFN for rice leaf plant diseases classification is represented in Figure 3.

z and a are assumed as premises and consequent that is formulated as:

$$K_{1,\alpha} = qR_{\alpha-2}(a), \forall \alpha = 1,2,3 \quad (14)$$

where a and z indicate all α^{th} entity input, qB_α and $qR_{\alpha-2}$ denotes membership function antecedent, and $K_{1,\alpha}$ represents membership degree function, which is formulated by:

$$qB_\alpha(z) = \frac{1}{1 + \left| \frac{z = A_\alpha}{I_\alpha} \right|^{2Q_\alpha}} \quad (15)$$

where Q_α, A_α and I_α is a premise constraint membership function that is trained for optimization.

Similarly, layer 2 is engaged for expressing rule clusters. The membership variable multiplication value signifies the rule strength:

$$K_{2,\alpha} = \psi_\alpha = qB_\alpha(z)qR_{\alpha-2}(a), \ \forall \alpha = 1,2 \quad (16)$$

In addition, layer 3 denotes the normalization that the whole entity estimates the proportion to α^{th} rule strength the addition of every rule strength. Therefore, the generic network parameter is denoted as ψ_α. Thus, each rule result is normalized with rule firing strength, and is expressed as:

$$K_{3,\alpha} = \bar{\psi}_\alpha = \frac{\psi_\alpha}{\psi_1 + \psi_2}, \forall \alpha = 1,2 \quad (17)$$

The fourth layer is represented as the defuzzification layer wherein all rule consequents are predicted for indicating an output:

$$K_{4,\alpha} = \bar{\psi}_\alpha W_\alpha = \bar{\psi}_\alpha \left(Y_\beta z + M_\alpha a + C_\alpha \right), \ \forall \alpha = 1,2 \quad (18)$$

where Y, M and C is a consequent constraint set. The last layer is termed as the summation layer that performs the prior layer summation results. Final result estimation is denoted as:

$$K_{5,\alpha} = \sum_\alpha \bar{\psi}_\alpha W_\alpha = \frac{\sum_\alpha \psi_\alpha W_\alpha}{\sum_\alpha \psi_\alpha} \quad (19)$$

Here, the developed RHGSO used to train the parameters of the approach. Additionally, all hidden layers are employed to make an efficient training system for large data. The DNFN outcome is depicted as Y_δ.

2.6.1.2. Proposed RHGSO-Based Training for DNFN

The developed RHGSO algorithm is developed for training the DNFN algorithm. The RHGSO algorithm is designed by the integration of ROA (Binu and Kariyappa, 2018) and HGSO Algorithm (Hashim et al., 2019). Here, the updated rule of ROA algorithm is applied to the best location of HGSO algorithm for achieving better performance.

Step 1: Initialization. The amount of gas (population size Y) and location of gases are initialized using the expression given below:

$$R_a(k+1) = R_{\min} + r \times (R_{\max} - R_{\min}) \quad (20)$$

where R_a depicts the location of a^{th} gas in Y population, r is the random number among 0 and 1, R_{min} and R_{max} be the boundary limit of the issue, k is the time of iteration. Let e be the amount of gas, $E_{a,b}$ be the gas partial pressure a in b^{th} cluster, and the initialization of gas is performed on the following equations:

$$N_b(k) = e_1 \times rand(0,1) \quad (21)$$

$$E_{a,b} = e_2 \times rand(0,1) \quad (22)$$

$$O_b = e_3 \times rand(0,1) \quad (23)$$

where e_1, e_2 and e_3 be the constant values.

Step 2: Fitness value computation. The fitness prediction is assists to discover the optimal value. The solution with less error is selected as best solution:

$$F = \frac{1}{K} \sum_{\delta=1}^{K} [Y_\delta - \Omega_\delta]^2 \quad (24)$$

where Y depicts the output of SNFN, Ω is the target output, and F is fitness measure.

Step 3: Clustering. The population agents are subdivided into several amounts of similar clusters that are similar to all gas categories. Thus, every cluster has a gas and similar Henry's constant value (N_b).

Step 4: Evaluation. All the clusters y are computed to identify the best gas relies on equilibrium state from the others in its category. After that, the gases are ordered to acquire the best gas in the whole swarm.

Step 5: Renew the coefficient of Henry. This step is used to up-date the Henry coefficient in best location, when the best gas is obtained. The update equation of Henry's coefficient is conveyed as follows:

$$N_b(k+1) = N_b(k) \times \exp\left(-O_b \times \left(\frac{1}{t(k)} - \frac{1}{t^\delta} \right) \right) \quad (25)$$

$$t(k) = \exp\left(-\frac{k}{iter} \right) \quad (26)$$

where N_b is the cluster b Henry coefficient, the heat is denoted as t is, t^δ is a stable value, $iter$ denotes all amount of iterations.

Step 6: Modify solubility. After updating the Henry coefficient, then the solubility of gas is also updated using the expression given below:

$$T_{a,b}(k) = \Re \times N_b(k+1) \times E_{a,b}(k) \quad (27)$$

where $T_{a,b}$ is the a^{th} gas solubility in b^{th} cluster, $E_{a,b}$ is the partial pressure on a^{th} gas in b^{th} cluster and $\Re$ is a stable value.

Step 7: Update location. The gas location is modified using Equation (28):

$$R_{a,b}(k+1) = R_{a,b}(k) + Gr\chi(R_{a,best}(k) - R_{a,b}(k)) + Gr\omega\left(T_{a,b}(k)R_{best}(k) - R_{a,b}(k)\right) \quad (28)$$

where $R_{a,b}$ denotes the position of gas a in cluster b, r is the random number, k is the iteration time, $R_{a,best}$ is the best gas a in cluster y, R_{best} denotes the optimal gas in swarm, χ is the interaction ability of gas b in cluster a, ω depicts the constant.

In order to overcome this, ROA algorithm is introduced to obtain the best solution in HGSO algorithm. From ROA follower (Binu and Kariyappa, 2018), the updated equation becomes:

$$R_{a,b}(k+1) = R^{L}_{a,b}(k) + \left[\cos(J^{k}_{a,b}) \times R^{L}(L,b) \times g^{k}_{a}\right] \quad (29)$$

$$R^{L}_{a,b}(k) = \frac{R_{a,b}(k+1)}{\left[1+\cos(J^{k}_{a,b}) \times g^{k}_{a}\right]} \quad (30)$$

Substitute leading rider position of ROA in the best solution of HGSO, thereby Equation (28) can be rewritten as:

$$R_{a,b}(k+1) = R_{a,b}(k) + Gr\chi(R_{a,best}(k) - R_{a,b}(k)) + Gr\omega\left(\frac{T_{a,b}(k)R_{a,b}(k+1)}{1+(\cos(J^{k}_{a,b}) \times g^{k}_{a})} - R_{a,b}(k)\right) \quad (31)$$

$$R_{a,b}(k+1) = \frac{1+(\cos(J^{k}_{a,b}) \times g^{k}_{a})}{1+(\cos(J^{k}_{a,b}) \times g^{k}_{a}) - Gr\omega T_{a,b}(k)} \left[R_{ab}(k)(1-Gr(\omega+\chi)) + Gr\chi R^{k}_{a,best}\right] \quad (32)$$

$$g^{k}_{a} = h^{k}_{a} \times \left(\frac{1}{T_{off}}\right) \quad (33)$$

$$\chi = \lambda \exp\left(\frac{-H_{best}(k)+\varepsilon}{F_{a,b}(k)+\varepsilon}\right), \ \varepsilon = 0.5 \quad (34)$$

where $J^{k}_{a,b}$ depicts the steering angle, h^{k}_{a} depicts the velocity of a^{th} rider, T_{off} depicts the off time, $R_{a,b}(k)$ represents the location of gas in b^{th} cluster, H_{best} depicts the fitness of best gas, $F_{a,b}(k)$ represents the fitness of gas a in cluster b, $T_{a,b}(k)$ represents the solubility of gas and $N_b(k+1)$ depicts the Henry coefficient.

Step 8: Getaway from local optimum. The local best is escaped in this step, and the worst agent selection is depicted as:

$$Y_{worst} = Y \times \left(rand\left(v_2 - v_1\right) + v_1\right), \ v_1 = 0.1 \text{ and } v_2 = 0.2 \quad (35)$$

where Y is the count of search agents.

Step 9: Renew worst agents location. The expression for updating the worst agents is given as:

$$B_{a,b} = B_{\min(a,b)} + r \times \left(B_{\max(a,b)} - B_{\min(a,b)}\right) \quad (36)$$

where $B_{a,b}$ is the location of gas a in cluster b, r denotes the random number, B_{max} and B_{min} denotes the bound of the problem. Table 1 describes the Pseudocode of developed RHGSO algorithm. The classified output of RHGSO-based DNFN is represented as Γ.

2.6.2. Second Level Classification

This classification is completed by DRN, which is utilized to classify the unhealthy plant image into blast, BLB or brown spot images. The advantage of DRN is the simple model, achieved low error rate and less number of constraints with good classification performance. The structural design of DRN classifier is given below.

Table 1. Pseudocode of Developed RHGSO

Pseudocode of developed RHGSO algorithm
Initialization: $R_a (a = 1, 2, ..., Y)$, number of gas types, N_b, $E_{a,b}$, O_b, e_1, e_2 and e_3
Categorize the gas into several types
Estimate every cluster b
Estimate the optimal gas $B_{a,best}$ and best exploration agent B_{best}
While $k \prec$ *Maximum number of iterations* **do**
For every search agent do
Renew the locations of all search agents by Equation (28)
End for
Renew Henry's coefficient of all gas category by Equation (25) and (26)
Renew solubility of every gas type through Equation (27)
Rank and chose the all worst agents through Equation (35)
Renew the position of the worst agents using Equation (36)
Renew the best gas $R_{a,best}$, and the best search agent R_{best} using Equation (32)
End while
$k = k + 1$
Return R_{best}

2.6.2.1. DRN Architecture

The architecture of DRN (Chen et al., 2019) comprises several layers, like linear layer, convolutional (Conv) layer, residual layer and average pooling layer. The unhealthy plant image Γ is given to the DRN input. The schematic diagram of DRN is given in Figure 4.

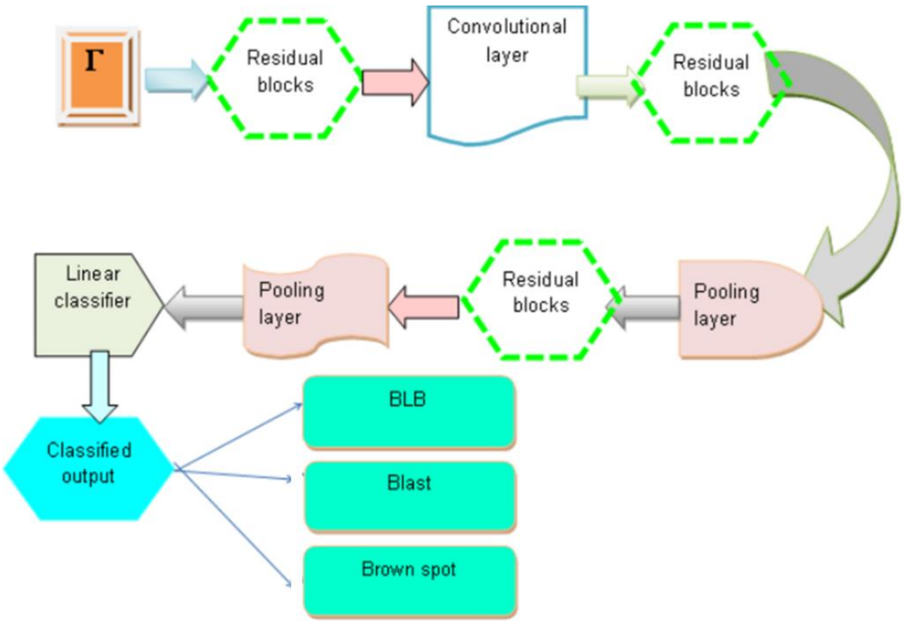

Figure 4. Architecture of DRN.

(*a*) *Conv Layer*

The Conv layer is indicated as:

$$P2e(\Phi)=\sum_{c=0}^{\varphi-1}\sum_{i=0}^{\varphi-1}\Gamma_{c,i}\bullet\Phi_{(k+c),(s+i)} \tag{37}$$

$$P1e(\Phi)=\sum_{W=0}^{X_{in}-1}T_W*\Phi \tag{38}$$

where preceding layer 2D outcome is Φ, *k* and *s* are employed to documented the coordinates, Γ indicates $\varphi \times \varphi$ kernel matrix that is the learnable attribute selected during the process of training, and *c* and *i* represents the position kernel matrix index. Thus, T_W shows the kernel size W^{th} input neuron, and * signifies the cross-correlation operator.

(*b*) *Pooling Layer*

The pooling layer input and output matrix is depicted as follows:

$$r_{out}=\frac{r_{in}-\rho_a}{\delta}+1 \tag{39}$$

$$x_{out}=\frac{x_{in}-\rho_b}{\delta}+1 \tag{40}$$

where r_{in} denotes the input matrix width, x_{in} depicts the input matrix span, r_{out} and x_{out} signifies the final result, ρ_a and ρ_b denotes kernel size-based width and height.

(*c*) *Activation Function*

Non-linear activation function activated the non-linearity of extracted objects from images. It engaged Re*LU* for classifying the images and the expression is depicted as:

$$\mathrm{Re}LU(\Phi)=\begin{cases}0;\ \Phi<0\\ \Phi;\ \Phi\geq 0\end{cases} \tag{41}$$

where Φ denotes feature.

(*d*) *Batch Normalization*

Here, mini batches are formed by the division of training data. This process is used to achieve tradeoff between prediction intricacy and the convergence. It improves the reliability and training speed by modifying the activation, in which the normalization is performed in the input layer during scaling.

(*e*) *Residual Blocks*

The Conv layers' shortcut relationships are established using the residual block. The size matching factor is applied when the input and output are not in same size. Otherwise, the input is linked directly to the outcome:

$$o=D(\Phi)+\Phi \tag{42}$$

$$o=D(\Phi)+C\Phi \tag{43}$$

where Φ and *o* denote the blocks of input and output, *D* depicts the association among mapping, and *C* demonstrates the dimension matching factor.

(*f*) *Linear Classifier*

The linear classifier classified the unhealthy plant image into several diseases from the classified image Γ. The linear classifier is formed by adding up the softmax function and fully connected layer:

$$o = C_{U \times V} \Phi_{V \times M} + Y_{L \times M} \tag{44}$$

Here, $C_{U \times V}$ demonstrates the matrix weight, $\Phi_{V \times M}$ signifies the input feature map, and $Y_{L \times M}$ denotes bias. The DRN architecture result is depicted as o, which means the obtained classified output is BLB, blast or brown spot diseases.

2.6.2.2. Training Procedure of DRN

The DRN training process is explained in this section. The DRN weights are tuned by the implemented RHGSO algorithm. Thus, the rice plant diseases classification is performed, and it obtains better performance by combining ROA with HGSO algorithm.

3. Results and Discussion

This section depicts the results and discussion of implemented approach for the rice plant leaf diseases identification and classification.

3.1. Experimental Setup

The developed rice plant disease detection and classification is implemented in python tool version 3.7.6 using rice plant dataset and rice disease dataset.

3.2. Performance Metrics

For the implemented rice leaf diseases classification metrics are sensitivity, accuracy, specificity, and F1-score.

3.3. Experimental Results

Figure 5 depicts the experimental outcomes obtained by the devised technique. Figure 5(a) shows the input image-1 and 2, Figure 5(b) depicts the pre-processed image-1 and 2, and Figure 5(c) illustrates the output of segmented image-1 and 2.

3.4. Performance Analysis

The performance analysis of devised techniques with different iteration is displayed in this section.

3.4.1. Performance Analysis of Developed RHGSO-Based DNFN for First Level Classification

Figure 6 shows the effectiveness analysis of developed first level classification based on varying the training data. Figure 6(a) illustrates the accuracy analysis of first level classification. For iteration 25, 50, 75 and 100, the accuracy achieved by the implemented approach is 0.877, 0.903, 0.918 and 0.924 by considering 90% training data. Figure 6(b) illustrates the sensitivity analysis of first level classification. The sensitivity for the iteration = 25 is 0.804, for the iteration = 50, the achieved sensitivity is 0.872, for the iteration = 75, the achieved sensitivity is 0.8774, and for the iteration = 100, the achieved sensitivity is 0.899 when the training data is 80%. Figure 6(c) illustrates the specificity analysis of first level classification. For the training data = 70%, then the specificity achieved by the developed model is 0.731, 0.771, 0.783 and 0.792, while the iteration is 25, 20, 75 and 100, respectively. The analysis of first level classification based on F1-score is illustrated in Figure 6(d). For the training data = 80%, then the specificity achieved by the implemented model is 0.785, 0.831, 0.841, and 0.871, while the iteration is 25, 20, 75 and 100, respectively.

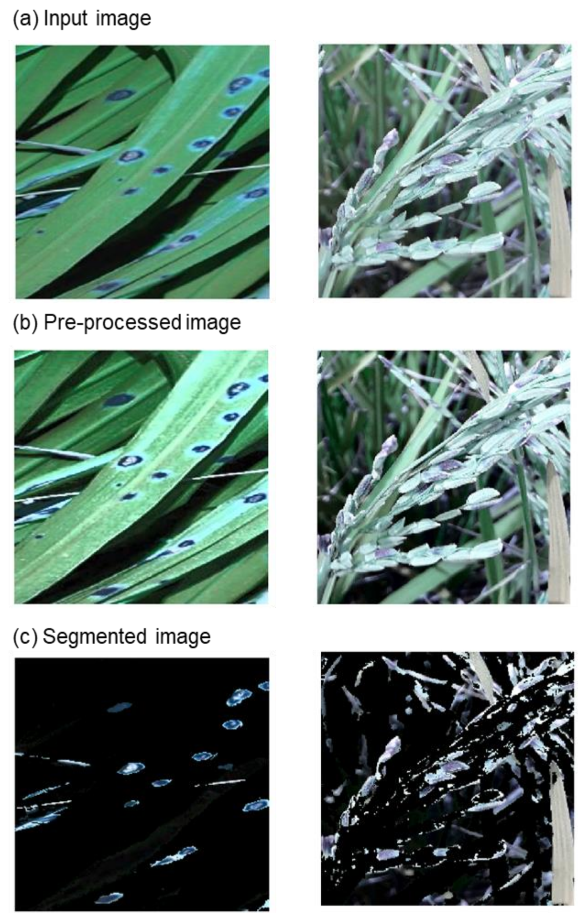

Figure 5. Experimental outcomes of developed method.

3.4.2. Performance Analysis for Second Level Classification

Figure 7 illustrates the effectiveness analysis of second level classification, named RHGSO-based DRN. Figure 7(a) represents the performance analysis of devised classification method based on accuracy. The RHGSO-based DRN achieved 0.885, 0.875, 0.894 and 0.930 accuracy, for the training data is 90% with iteration 25, 50, 75, and 100. The sensitivity analysis is given in Figure 7(b). When the training data = 60%, the corresponding sensitivity value achieved by the developed model is 0.761, 0.790, 0.849 and 0.859 for the iterations 25, 50, 75 and 100. Figure 7(c) depicts the specificity analysis of developed second level classification. For training data is 70%, then the specificity is 0.699 for the iteration = 25, the 0.746 for the iteration = 50, 0.759 for the iteration = 75, and 0.772 for the iteration = 100. The analysis of first level classification based on F1-score is illustrated in Figure 7(d). For the training data = 70%,

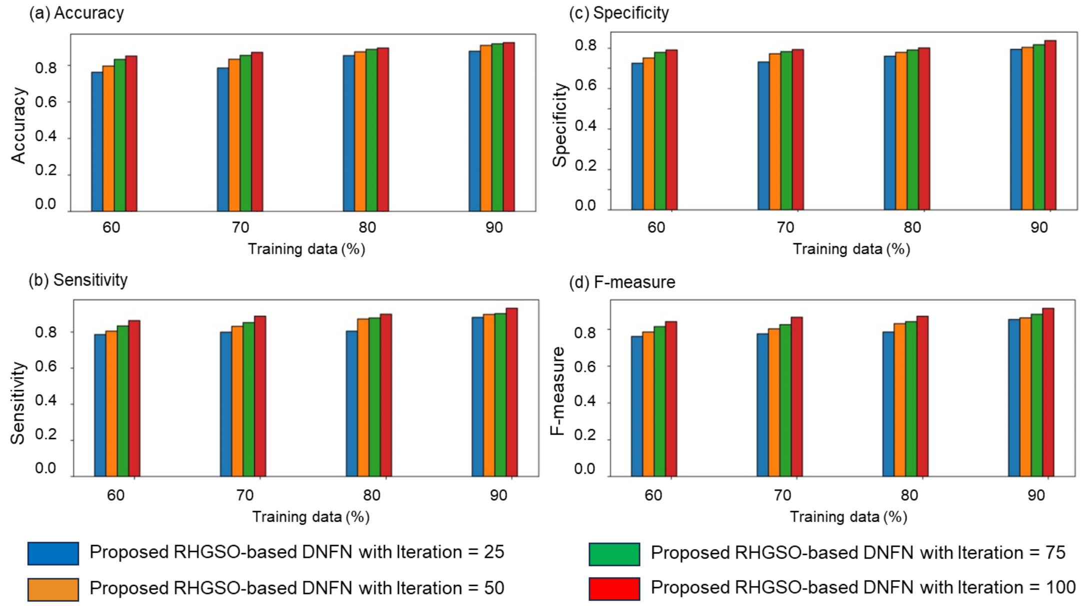

Figure 6. Performance analysis of first level classification.

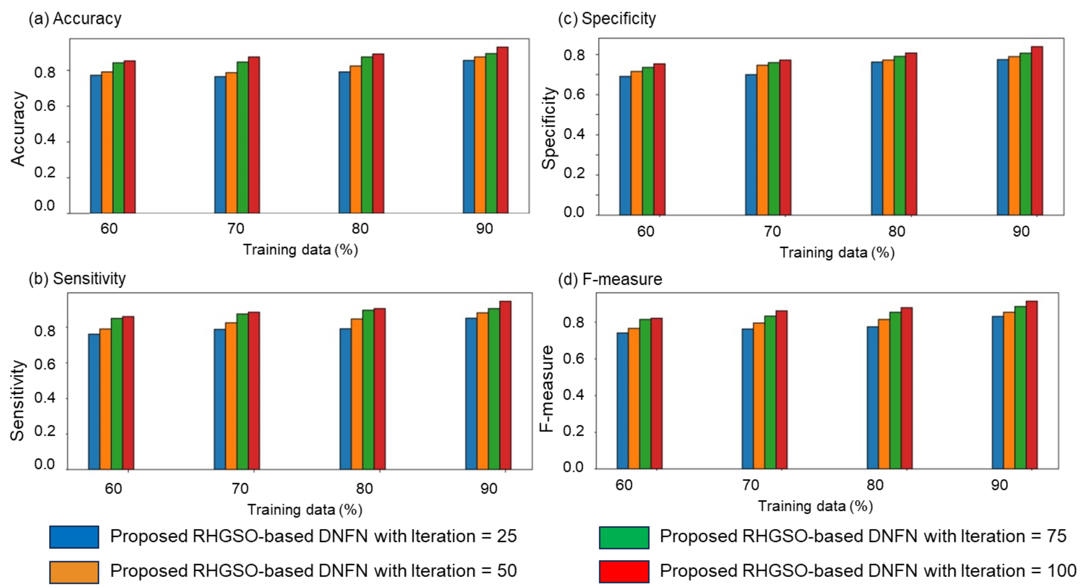

Figure 7. Performance analysis of second level classification.

then the specificity achieved by the developed model is 0.762, 0.795, 0.833, and 0.861, while the iteration is 25, 20, 75 and 100, respectively.

3.5. Competitive Methods

The performance improvement of the developed method is analysed using various existing approaches, like Deep CNN (Li et al., 2020), Multi-level colour image thresholding (Bakar et al., 2018), SVM (Chawal and Panday, 2019), RWW-based neural network (NN) (Daniya and Vigneshwari 2021), RSW-based Deep RNN, Simple CNN (Rahman et al., 2020), DENS-INCEP (Chen et al., 2020), and CNN + SVM (Liang et al., 2019).

3.6. Comparative Analysis

This section represents the comparative analysis of developed RHGSO-based DNFN and RHGSO-based DRN technique by varying training data.

3.6.1. Comparative Analysis for First Level Classification

Figure 8 illustrates the comparative analysis of first level classification, named RHGSO-based DNFN. Figure 8(a) repre-

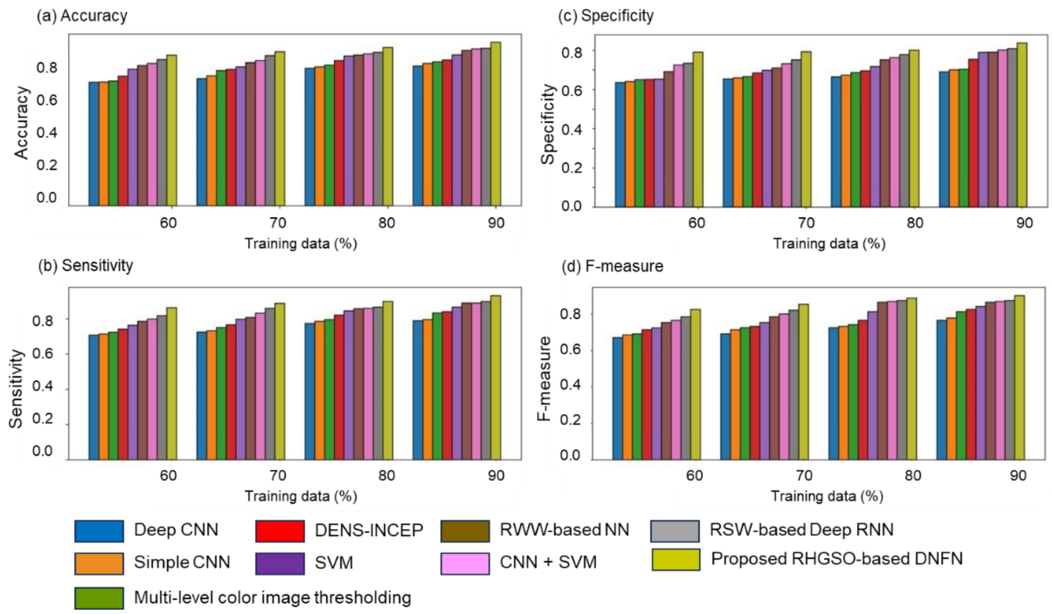

Figure 8. Comparative analysis of first level classification.

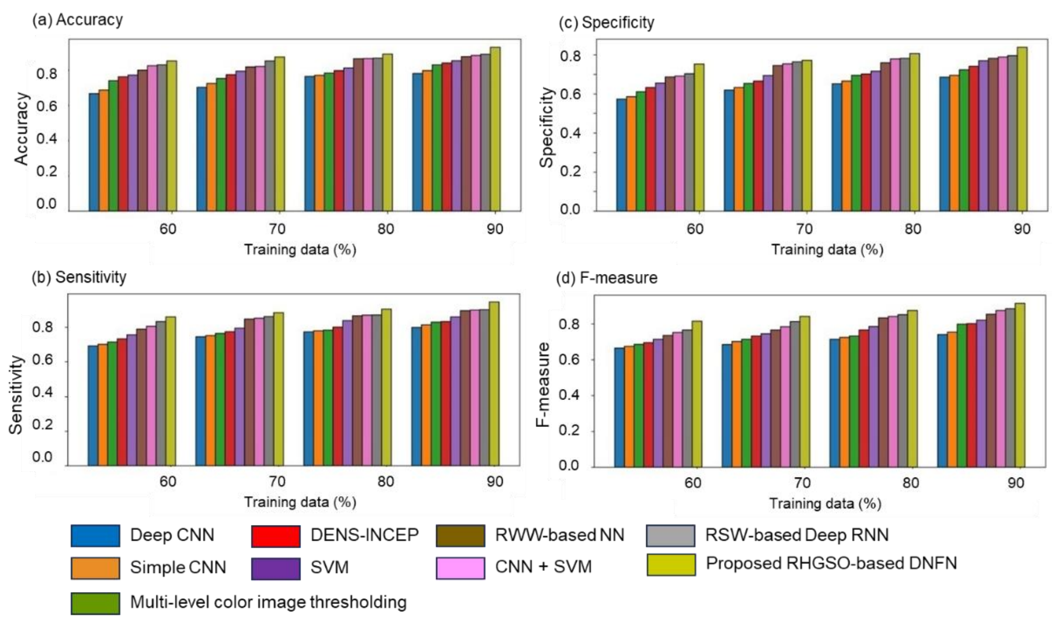

Figure 9. Comparative analysis of second level classification.

sents the comparative analysis of implemented classification method based on accuracy. For the training data is 90%, the accuracy offered by the RHGSO-based DNFN is 0.924, and the existing approaches, like Multi-level colour image thresholding, Deep CNN, RWW-based NN, SVM, RSW-based Deep RNN, Simple CNN, DENS-INCEP, and CNN + SVM achieved the accuracy of 0.813, 0.790, 0.878, 0.853, 0.890, 0.805, 0.825, and 0.887. Here, developed RHGSO-based DNFN obtained a percentage improvement of 12.02, 14.56, 5.00, 7.70, 3.66, 12.91, 10.75, and 4.04%, while compared with the comparison methods. Furthermore, Figure 8(b) indicates the analysis of developed RHGSO-based DNFN using sensitivity. The sensitivity obtained by existing methods are 0.795, 0.773, 0.858, 0.845, 0.866, 0.785, 0.821, and 0.860, and developed RHGSO-based DNFN achieved 0.899 for 80% of training data and the performance improvement is 11.47, 13.98, 4.54, 5.97, 3.57, 12.64, 8.63, and 4.33%, with other methods. The specificity value obtained by Deep CNN is 0.664, Multi-level colour image thresholding is 0.686, SVM is 0.717, RWW-based NN is 0.751 RSW-based Deep RNN is 0.778, Simple CNN is 0.674, DENS-INCEP is 0.695, and CNN + SVM is 0.763, and developed RHGSO-based DNFN is 0.8 for 80% of training data. Figure

Table 2. Comparative Discussion for 1st Level Classification

Parameter	Deep CNN	Multi-level color image threshold	SVM	RWW-based NN	RSW-based Deep RNN	Simple CNN	DENS-INCEP	CNN+SVM	Proposed RHGSO-based DNFN
Accuracy	0.7901	0.8136	0.8536	0.8785	0.8909	0.8054	0.8254	0.8875	0.9248
Sensitivity	0.7894	0.8330	0.8662	0.8893	0.8983	0.7954	0.8412	0.8901	0.9315
Specificity	0.6909	0.7029	0.7895	0.7907	0.8085	0.7001	0.7541	0.8014	0.8374
F1-score	0.7654	0.8142	0.8412	0.8654	0.8745	0.7785	0.8254	0.8696	0.9014

Table 3. Comparative Discussion for 2nd Level Classification

Parameter	Deep CNN	Multi-level color image threshold	SVM	RWW-based NN	RSW-based Deep RNN	Simple CNN	DENS-INCEP	CNN+SVM	Proposed RHGSO-based DNFN
Accuracy	0.7901	0.8136	0.8536	0.8785	0.8909	0.8054	0.8254	0.8875	0.9248
Sensitivity	0.7894	0.8330	0.8662	0.8893	0.8983	0.7954	0.8412	0.8901	0.9315
Specificity	0.6909	0.7029	0.7895	0.7907	0.8085	0.7001	0.7541	0.8014	0.8374
F1-score	0.7654	0.8142	0.8412	0.8654	0.8745	0.7785	0.8254	0.8696	0.9014

8(d) displays the F1-score analysis of devised RHGSO-based DNFN. The F1-score obtained by Deep CNN is 0.725, Multi-level colour image thresholding is 0.743, SVM is 0.814, RWW-based NN is 0.865, RSW-based Deep RNN is 0.875, Simple CNN is 0.778, DENS-INCEP is 0.825, and CNN + SVM is 0.870, and implemented approach is 0.887 for 80% of training data.

3.6.2. Comparative Analysis for Second Level Classification

Figure 9 illustrates the comparative analysis of second level classification, named RHGSO-based DRN. Figure 9(a) indicates the accuracy analysis of RHGSO-based DRN. The accuracy obtained by existing Deep CNN is 0.765, Multi-level colour image thresholding is 0.783, SVM is 0.845, RWW-based NN is 0.813, RSW-based Deep RNN is 0.868, Simple CNN is 0.771, DENS-INCEP is 0.799, and CNN + SVM is 0.866, and developed RHGSO-based DRN achieved 0.891 for 80% of training data. The improved percentage is 12.11, 14.18, 2.90, 8.75, 2.64, 13.50, 10.46, and 2.88%, with other existing methods. Similarly, Figure 9(b) represents the sensitivity analysis of developed classification method. For the training data is 90%, then the RHGSO-based DRN sensitivity is 0.945, then the existing approaches, like Deep CNN is 0.798, Multi-level colour image thresholding is 0.829, SVM is 0.859, RWW-based NN is 0.895, RSW-based Deep RNN is 0.9, Simple CNN is 0.814, DENS-INCEP is 0.833, and CNN + SVM is 0.900. The percentage improvement is 15.58, 12.30, 9.13, 5.30, 4.78, 13.92, 11.99, and 4.91%, while compared with the existing techniques. Figure 9(c) displays the specificity analysis of RHGSO-based DRN. The specificity obtained by Deep CNN is 0.651, Multi-level colour image thresholding is 0.694, SVM is 0.716, RWW-based NN is 0.758, RSW-based Deep RNN is 0.782, Simple CNN is 0.666, DENS-INCEP is 0.701, and CNN + SVM is 0.779, and developed RHGSO-based DNFN is 0.806 for 80% of training data. Figure 9(d) displays the analysis of devised RHGSO-based DNFN with respect to F1-score through varying percentage of training data. The F1-score value obtained by Deep CNN is 0.714, Multi-level colour image thresholding is 0.7325, SVM is 0.785, RWW-based NN is 0.833, RSW-based Deep RNN is 0.851, Simple CNN is 0.725, DENS-INCEP is 0.765, and CNN + SVM is 0.841, and developed RHGSO-based DNFN is 0.875 for 80% of training data.

3.7. Comparative Discussion

Tables 2 and 3 show the comparative discussion of first level classification and second level classification, correspondingly. By considering the 1st level classification the accuracy is 0.925, which is 12.02, 14.56, 5.01, 7.69, 10.75, 3.66, 12.91, and 4.04%, better than the existing Multi-level colour image thresholding, Deep CNN, RWW-based NN, SVM, Simple CNN, RSW-based Deep RNN, DENS-INCEP, and CNN + SVM, respectively. While considering the sensitivity, the existing RSW-based Deep RNN has sensitivity of 0.898, which is 3.56% minimum than the proposed technique. Likewise, the specificity of the existing Deep CNN is 0.691, which is 17.49% minimum than the proposed RHGSO-based DNFN. The devised RHGSO-based DNFN has the F1-score of 0.901, which is 2.98% better than the existing RWW-based NN.

By considering the 2nd level classification the accuracy of the devised RHGSO-based DRN is 0.930, which is 10.69, 16.00, 5.67, 8.15, 14.18, 4.27, 9.57, and 4.98%, better than the existing Multi-level colour image thresholding, Deep CNN, RWW-based NN, SVM, Simple CNN, RSW-based Deep RNN, DENS-INCEP, and CNN + SVM, respectively. Similarly, the sensitivity of the proposed RHGSO-based DRN is 0.946, which is 15.58% better than the existing Deep CNN. Likewise, the sensitivity is 5.09% superior to existing RSW-based Deep RNN. The F1-score is 3.15% superior to existing RSW-based Deep RNN. The conclusion is the effectiveness of the current both levels of classification is superior to the existing methods.

4. Conclusions

This paper presents the developed two-level classifications, named RHGSO-based DNFN and RHGSO-based DRN. The pre-processing process is made and then performing the segmentation process using deep fuzzy clustering to segment the affected area. The feature extraction process is made to extract the features, like statistical features, entropy, LOOP and LGXP and CNN features. Then, the data augmentation process is done by increasing the size of extracted features. The first level classification is completed to classify the image as either healthy or unhealthy using DNFN. Then, the second level classification is performed on the unhealthy images in order to classify the three diseases like blast, BLB and brown spot using DRN. In the first level classification, the developed RHGSO-based DNFN achieved the better accuracy, sensitivity, specificity, and F1-score of 0.925, 0.932, 0.837, and 0.901, whereas the second level classification achieved the better accuracy, sensitivity, specificity, and F1-score of 0.930, 0.946, 0.838, and 0.9142. The future enhancement of this work can be done by including other optimization approaches.

Acknowledgments. I would like to express my very great appreciation to the co-authors of this manuscript for their valuable and constructive suggestions during the planning and development of this research work.

Appendix A. Nomenclature

Abbreviations	Definition
DL	Deep Learning
ROI	Region of Interest
CNN	Convolutional Neural Network
LOOP	Local Optimal-Oriented Pattern
LGXP	Local Gabor XOR Pattern
DNFN	Deep Neuro-Fuzzy network
RHGSO	Rider Henry Gas Solubility Optimization
ROA	Rider Optimization Algorithm
HGSO	Henry Gas Solubility Optimization
DRN	Deep Residual Network
BLB	Bacterial Leaf Blight
SVM	Support Vector Machine
VGG	Visual Geometry Group
LBP	Local Binary Pattern
LDP	Local Directional Pattern
LXP	Local XOR Pattern
ReLU	Rectified Linear Unit
RWW-based NN	RideSpider Water Wave-based Neural Network
RSW	Resistance Spot Welding
DENS-INCEP	DenseNet-Inception

References

Abed-Ashtiani, F., Arzanlou, M., Nasehi, A., Kadir, J., Vadamalai, G. and Azadmard-Damirchi, S. (2018). Plant tonic, a plant-derived bioactive natural product, exhibits antifungal activity against rice blast disease. *Ind. Crops. Prod.,* 112, 105-112. https://doi.org/10.1016/j.indcrop.2017.11.013

Anandkumar, M. (2020). Texton features and deep belief Network for leaf disease classification. *Multimed. Res.,* 3(4), 43-50. https://doi.org/10.46253/j.mr.v3i4.a5

Babu, M.P. and Rao, B.S. (2015). Leaves recognition using back propagation neural network-advice for pest and disease control on crops. *IndiaKisan. Net: Expert Advisory Syst.,* 607-626.

Bakar, M.A., Abdullah, A.H., Rahim, N.A., Yazid, H., Misman, S.N. and Masnan, M.J. (2018). Rice leaf blast disease detection using multi-level colour image. *Thresholding J. Telecommun. Electron. Comput. Eng.,* 10(1-15), 1-6.

Barbedo, J.G.A. (2018). Impact of dataset size and variety on the effectiveness of deep learning and transfer learning for plant disease classification. *Comput. Electron. Agr.,* 153, 46-53. https://doi.org/10.1016/j.compag.2018.08.013

Bhagawati, R., Bhagawati, K., Singh, A.K.K., Nongthombam, R., Sarmah, R. and Bhagawati, G. (2015). Artificial neural network assisted weather based plant disease forecasting system. *IJRITCC*, 3(6), 4168-4173. https://doi.org/10.17762/ijritcc.v3i6.4612

Binu, D. and Kariyappa, B.S. (2018). RideNN: A new rider optimization algorithm-based neural network for fault diagnosis in analog Circuits. *IEEE Trans. Instrum. Meas.,* 68(1), 2-26. https://doi.org/10.1109/TIM.2018.2836058

Bojja, G.R. and Ambati, L.S. (2020). A novel framework for crop pests and disease identification using social media and AI. *Proceeding of Midwest Association for Information Systems (MWAIS).* Des Moines, Iowa, 1-4.

Bojja, G.R., Liu, J. and Ambati, L.S. (2020). Health information systems capabilities and hospital performance – An SEM analysis. *Proceeding of Americas Conference on Information Systems*.

Brahimi, M., Boukhalfa, K. and Moussaoui, A. (2017). Deep learning for tomato diseases: classification and symptoms visualization. *Appl. Artif. Intell.,* 31(1-4), 299-315. https://doi.org/10.1080/08839514.2017.1315516

Chakraborti, T., Cane, B.M., Mills, S. and Pal, U. (2018). LOOP Descriptor: local optimal-oriented pattern. *IEEE signal process. Lett.,* 25(5), 635-639. https://doi.org/10.1109/LSP.2018.2817176

Chawal, B. and Panday, S.P. (2019). Rice plant disease detection using twin support vector machine (TSVM). *J. Sci. Eng.,* 7, 61-69. https://doi.org/10.3126/jsce.v7i0.26794

Chen, J., Zhang, D., Nanehkaran, Y.A. and Li, D. (2020). Detection of rice plant diseases based on deep transfer learning. *J. Sci. Food Agr.,* 100(7), 3246-3256. https://doi.org/10.1002/jsfa.10365

Chen, Z., Chen, Y., Wu, L., Cheng, S. and Lin, P. (2019). Deep residual network based fault detection and diagnosis of photovoltaic arrays using current-voltage curves and ambient conditions. *Energy Convers. Manag.,* 198, 111793. https://doi.org/10.1016/j.enconman.2019.111793

Daniya, T. and Vigneshwari, S. (2021). Deep neural network for disease detection in rice plant using the texture and deep features. *Comput. J.,* 65(7): 1812-1825. https://doi.org/10.1093/comjnl/bxab022

Dong, P.X. and Wang, X.D. (2013). Recognition of greenhouse cucumber disease based on image processing technology. *Open J. Appl. Sci.,* 3(1), 27-31. https://doi.org/10.4236/ojapps.2013.31B006

Feng, Q., Chen, L., Chen, C.L.P. and Guo, L. (2020). Deep fuzzy clustering - a representation learning approach. *IEEE Trans. Fuzzy Syst.,* 28(7), 1420-1433. https://doi.org/10.1109/TFUZZ.2020.2966173

Hashim, F.A., Houssein, E.H., Mabrouk, M.S., Al-Atabany, W. and Mirjalili, S. (2019). Henry gas solubility optimization: a novel physics-based algorithm. *Future Gener. Comp. Sys.,* 101, 646-667. https://doi.org/10.1016/j.future.2019.07.015

Javaid, S., Abdullah, M., Javaid, N., Sultana, T., Ahmed, J. and Abdul Sattar, N. (2019). Towards buildings energy management: using seasonal schedules under time of use pricing tariff via deep neuro-fuzzy optimizer. *Proceedings of 15th International Wireless Communications and Mobile Computing Conference (IWCMC)*, Tangier, Morocco, 1594-1599. https://doi.org/10.1109/IWCMC.2019.87666

73
Jiang, F., Lu, Y., Chen, Y., Cai, D. and Li, G. (2020). Image recognition of four rice leaf diseases based on deep learning and support vector machine. *Comput. Electron. Agric.*, 179(2), 105824. https://doi.org/10.1016/j.compag.2020.105824
Karmokar, B.C., Ullah, M.S., Siddiquee, M.K. and Alam, K.M.R. (2015). Tea leaf diseases recognition using neural network ensemble. *Int. J. Comput. Appl.*, 114(17), 27-30. https://doi.org/10.5120/20071-1993
Khairnar, K. and Dagade, R. (2014). Disease detection and diagnosis on plant using image processing – a review. *Int. J. Comput. Appl.*, 108(13), 36-38. https://doi.org/10.5120/18973-0445
Khush, G.S. (2005). What it will take to feed 5.0 billion rice consumers in 2030. *Plant Mol. Biol.*, 59(1), 1-6. https://doi.org/10.1007/s11103-005-2159-5
Krishnamoorthy, D.N. and Parameswari, V.R.L. (2018). Rice leaf disease detection via deep neural networks with transfer learning for early identification. *Turk. J. Physiother. Rehabil.*, 32, 2.
LeCun, Y., Bengio, Y. and Hinton, G. (2015). Deep learning. *Nature.*, 521(7553), 436-444. https://doi.org/10.1038/nature14539
Li, D., Wang, R., Xie, C., Liu, L., Zhang, J., Li, R., Wang, F., Zhou, M. and Liu, W. (2020). A recognition method for rice plant diseases and pests video detection based on deep convolutional neural network. *Sensors*, 20(3), 578. https://doi.org/10.3390/s20030578
Li, Z., Zhang, Q., Wei, W., Cui, S., Tang, W. and Li, Y. (2020). Determining effects of water and nitrogen inputs on wheat yield and water productivity and nitrogen use efficiency in China: A quantitative synthesis. *Agric. Water Manag.*, 242, 106397. https://doi.org/10.1016/j.agwat.2020.106397
Liang, W.J., Zhang, H., Zhang, G.F. and Cao, H.X. (2019). Rice blast disease recognition using a deep convolutional neural network. *Sci Rep.*, 9(1), 1-10. https://doi.org/10.1038/s41598-019-38966-0
Liu, B., Zhang, Y., He, D.J. and Li, Y.X. (2018). Identification of apple leaf diseases based on deep convolutional neural networks. *Symmetry*, 10(1), 11. https://doi.org/10.3390/sym10010011
Lu, Y., Yi, S., Zeng, N., Liu, Y. and Zhang, Y. (2017). Identification of rice diseases using deep convolutional neural networks. *Neurocomputing*, 267, 378-384. https://doi.org/10.1016/j.neucom.2017.06.023
Miao, F., Yuan, L., Cui, S., Jagadamma, S., Yang, G. and Zhang, Q. (2019). Soil extracellular enzyme activities under long-term fertilization management in the croplands of China: a meta-analysis. *Nutr. Cycl. Agroecosys.*, 114(2), 125-138. https://doi.org/10.1007/s10705-019-09991-2
Niu, X.X. and Suen, C.Y. (2012). A novel hybrid CNN–SVM classifier for recognizing handwritten digits. *Pattern Recognit.*, 45(4), 1318-1325. https://doi.org/10.1016/j.patcog.2011.09.021
Phadikar, S., Sil, J. and Das, A.K. (2013). Rice diseases classification using feature selection and rule generation techniques. *Comput. Electron. Agric.*, 90, 76-85. https://doi.org/10.1016/j.compag.2012.11.001
Rahman, C.R., Arko, P.S., Ali, M.E., Khan, M.A.I., Apon, S.H., Nowrin, F. and Wasif, A. (2020). Identification and recognition of rice diseases and pests using convolutional neural networks. *Biosyt Eng.*, 194, 112-120. https://doi.org/10.1016/j.biosystemseng.2020.03.020
Rice disease dataset. https://github.com/aldrin233/RiceDiseasesDataSet (accessed on May 2021).
Rice plant dataset. https://www.kaggle.com/rajkumar898/rice-plant-dataset (accessed on May 2021).
Roy-Barman, S. and Chattoo, B.B. (2005). Rice blast fungus sequenced. *Curr. Sci.*, 89(6), 930-932.
Sabrol, H. and Satish, K. (2016). Tomato plant disease classification in digital images using classification tree. *Proceedings of 2016 International Conference on Communication and Signal Processing (ICCSP)*, Melmaruvathur, India, 1242-1246.
Sethy, P.K., Barpanda, N.K., Rath, A.K. and Behera, S.K. (2020). Deep feature based rice leaf disease identification using support vector machine. *Comput. Electron. Agr.*, 175, 105527. https://doi.org/10.1016/j.compag.2020.105527
Shrivastava, V.K., Pradhan, M.K., Minz, S. and Thakur, M.P. (2019). Rice plant disease classification using transfer learning of deep convolution neural network. *ISPRS Archives.*, 3(6), 631-635. https://doi.org/10.5194/isprs-archives-XLII-3-W6-631-2019
Sladojevic, S., Arsenovic, M., Anderla, A., Culibrk, D. and Stefanovic, D. (2016). Deep neural networks based recognition of plant diseases by leaf image classification. *Comput. Intell. Neurosci.*, 3289801. https://doi.org/10.1155/2016/3289801
Zhang, L.N. and Yang, B. (2014). Research on recognition of maize disease based on mobile internet and support vector machine technique. *Adv. Mat. Res.*, 905(3), 659-662. https://doi.org/10.4028/www.scientific.net/AMR.905.65