



# OPEN Precise and Quantitative Chlorosis Severity Assessment Framework (PQCSAF) using evolutionary superpixels

Sourav Samanta<sup>✉</sup>, Sanjoy Pratihar & Sanjay Chatterji

The increasing need for smart agriculture in the twenty-first century has increased demand for computer-vision-based disease recognition systems encompassing small and large application areas. In recent years, one of the most promising agricultural research areas has been the automated identification of plant diseases using computer vision. Chlorosis is one of the most common diseases of green leaves. It causes the leaves to turn yellow. The severity of chlorosis (chlorophyll decrease) can be perceived by observing the degree of yellowness in the leaves. One of the most challenging tasks in computer vision-based disease diagnosis is severity estimation with precision. This research has developed a novel approach with a high degree of accuracy to detect the disease-affected lesion areas and the degree of severity. The proposed method involves multiple steps, including initial optimization of superpixel algorithm parameters, feature extraction, feature selection, classification, and disease severity estimation. An evolutionary superpixel-based method has been proposed for grouping different colour patches on the leaf. To detect the presence of yellowness, texture features from several categories of superpixels are extracted using color-GLCM techniques. In this work, a multi-swarm Cuckoo search-based feature selection approach has been proposed and utilized to reduce the feature set designed using the color-GLCM measures. Subsequently, the reduced feature set has been employed to classify the superpixels into four distinct categories based on the degree of yellowness. The proposed PQCSAF has been tested with the chlorosis-affected images of *Pongamia pinnata* leaves. The proposed system has been trained using four classifiers: *decision tree* (DT), *k-nearest neighbors* (KNN), *support vector machine* (SVM), and *multi-layer perceptron* (MLP). For categorization of the superpixels according to the four chlorosis stages, the DT, KNN, SVM, and MLP obtained average classification accuracies of 85.6%, 90.00%, 95.00%, and 97.60%, respectively. Finally, the severity index of chlorosis for the whole leaf is reported based on the weighted scores of the superpixels based on their categories. The proposed method demonstrates applicability for its robustness and detection accuracy, as indicated by comparison studies with existing literature. The proposed method can be used to measure chlorosis severity for different types of plants and various leaf diseases. Due to its adaptive qualities, the proposed model has the potential to be applied to on-field AI edge devices in future.

**Keywords** *Pongamia pinnata*, Plant pathology, Chlorosis disease, Disease detection, Superpixel, Evolutionary feature selection, Multi swarm Cuckoo search, Disease severity index

Recently, there has been a noticeable trend towards the convergence of computer vision and artificial intelligence within plant pathology. This convergence between the two domains is becoming more prominent and increasing the scope of development of new technology in the early diagnosis and monitoring of plant diseases<sup>1,2</sup>. Plant pathology aims to investigate the plant diseases induced by pathogens and environmental factors<sup>3,4</sup>. Particularly, using computer vision to detect crop diseases helps farmers take steps at the right moment for associated treatments. Apart from the crops, other types of trees are also considered for pathological research, depending on their medicinal and economic significance.

Department of Computer Science and Engineering, Indian Institute of Information Technology Kalyani, Kalyani, West Bengal 741235, India. ✉email: sourav.uit@gmail.com

Currently, researchers are eager to identify diseases at the earliest possible stage. This is because early detection of diseases can help prevent problems such as minimizing yield losses<sup>5,6</sup>, preventing disease spread in large-scale spread<sup>7,8</sup>, managing specific pathogens and emerging diseases<sup>9,10</sup>, and so on. Furthermore, early disease identification is essential for sustainable and precision agriculture as it enables accurate and timely treatments that reduce environmental impact and enhance resource utilization<sup>11,12</sup>. Current methods for early detection reduce the need for chemical pesticides, encouraging environmentally sensitive disease management and substantially enhancing sustainable agricultural practices<sup>13</sup>. Severity measurement provides a quantitative assessment of how much a disease impacts a plant or its parts. This provides more information than identifying the existence or absence of a disease. An adequate understanding of severity assists in evaluating the intensity of the illness in individual plants or within a population. It can serve as a crucial component of disease monitoring and management. An appropriate severity-measuring approach may accurately assess overall plant health, monitor disease progression, and enable intelligent decision-making for precise and targeted treatment. In the automatic severity estimation approach, the accurate localization of the disease region on the leaf is crucial. Furthermore, accurately classifying the lesion stage is essential for exact severity assessment. The proposed work addresses the first task by optimizing parameters for simple linear iterative clustering (SLIC) and the second task by classifying superpixels using the optimized feature set. Researchers have developed several strategies for illness detection and severity estimation in recent years, explored in the section “[Related works on plant disease detection](#)”. A limited number of research studies have investigated the SLIC for severity estimation. The SLIC parameters have been optimized to provide more precise results, followed by feature selection, superpixel classification, and, ultimately, severity score computation in the proposed work.

*Chlorosis* is among the most frequently encountered diseases affecting green leaves, leading to the yellowing of leaves<sup>14</sup>. The main reason for chlorosis is a viral infection. At the beginning of yellowing, no significant leaf structure and geometry changes are observed. Chlorophyll decreases cause due to change of the structure and function of the chloroplast. Chlorophyll is essential for photosynthesis and gives a green color to the leaf. Iron and manganese deficiency results in leaf yellowing (chlorosis)<sup>15</sup>. Chlorosis is also caused by excess potassium, magnesium, and phosphorus<sup>16</sup>. Chlorosis is a visible symptom that might indicate several underlying concerns, including nutrient deficits, infections, pests, environmental stresses, or physiological disorders. Plant pathologists can accurately diagnose and identify the underlying causes of plant health problems by comprehending chlorosis patterns and related factors. Furthermore, it has the potential to greatly influence both the quantity and quality of crop production. Plant pathologists can enhance agricultural productivity by analysing chlorosis patterns and identifying their underlying causes. Implementing effective techniques for alleviating nutritional deficiencies, diseases, and other stress factors is made possible because of this, which ultimately results in a reduction in yield losses. In addition, chlorosis is a symptom that can be caused by a wide variety of illnesses and pests. Developing an integrated disease prevention and surveillance control strategy necessitates thoroughly comprehending the relationship between chlorosis and insect or disease infestations. Plant pathologists must accurately identify and measure chlorosis in order to develop the appropriate strategy for automated decision-making in smart agriculture, as evidenced by the previous discussion. The principal objective of our investigation is to develop a framework for accurately quantifying chlorosis, thereby enabling the possibility of early prediction. Therefore, we developed a system that enables automated localization. The localization of the diseased region is enhanced by an optimization layer that enables the automated adjustment of system parameters based on disease characteristics to some extent. Moreover, precise classification of the superpixels enables the system to identify the severity stages. Ultimately, it may serve as a tool for high-precision disease detection, facilitating early identification in precision agriculture.

In the framework, the superpixel method is employed on plant leaves to produce superpixels that cluster the different yellowing stages. An evolutionary optimization method is utilized to improve the quality of superpixels and adjust parameter values based on leaf conditions. Subsequently, color features are derived from four different types of superpixels. A novel feature selection strategy based on multi-swarm Cuckoo search has been implemented to identify crucial features from the extensive feature set.

Finally, a model for estimating severity based on superpixels has been introduced to assess the overall severity score accurately. The following points distinguish the proposed method from other existing models.

1. A machine learning-based precise chlorosis quantification model for plant chlorosis monitoring has been proposed.
2. The Evolutionary superpixel method has been introduced for accurate disease localization. Furthermore, the evolutionary approach helps to automatically select the superpixel's parameter value according to the leaf's disease stage.
3. The suggested approach is a comprehensive model for chlorosis disease detection, lesion area localization, disease stages, and severity score calculation.
4. Additionally, a novel feature selection method based on a multi-swarm Cuckoo search has been deployed to optimize the selection of features from a vast collection, thus enhancing the efficiency of the classification model.
5. The chlorosis severity index of the diseased leaf has been proposed and calculated to understand the effect of chlorosis on the leaf.
6. Extensive experiments have been conducted to validate the proposed model with different evaluation parameters.

The structure of the paper is as follows. An overview of the possible use of computer vision in plant pathology is provided in the section “[Introduction](#)”. The paper discusses recent advancements in plant disease diagnosis technologies in the section “[Related works on plant disease detection](#)”. The required techniques, assessment

metrics, and details on the dataset are found in the section “Materials and methods.” The section “Proposed precise chlorosis severity assessment system” comprehensively describes the proposed precise chlorosis assessment system. The section “Results and discussion” illustrates the exhaustive analysis of the results obtained from the various stages of the proposed system. The conclusions have been summarized in the section “Conclusion & future work”.

### Related works on plant disease detection

Over the years, many approaches have been developed to detect and classify an infection on plant leaves automatically. Compared to conventional processing methods, the wavelet transform results improve image resolution and better preserve edge information. Discrete Wavelet Transform (DWT) is one of the most widely used methods in leaf disease identification agriculture. Pankaja et al.<sup>17</sup> presented a leaf identification model which consists of statistical information from HSV color moments and local features. Texture information and shape features are extracted by DWT and Chebyshev Moments, respectively. All three features are then combined to produce the final feature vector. The proposed model took thirty classes from the Flavia dataset, totalling 270 leaf images. This proposed feature extraction scheme improves the system’s accuracy to 96.29%. Islam et al.<sup>18</sup> presents the method which decomposes the input disease image into horizontal, vertical, and diagonal sub-bands using the multi-resolution analysis and Discrete Wavelet Transform (DWT). The random subspace Method was used to classify data by a combination of linear classifiers. The method has an accuracy rate of 95% in identifying four major rice plant diseases. Tampinongkol et al.<sup>19</sup> proposed a method to identify the source of diseases from Jabon leaf image in the nursery. Segmentation has been done by reducing the RGB color cylinders to isolate the disease entity from the background. Finally, using DWT in the 3-level decomposition process, the disease type can be identified. SVM classification results show that the disease groups distinguish from each other with accuracy 84.67%. Dhingra et al.<sup>20</sup> have proposed a computer vision-based leaf disease detection method by combining DWT with different Gabor filter orientations to find the features. The later stage employs a histogram binning pattern. A Gaussian classifier makes the final prediction on the extracted features. They did a comparative analysis on five leaf data sets and proved the supremacy of the method. On various data sets, the proposed system got results with an accuracy of 98.45%, 96.45%, 99.85%, 99.83%, and 99.75%. Superpixeling is a clustering technique that groups adjacent pixels to form homogeneous regions, seeing the similarity of color or texture information<sup>21</sup>. A superpixel-based method for identifying leaf disease has been proposed in recent years. Zhang et al.<sup>22</sup> introduced a hybrid clustering-based segmentation approach for plant disease identification. In this work, the *simple linear iterative clustering (SLIC)* divides the entire color leaf image into several compact and nearly uniform superpixels. Then, the pixels defining the lesion are segmented from each superpixel by expectation (objective) maximization. The experimental results show that the method is effective compared to other methods. Another hybrid approach proposed by Zhang et al.<sup>23</sup> for detecting cucumber diseases uses superpixels. The method consists of two stages; where in the first stage, the diseased leaf image is partitioned into several compact regions using the SLIC algorithm. The logarithmic frequency PHOG features are extracted and used for disease detection in the second stage. A fusion-based approach presented by Khan et al.<sup>24</sup> segments the lesion area of the diseased leaf, which suffers from uneven illumination. In this method, the input image is transformed into a color-balanced image and subsequently the SLIC method is applied. Singh<sup>25</sup> proposed PSO-based lesion area segmentation of six types of sunflower disease. The texture features are extracted from each lesion category and classified with a minimum distance classifier. Finally, this method achieved the average classification of 98%. An adaptive pixel intensity-based thresholding approach was developed by Sengar et al.<sup>26</sup> to segment the lesion area of cherry leaves. The method achieved a very high accuracy of 99.0%. Pandey et al.<sup>27</sup> presents an automatic and non-destructive detection of the healthy, mild, and severe categories of Vigna-mungo leaf. This method segments the leaf images and extract features thereafter. Finally, the disease is predicted using a support vector machine (SVM). Moreover, it can predict the category of the healthiness of the leaf with an accuracy of 95.69%. Chakraborty et al.<sup>28</sup> evaluate the effectiveness of various deep learning models, including VGG16, VGG19, MobileNet, and ResNet50, on the PlantVillage Dataset to create an automated system for detecting late and early blight infections in potato leaves. Compared to other models, VGG16 exhibits the highest accuracy of 92.69%, as evidenced by the experiment. Additionally, they improve the efficacy of VGG16 by adjusting its parameters. Compared to healthy potato leaves, the improved model achieved a classification accuracy of 97.89% in distinguishing between late and early blight disease. Phan et al.<sup>29</sup> proposes an automated method to detect diseased regions on corn leaves using deep learning and image segmentation. By applying Simple Linear Iterative Clustering (SLIC) to segment corn leaf images into super-pixels, and using pre-trained CNN models (like VGG16, ResNet50, DenseNet121, etc.), the system classifies regions as healthy or affected by specific diseases (e.g., NLB, GLS, rust). The DenseNet121 model achieved the highest accuracy of 97.77% on the CD&S dataset with optimal SLIC and train-test settings. The approach proves effective for replacing manual scouting with automated, image-based disease detection and has been deployed via web and mobile applications. Abisha et al.<sup>30</sup> presents a novel approach for detecting and categorising diseased brinjal leaves using Deep Convolutional Neural Networks (DCNN). The first phase was the successive use of image enhancement and segmentation techniques on the original brinjal leaf image. Subsequently, the discrete Shearlet transform is utilised to extract the characteristics of the segmented region afflicted by the illness. The Deep Convolutional Neural Network (DCNN) attained a mean accuracy of 93.30% in accurately categorising leaf diseases. Singla et al.<sup>31</sup> have created a web-based application that utilises a deep learning model. The objective is to detect plant leaf diseases by analysing a image of a leaf and then alerting farmers through message. They examined the different deep learning models using the data from PlantVillage. MobileNet attained a classification accuracy of 97.35% in the multi-classification challenge. Bedi et al.<sup>32</sup> presented a computationally lightweight method for the early diagnosis of plant diseases. The system comprises a Convolutional Autoencoder (CAE) and Few-Shot Learning (FSL), whereby the CAE is trained to reconstruct leaf images with minimal loss, and its pre-trained layers are

utilized to develop classification and segmentation models trained with FSL. The technique offers a scalable solution for diagnosing plant diseases and estimating their severity while requiring minimal annotation effort.

Table 1 gives a summary of the related works.

## Materials and methods

This section briefly explains the SLIC method, Colour GLCM features, principal component analysis, evolutionary feature selection, the data used in the experiment, and other relevant entities associated with this experiment.

### Simple linear iterative clustering

Supapixel methods are quickly becoming famous for a variety of computer vision applications. It decreases image redundancy and reduces the complexity of subsequent image-processing tasks. Among the various superpixel techniques, the work reported in<sup>34</sup> proposed *imple linear iterative clustering (SLIC)* as an effective superpixel generation method. The method uses three-dimensional color space and a two-dimensional image plane, forming a five-dimensional space. The inputs to the algorithm are input image ( $i_m$ ), number of superpixels ( $k$ ), weighting factor ( $m$ ), and radius ( $r$ ). The details of the parameters,  $k$ ,  $m$  and  $r$  are discussed in the section “[Performance analysis of evolutionary superpixel](#)” and a suitable optimization of these parameters has been presented.

---

```

1 Input:  $i_m, k, m, r$ ;
2 Initialize cluster centers  $C_k = [l_k, a_k, b_k, x_k, y_k]^T$  by sampling pixels at regular grid steps  $S$ ;
3 Perturb cluster centers in an  $n \times n$  neighborhood, to the lowest gradient position;
4 while  $E \leq threshold$  do
5   for  $i \leftarrow 1$  to  $c_k$  do
6     Assign the best matching pixels from a  $2S \times 2S$  square neighborhood around the cluster center according to the
       distance measure;
7   Compute new cluster centers and residual error  $E$ ;
8 Return  $i_{spix}$ ;
```

---

### Algorithm 1. SLIC-Algorithm

---

Work	Type of data (leaf)	Objective	Method	Perform.
Zhang et al. (2017) <sup>23</sup>	Diseased cucumber leaf	Lesion segmentation	SLIC, PHOG and EM	91.18%
Pankaja et al. (2018) <sup>17</sup>	Flavia dataset	Leaf recognition and classification	Chebyshev moments and roundness, HSV color moments, DWT texture, SVM	96.29%
Sengar et al. (2018) <sup>26</sup>	Diseased cherry leaf	Segmentation and quantization of lesion area	Adaptive intensity based thresholding	99.0%
Islam et al. (2019) <sup>18</sup>	Diseased rice plant leaf	Rice disease classification	Two level DWT, Ensemble of linear classifier	95.0%
Singh (2019) <sup>25</sup>	Diseased sunflower leaf	Segmentation and disease classification	PSO and texture feature, Minimum distance classifier	98.0%
Tampinongkol et al. (2020) <sup>19</sup>	Diseased Jabon leaf	Classification of spots and blight disease	Three level DWT, SVM	84.67%
Dhingra et al. (2020) <sup>20</sup>	Basil, Tomato, Cherry, Pepper, and Apple Leaf	Healthy and disease classification	DWT, Gabor filter, and Histogram binning pattern, Gaussian classifier	98.86%
Khan et al. (2020) <sup>24</sup>	PlantVillage data set and real world data set	Disease classification	SLIC, GLCM and PHOG, Random Forest	93.12%
Chouhan et al. (2021) <sup>33</sup>	Jatropha Curcas and Pongamia Pinnata	Segmentation and classification	SLIC, ADALINE, SIFT and LBP	98.5%
Pandey et al. (2021) <sup>27</sup>	Diseased vigna mungo leaves	Healthy and disease level identification	Statistical and GLCM features, PCA, SVM	95.69%
Chakraborty et al. (2022) <sup>28</sup>	Potato leaves	Healthy and disease identification	Improved VGG16	97.89%
Phan et al. (2022) <sup>29</sup>	Corn leaves	Healthy and disease identification	SLIC and pre-trained model	97.77%
Abisha et al. (2023) <sup>30</sup>	Diseased brinjal leaves	Healthy and disease identification	Deep CNN, Discrete Shearlet transform	93.30%
Singla et al. (2024) <sup>31</sup>	PlantVillage data set	Healthy and disease identification	MobileNet	97.35%
Bedi et al. (2024) <sup>32</sup>	PlantVillage data set	Disease identification and severity estimation	Few-Shot Learning	99.00%
Proposed method	Pongamia pinnata leaf	Disease severity finding	Optimized SLIC and proposed MSCS based feature selection	97.60%

**Table 1.** A summary of the recent works related to disease detection.

---

### Color-GLCM: texture features

The gray-level co-occurrence matrix (GLCM) is used widely for texture analysis. In this work, a color-based texture analysis has been used based on the color co-occurrence matrix (CCM) to detect disease and subsequently for severity measurement of a leaf. The CCM is calculated using four distance values,  $d = 1, 2, 4$ , and 8 and thirteen (13) directions considering the 3D color channels<sup>35,36</sup>. The twelve (12) features used by us are known as Haralick features<sup>37</sup> extended in 3D GLCM (color-GLCM), and they are given in the equations given below (Eqs. 1–12). They are also reported in<sup>35,36</sup>.

$$Energy = \sum_{i=1}^L \sum_{j=1}^L M_d^\phi(i, j)^2 \quad (1)$$

$$Entropy = \sum_{i=1}^L \sum_{j=1}^L M_d^\phi(i, j) \log_2(M_d^\phi(i, j)) \quad (2)$$

$$Correlation = \sum_{i=1}^L \sum_{j=1}^L M_d^\phi(i, j) \frac{(i - \mu_x)(j - \mu_y)}{\sigma_x \sigma_y} \quad (3)$$

$$Contrast = \sum_{n=1}^L \left( n^2 \sum_{i=1}^L \sum_{j=1}^L M_d^\phi(i, j) \right), n = |i - j| \quad (4)$$

$$Inverse\ difference\ moment = \frac{1}{1 - (i - j)^2} \sum_{i=1}^L \sum_{j=1}^L M_d^\phi(i, j) \quad (5)$$

$$Variance = \sum_{i=1, j=1}^L (1 - \mu)^2 M_d^\phi(i, j) \quad (6)$$

$$Sumaverage = \sum_{i=1}^{2L-1} (ip_{x+y}(i)) \quad (7)$$

$$Dissimilarity = \sum_{i=1}^L \sum_{j=1}^L (i - j) M_d^\phi(i, j) \quad (8)$$

$$Cluster\ shade = \sum_{i=1}^L \sum_{j=1}^L (i + j - \mu_x + \mu_y)^3 M_d^\phi(i, j) \quad (9)$$

$$Cluster\ prominence = \sum_{i=1}^L \sum_{j=1}^L (i + j - \mu_x + \mu_y)^4 M_d^\phi(i, j) \quad (10)$$

$$Maximum\ probability = \sum_{i=1}^L \sum_{j=1}^L \max_{i,j} \{M_d^\phi(i, j)\} \quad (11)$$

$$Difference\ variance = \sum_{i=1}^L \sum_{j=1}^L i^2 p_{x-y}(i) \quad (12)$$

Here,  $L$  is the number of distinct gray levels in the image window,  $M(i, j)$  represents the  $(i, j)$ -th entry in the color-GLCM (3D-GLCM considering three channels R, G and B), and  $p_x(i) = \sum_{j=1}^L M_d^\phi(i, j)$ ,  $p_y(j) = \sum_{i=1}^L M_d^\phi(i, j)$  denotes the  $i$ -th and  $j$ -th entry in the marginal-probability distribution matrix;  $\sigma_x, \sigma_y, \mu_x, \mu_y$  denotes the means and standard deviations of the partial probability density functions  $p_x$  and  $p_y$  respectively and expressed as given in Eqs. 13–16.

$$\sigma_x = \left( \sum_{i=1}^L \sum_{j=1}^L (i - \mu_x)^2 \right)^{1/2} \quad (13)$$

$$\sigma_y = \left( \sum_{i=1}^L \sum_{j=1}^L (j - \mu_y)^2 \right)^{1/2} \quad (14)$$

$$\mu_x = \sum_{i=1}^L \sum_{j=1}^L i M_d^\phi(i, j) \quad (15)$$

$$\mu_y = \sum_{i=1}^L \sum_{j=1}^L j M_d^\phi(i, j) \quad (16)$$

### Principal component analysis

In general, Principal Component Analysis (PCA) is a technique used to reduce the dimensionality of a dataset without significant loss of information<sup>38</sup>. It transforms the original set of possibly correlated features into a new set of uncorrelated variables called principal components<sup>39</sup>. These components are ordered by the amount of variance they capture from the data, allowing for effective dimensionality reduction. PCA can thus identify the most informative directions in the feature space, which can then be used as input to supervised or unsupervised learning algorithms. In the literature, PCA-based feature extraction has been widely employed to improve the performance of classification algorithms<sup>40</sup>.

### Evolutionary feature selection

Evolutionary Feature selection strategy has advantages over traditional feature selection methods when the feature vector dimension is very large<sup>41</sup>. It treats the feature selection problem as a subset selection problem. The evolutionary approach quickly finds the optimal or near-optimal feature set by avoiding the exhaustive search on a large feature set<sup>41</sup>. Recent research on evolutionary feature selection will be covered here. Saraç et al.<sup>42</sup> used an ant colony optimization (ACO) technique to choose the best features for the automatic classification of web pages. They reported that the ACO-based strategy finds better features over information gain and chi-square techniques. Yadav et al.<sup>43</sup> introduced a filter-based feature selection method that utilizes particle swarm optimization (PSO) to find the optimal feature set to improve classifier performance in the named entity recognition problem of biomedical natural language processing. Marie-Sainte et al.<sup>44</sup> applied the firefly algorithm to determine the best feature for classifying Arabic text. The experiment was conducted on an OSAC dataset, and the approach achieved a precision value of 99.40%. Alzaqebah et al.<sup>45</sup> proposed a modified Cuckoo search for feature (genes) selection from the cancer gene expression datasets. They used the concept of memory with the traditional Cuckoo search algorithm to consider the selected features of each iteration. Finally, microarray datasets have been used to compare the method with other algorithms. Khan et al.<sup>46</sup> introduced a new feature selection algorithm based on genetic algorithm (GA) called *diversification of Population (DP)* in GA, and named DPGA. They utilized this method to identify the most advantageous features from the Uniform variant of the LTrP (ULTrP) feature set extracted from the microstructural images. Finally, classifying a seven-class microstructural image dataset using a reduced feature set confirms its superiority over other contemporary methods. In our work, a new multi-swarm-based feature selection strategy has been proposed that will be discussed in the section “Multi-swarm cuckoo search for feature selection”.

### Supervised machine learning

Supervised learning involves an input variable  $X$  and an output variable  $Y$ , and it uses an algorithm to learn the mapping from the input set to the output set. The goal is to approximate the mapping function for an unknown input data  $x'$ , such that we can predict the corresponding output  $y'$ . The algorithm learns using the training dataset, and when the algorithm achieves an acceptable level of accuracy in performance, it stops learning. A brief introduction of the four methods used in our experimentation is given in the following subsections. The parameters' settings of all the classifiers used by us in this work are shown in Table 2.

#### Decision tree

The decision tree (DT) obtains the knowledge as a tree, which can also be expressed as a set of discrete rules<sup>47</sup>. The critical aspect of the decision tree classifier is its ability to use different feature subsets. Moreover, decision rules should be used at different steps of classification. The decision tree classifier is an efficient and robust classifier for complex decision-making problems. Each internal node of the decision tree implies the test on an

Classifier	Type	Specification
Decision Tree	Complex Tree	Number of splits= 100
		Split Criterion= Gini index
k-Nearest Neighbors	Cosine KNN	Number of neighbours=10
		Distance Metric= Cosine
Support Vector Machine	Quadratic Kernel	Multi class method = One to one
		Distance Weight= Equal
Multi Layer Perceptron	Conjugate Gradient	Performance measure= Cross entropy
		No of Hidden Layer= 10
		Data division= Random

**Table 2.** Parameters' settings for various classifiers used in the work.

attribute, and each branch node denotes the test outcome. Each leaf node of the decision tree implies the class label. It performs the classification by splitting branches of the tree. The classification algorithms based on the decision tree can be found in the literature in different forms<sup>48</sup>.

#### *K*-nearest neighbors

The *k*-Nearest-Neighbours (KNN) classifier is a simple and effective classifier based on one lazy learning method. It has been used as a non-parametric technique in statistical estimation and pattern recognition. The basic concept of the KNN classifier is to store the available instances and classify the testing instances as per the similarity that is based on distance<sup>49</sup>. For a specific test sample and a training set, the distances between the test sample and the other samples in the training set can be calculated. The training set closest to the test sample has the shortest distance. As a result, the test sample is classified based on the classification of its nearest neighbor<sup>50</sup>.

#### Support vector machine

The Support Vector Machine (SVM) theory was introduced by<sup>51</sup>. A hyperplane is built to classify data in appropriate classes in realistic time<sup>51</sup>. The data points closest to the hyperplane are referred to as support vectors. The performance of the SVM classifier depends on maximizing the distance between class data points on the hyperplane<sup>52</sup>. SVM uses a different mathematical function called the kernel function. Choosing the proper kernel function is vital to obtain the maximum performance<sup>53</sup>.

#### Multilayer perceptron (MLP)

Multilayer perceptron (MLP) classifier consists of three layers: input, output, and hidden. The input layer receives the signal for further processing, and classification is done by the output layer<sup>54</sup>. The hidden layer between input and output works as the central computation unit of multilayer perceptron. The perceptron generates a single output from multiple real-valued inputs using a linear combination of weighted inputs. The output is forwarded through some nonlinear activation functions. It uses the back propagation learning algorithms to train the network<sup>55</sup>. This classifier can solve most problems that are not linearly separable<sup>56</sup>.

## Experimental setup

The necessary parameters' settings are shown in Table 2. The default settings for all classifiers were used as provided by the respective tools (e.g. WEKA, MATLAB) to ensure reproducibility and alignment with standard baseline practices. This study aims to assess the performance of the proposed model independent of significant hyperparameter tuning.

### Pongamia pinnata dataset

The photos of *Pongamia pinnata* were taken in a controlled setting between March and May 2019 by the Madhav Institute of Technology & Science in India<sup>57,58</sup>. The acquisition procedure was fully equipped with Wi-Fi connectivity. The leaf images were taken with a Nikon D5300 camera equipped with a built-in performance timing feature for capturing shots. The images were captured with a 18 – 55 mm lens in JPG format, with RGB color representation, 24-bit depth, 1000-ISO sensitivity, and without a flash. For experimentation, a set of 550 images has been used in this work. Out of 550 images, 275 are healthy images and 275 images contain chlorosis patches of various severity.

### Performance evaluation metrics

The different classification evaluation parameters have been discussed here. This experiment has considered a four-class classification problem with categories of superpixels labelled as Type-1, Type-2, Type-3, and Type-4. The terms True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) are defined by focusing on one type at a time. For example, if we consider Type-1 as a True Positive (TP), the model correctly predicts a superpixel as Type-1 when it is Type-1. A False Positive (FP) occurs when the model predicts Type 1, but the actual class is either Type 2, Type 3, or Type 4. A False Negative (FN) happens when the actual class is Type-1, but the model incorrectly predicts it as Type-2, Type-3, or Type-4. Finally, a True Negative (TN) means the model correctly predicts an image as not Type-1, i.e., it correctly identifies an image as Type-2, Type-3, or Type-4 when it is truly one of those. This same consideration is repeated separately for Type-2, Type-3, and Type-4 to fully evaluate the model's performance across all classes.

**Accuracy.** Classification accuracy is a quantitative measure that evaluates the effectiveness of a classification model by calculating the ratio of correct predictions to the total number of predictions made.

$$Accuracy = \frac{TP + TN}{TP + FN + FP + TN} \quad (17)$$

**Precision.** Precision evaluates the fraction of correctly classified instances or samples among the ones classified as positives.

$$Precision = \frac{TP}{TP + FP} \quad (18)$$

**Recall.** In an imbalanced classification problem with more than two classes, recall is calculated as the sum of true positives across all classes divided by true positives and false negatives across all classes.

$$Recall = \frac{TP}{TP + FN} \tag{19}$$

**Sensitivity.** Sensitivity (true positive rate) refers to the proportion of those who have the condition truly that received a positive test result.

$$Sensitivity = \frac{TP}{TP + FN} \tag{20}$$

**Specificity.** Specificity (true negative rate) refers to the proportion of those which are truly negative and received a negative test result.

$$Specificity = \frac{TN}{TN + FP} \tag{21}$$

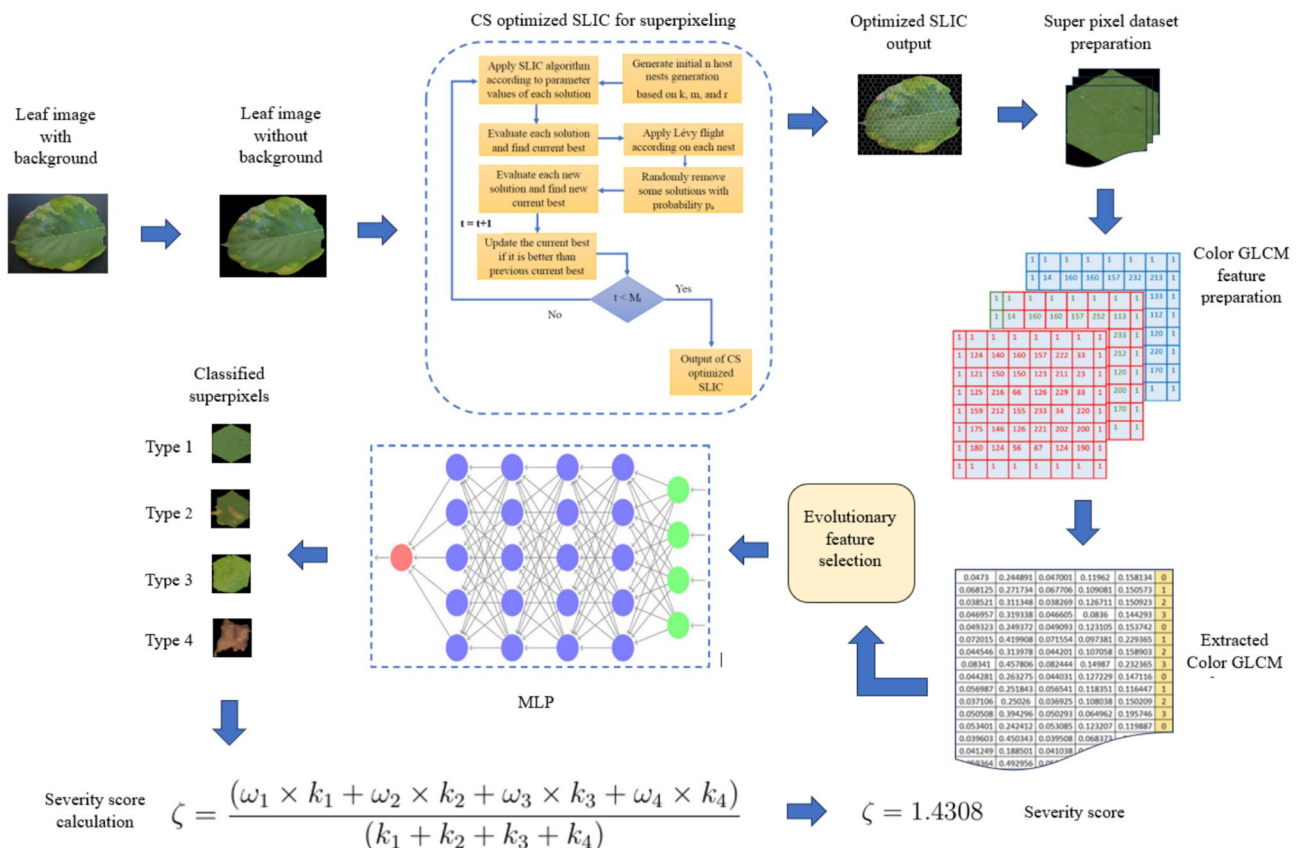
**F1-Score.** The F1-score or F-measure calculates the harmonic mean of precision and recall, providing an equitable assessment of the model's performance. A high F-score signifies that the model exhibits precision (making few errors) and strong recall (identifying the most relevant instances).

$$F1\ Score = 2 \times \frac{Recall \times Precision}{Recall + Precision} \tag{22}$$

$$Error = 1 - Accuracy \tag{23}$$

**Kappa.** ( $\kappa$ ) It provides a measure of the improvement in performance of the classifier compared to a classifier that randomly guesses based on the frequency of each class. Cohen's Kappa coefficient is always less than or equal to 1. Values less than or equal to zero indicate that the classifier is ineffective.

**Matthews correlation coefficient (MCC).** The Matthews correlation coefficient is a more dependable statistical measure that produces a high score only when the prediction performs well in all four categories of the confusion matrix. It is responsible for the dataset's proportion of positive and negative elements.



**Fig. 1.** The flow diagram of the proposed framework.

## Proposed precise chlorosis severity assessment system

The steps that comprise the proposed framework are illustrated in Fig. 1. The stages of the PQCSAF are discussed in the following sections.

### Preprocessing

All leaf images are taken in a controlled environment with a consistent dark-colored background. A preprocessing step eliminates the backdrop and retains only the leaf area as the region of interest. The preprocessing stage has three main steps: global thresholding, mask construction, and leaf extraction. The Green channel  $\hat{G}$  is extracted from the original leaf with background color image  $\hat{I}$  and Otsu's thresholding<sup>59</sup> is applied to estimate  $\hat{T}_h$  for the elimination of the background. The  $\hat{G}_{binary}$  image is generated using image binarization method based on the  $\hat{T}_h$  where the pixel value which is greater or equal than the  $\hat{T}_h$  are considered as 1 otherwise 0.

### Evolutionary superpixel

The primary goal of superpixeling is the delineation of diverse color regions based on the level of chlorosis manifested in the leaf. Consequently, the parameters of the initial SLIC algorithm have been fine-tuned through an evolutionary methodology. This evolutionary strategy aids in the automatic selection of suitable parameter values in accordance with the textural patterns of the diseased leaf by reducing the overall standard deviation in color distribution. In this study, four commonly utilized bio-inspired algorithms, namely Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), and Cuckoo Search (CS) have been employed to facilitate evolutionary superpixels. Among these algorithms, the approach based on Cuckoo search demonstrates superior performance compared to the others. The implementation details of the Cuckoo search-based strategy have been detailed in the current section, and the analysis of the outcomes is provided in the section "Performance analysis of evolutionary superpixel". The steps for optimizing superpixels using the Cuckoo search method are outlined in the algorithm 2. Here, the cuckoo is represented in the form of  $ck_n = \{k, m, r\}$  and initialized within a given range as mentioned Table 3 at Line-1. The Cuckoo search algorithm is a metaheuristic algorithm developed by Yang and Deb in 2009<sup>60</sup>. The algorithm mimics the brood parasitic behaviour of specific cuckoo species. Cuckoo species like Ani and Guira cuckoos follow an aggressive reproduction strategy. It lays eggs in the host bird's nest and destroys the host bird's egg to increase the hatching probability of its eggs. It visits one nest to another by Lévy flights. The recent review of the Cuckoo search algorithm by Guerrero-Luis et al.<sup>61</sup> studied the effectiveness of the Cuckoo search algorithm in diverse domains, including clustering, scheduling, parameter estimation, image analysis, and other various areas. The SLIC algorithm has three parameters: number of superpixels ( $k$ ), weighting factor ( $m$ ), and radius ( $r$ )<sup>34</sup>. The number of superpixels should be chosen so that pixel variance is low within superpixels. The weighting factor,  $m$ , determines how color and spatial information are balanced. A high  $k$ -value signifies the superpixel to have a more regular and smoother form. If one region is less than  $r$ , it will be united with a neighboring region. The textural pattern of a diseased leaf varies depending on the stage of the disease. As a result, it is critical to provide suitable values for all three factors to obtain a compact superpixel capable of calculating the lesion area and stage precisely. Here, the Cuckoo search algorithm has been utilized to optimize the value of  $k$ ,  $m$ , and  $r$ . Here, the summation of the standard deviation of color components of all superpixels has been considered as the objective function as mentioned in Eq. 24. Initial nests are randomly generated within the ranges of  $k$ ,  $m$ , and  $r$ . The range of all these parameters are given in Table 3. From each solution, superpixels are generated using SLIC method. After the generation of the superpixel-based image  $i_{sp}$  and evaluated by the objective function, all solutions are ranked based on fitness. The best cuckoo is calculated from the sorted solutions. New set solution is generated according to the best cuckoo with lévy flight as given in Eqs. 25 and 26. A fraction of the nests are randomly deleted with discovering probability  $p_a$  to bring diversification in the population. All the modified solutions are again used to generate and evaluate superpixel-based images. Finally, the new best cuckoo is compared with the previous one and updated if the new one is better than the previous one. It iterates up to maximum iteration ( $M_t$ ). The Cuckoo search-based optimized parameters finding is outlined in Algorithm 2. Sample healthy leaves and various types of unhealthy leaves are shown in Fig. 4. The optimized value of  $k$ ,  $m$  and  $r$  for each of the samples shown in Fig. 4 are given in Table 4.

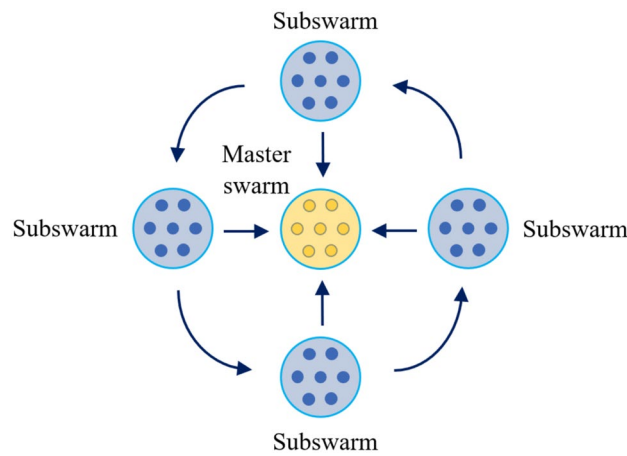
$$f = \sum_{i=1}^k \sum_{j=1}^c i_{sp}(i, j) \quad (24)$$

Parameter Name	Notation	Value Range
No of nests	$n$	10
No of superpixel	$k_{min} - k_{max}$	300–500
Weight factor	$m$	10–40
Radius	$r$	1–3
Discovering probability	$p_a$	0.25
Maximum iteration	$M_t$	50

**Table 3.** Parameters' settings for the cuckoo search used in the proposed method.

Sample No.	No of superpixel ( $k$ )	Weighting factor ( $m$ )	Radius ( $r$ )
1	310	33	1.31
2	315	21	1.31
3	315	28	1.96
4	315	23	1.41
5	309	20	2.47
6	324	31	1.01
7	307	15	2.96
8	321	23	2.15

**Table 4.** Optimized parameters applicable for SLIC for some of the samples (representative examples) as decided by Cuckoo search.



**Fig. 2.** Multi swarm cuckoo search architecture.

$$x_{i+1} = x_i + \alpha \oplus Lévy(\lambda) \tag{25}$$

$$Lévy(u) = t^{-\lambda} \quad \text{where } (1 < \lambda \leq 3) \tag{26}$$

---

**Input:**  $I_{Leaf}$  ← Original leaf image  
 Set all parameters and range  
**Output:**  $I_{Leaf_{spx}}$  ← Superpixel leaf image

- 1 Initialize the  $ck_n$  within the given range
- 2 **while**  $i \leq M_i$  **do**
- 3     Get cuckoo randomly and replace its solution by Equation 25;
- 4     Perform the superpixeling by  $SLIC()$  algorithm;
- 5     Evaluate each solution by Equation 24;
- 6     **if**  $F_i < F_j$  **then**
- 7         Replace solution  $j$ ;
- 8     A fraction ( $p_a$ ) of the least effective nests are discarded, and new ones are generated.
- 9     Preserve the most effective solutions or nests.
- 10    Rank all solutions to identify the best one currently available.
- 11    Transfer the existing best solutions to the future generation.
- 12     $i \leftarrow i + 1$ ;

---

**Algorithm 2.** Cuckoo-Search-Optimized-SLIC.

### Multi-swarm cuckoo search for feature selection

The original Cuckoo search algorithm was a single swarm. In this work, a multi-swarm Cuckoo search has been proposed for feature selection. The total feature set is partitioned into four groups with equal length. Initially, the four subswarms are created to select the feature for four groups, which means each subswarm will take care of each feature group. After a fixed iteration, the master swarm will be constructed from the subswarm. From each subswarm, a fixed number of the cuckoo is represented by a real number between 0 and 1. The general architecture of the proposed multi-swarm Cuckoo search strategy has been shown in Fig. 2, and detailed steps of the algorithm are also mentioned in the Algorithm 3. The following steps are involved in the algorithm.

**Initialization.** In this phase, the four subswarm are randomly initialized by 0 or 1 with length  $k$ . 1 indicates that the particular feature is selected, and 0 indicates the feature is not included in the selected set. So, all subswarms are randomly initialized. However, the master swarm is still not initialized. It is initialised after the  $g$  number of iterations because after each subswarm will be executed  $g$  in parallel. After the  $g$  iterations end, the first  $C_s^{r5}ub$  solutions are selected to form the master swarm.

**Communication Strategy.** In the proposed architecture, two types of communication strategies have been taken. After a fixed interval of iteration, the subswarms return a set of best-performing cuckoos that are performing best up to that iteration in their subswarm. The arrow from the subswarm mentions this to the master swarm in Fig. 2. The symbol ● represents a candidate solution within both the subswarm and the master swarm. After obtaining such a subset, the master swarm is formed and begins executing a fixed iteration. The master swarm is responsible for identifying the set of features that can achieve high accuracy in classifying the four types of superpixels. Information exchange among individual subswarms is also permitted in the multi-swarm framework. Periodically, the subswarms share their few solutions, which helps the algorithm maintain a balance between exploration and exploitation while identifying optimal or near-optimal feature subsets. Here, the term “information exchange” refers to the transfer of binary-encoded feature subsets, where each binary string represents a selected feature set. This periodic communication is illustrated by arrows between the subswarms in Fig. 2. According to the traditional Cuckoo Search rules, there is a probability that the host bird will discover some of the cuckoo nests in each iteration, enabling each subswarm to explore its respective sub-search space independently. Similarly, the master swarm employs the same strategy to explore the combined feature space effectively.

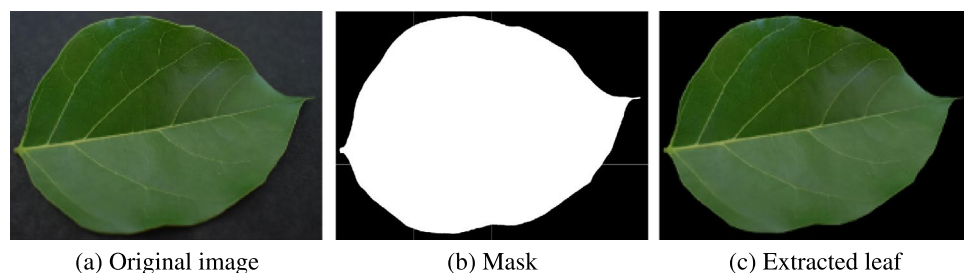


Fig. 3. Preprocessing of images.

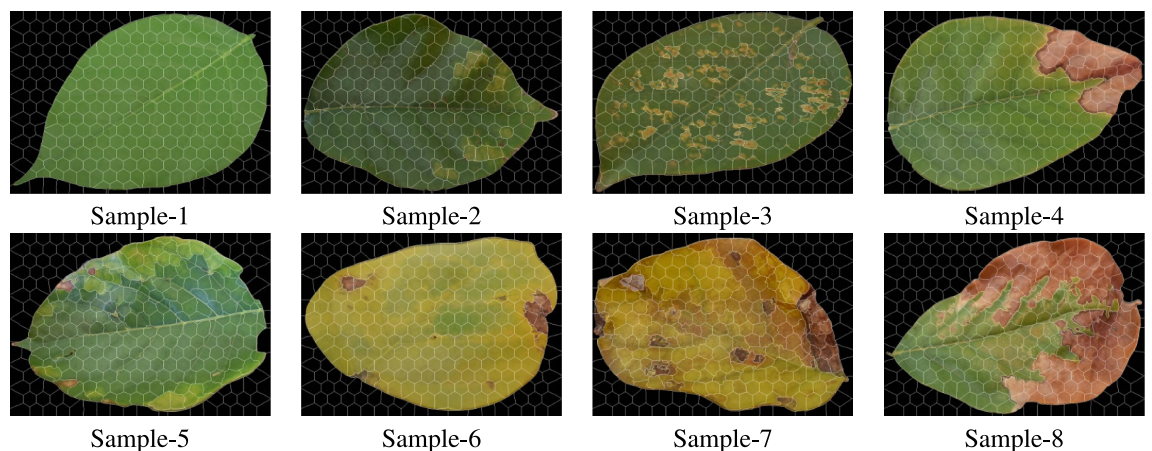


Fig. 4. Superpixels by Cuckoo search optimized SLIC.

---

```

1 Partition the Dataset  $Data$  into  $data_u$  where  $u \in 4$ ;
2 Length of each  $data_u$  is 102;
3 Number of subswarm:  $k$ ; master swarm  $K$ ;
4 Size of each subswarm is  $m$ ;
5 Length of each cuckoo of a subswarm is  $k$ ;
6 Initialize the multiswarm  $C_{ms}^d$ ;
7 for  $b \leftarrow 1$  to  $k$  do
8   Convert each solution of  $C_{sub}^b$  According Equation 27 & 28.
9   Perform the classification, evaluate each solution by 29 and Rank.
10  while  $t \leq max_t$  do
11    Get cuckoo randomly and replace its solution by Equation 25;
12    Perform the classification, evaluate each solution by 29 and Rank.
13    Choose a random cuckoo  $j$  randomly.
14    if  $F_j^b < F_i^b$  then
15      Replace solution  $i$ ;
16    A fraction ( $p_a$ ) of the least effective nests are removed, and new ones are constructed.
17    Retain the most effective solutions or nests.
18    Rank the solutions to identify the current best.
19    Transfer the best current solutions to the next generation.
20     $t \leftarrow t + 1$ ;
21 Find the  $C_{sub}^{r5}$  from  $k$  subswarm;
22 Prepare  $C_{master}$  by combining  $C_{sub}^{r5}$ ;
    // Master module of cuckoo swarm constructed from subswarm.
23 while  $i \leq m_t$  do
24   Get cuckoo randomly and replace its solution by Equation 25;
25   Convert each solution of  $C_{master}$  according to Equation 27 & 28.
26   Perform the classification, evaluate each solution by 29 and Rank.
27   if  $F_j < F_i$  then
28     Replace solution  $i$ ;
29   A fraction ( $p_a$ ) of the least effective nests are removed, and new ones are constructed.
30   Retain the most effective solutions or nests.
31   Rank the solutions to identify the current best.
32   Transfer the best current solutions to the next generation.
33    $i \leftarrow i + 1$ ;

```

---

**Algorithm 3.** Multi-swarm parallel binary Cuckoo search for feature selection.

$$S(x_i^k(t)) = \frac{1}{1 + e^{-x_i^k(t)}} \quad (27)$$

$$S_{t+1}^k = \begin{cases} 1, & \text{if } S(x_i^k(t)) > \sigma. \\ 0, & \text{otherwise.} \end{cases} \quad (28)$$

$$Fitness = (1 - accuracy) \quad (29)$$

### Quantification of chlorosis through severity index

The chlorosis severity score,  $\zeta$ , as proposed by us, is the weighted average of the scores concerning all the superpixels in the population as shown in Eq. 30. Here,  $k^1$  denotes the count of the Type-1 superpixels coming from the leaf. Similarly,  $k^2$ ,  $k^3$ , and  $k^4$  denote the counts of Type-2, Type-3, and Type-4 superpixels, respectively, from the leaf. Now, each Type of superpixels is assigned a weight  $\omega_j \in \{1, 2, 3, 4\}$ , where  $j \in \{1, 2, 3, 4\}$ . The

severity index calculation does not consider a superpixel with a size of less than seventy (70) pixels. The severity index  $\zeta$  will lie in the range [1, 4]. The value of  $\zeta$  towards 1 indicates less or negligible chlorosis effect, whereas a higher value of  $\zeta$  indicates the presence of a higher degree of chlorosis. We can use Eq. 30 to collectively measure the chlorosis severity for a random group of plant leaves. Instead of individual leaves, a severity score can be measured using Eq. 31 where  $N_{sp}$  denotes the total number of sample leaves taken from the field.

$$\zeta = \frac{(\omega_1 \times k^1 + \omega_2 \times k^2 + \omega_3 \times k^3 + \omega_4 \times k^4)}{(k^1 + k^2 + k^3 + k^4)} \quad (30)$$

$$\zeta = \frac{(\omega_1 \sum_{i=1}^{N_{sp}} k_i^1 + \omega_2 \sum_{i=1}^{N_{sp}} k_i^2 + \omega_3 \sum_{i=1}^{N_{sp}} k_i^3 + \omega_4 \sum_{i=1}^{N_{sp}} k_i^4)}{(\sum_{i=1}^{N_{sp}} k_i^1 + \sum_{i=1}^{N_{sp}} k_i^2 + \sum_{i=1}^{N_{sp}} k_i^3 + \sum_{i=1}^{N_{sp}} k_i^4)} \quad (31)$$

## Results and discussion

This section provides a comprehensive analysis of the experimental findings for each stage of the PQCSAF.

### Leaf area extraction

The leaf dataset that is utilized here has a fixed background. Hence, a preprocessing step has been applied to remove the background and keep only the leaf area as the region of interest. The preprocessing stage has three main steps: global thresholding, mask construction, and leaf extraction. The picture shown in Fig. 3a is obtained from the dataset, while the mask displayed in Fig. 3b is constructed using the previously discussed thresholding approach. Fig. 3c represents the ultimate extracted leaf area employed for the superpixeling process in the subsequent phase.

The classification problem in this work has been designed as a multi-class classification problem. Classification models were created using DT, KNN, SVM, and MLP for the work. The models were tested with a  $k$ -fold cross-validation strategy. We considered  $k = 40$  for the experiment.

### Performance analysis of evolutionary superpixel

The parameters of the SLIC algorithm have been optimized using an evolutionary technique and applied to the extracted leaf area. The CS-optimized SLIC outputs of Sample-1 to 8 are illustrated in Fig. 4. The visual results show that the superpixels should be adaptable to correctly capture the lesion depending on the affected region and yellowing. Indeed, the outputs of Fig. 4 are produced by the optimized parameter values in Table 4. The efficacy of the CS in optimizing the superpixel parameters has been compared to that of the other three metaheuristic algorithms: Genetic Algorithm (GA), Particle Swarm Optimization (PSO), and Ant Colony Optimization (ACO). Figure 5a displays the Convergence curve of GA, PSO, ACO, and CS algorithms for optimizing parameters for sample-6. The performance of the CS algorithm surpasses that of the other three methods. Figure 5b shows the convergence curve for all the samples 1 to 8. Based on the graph, it is clear that when the leaf has a minimal color variation (either a small affected area or less variation in yellowing), the convergence curve does not show significant improvement with each iteration. Conversely, in the opposite scenario, the convergence graph significantly improves as the iterations progress.

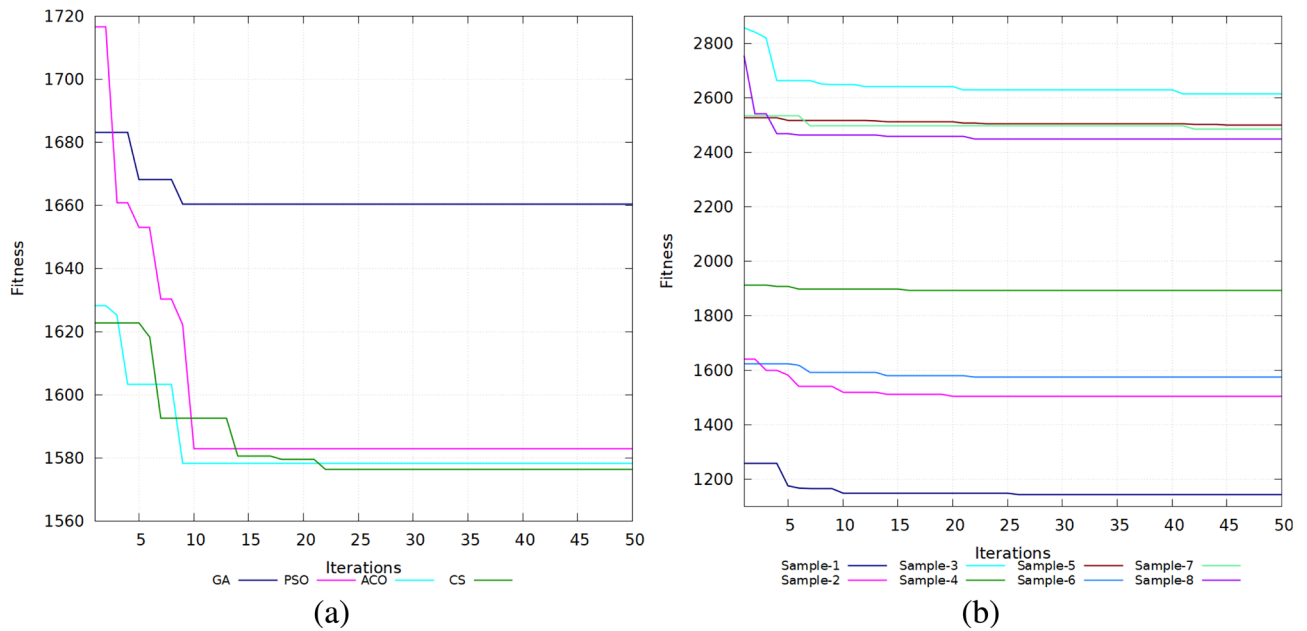
### Performance analysis of multi-swarm cuckoo search for feature selection

The efficacy of the multi-swarm Cuckoo search for feature selection has been discussed in this section. The Table 5 provides a comparative analysis of five algorithms (GA, PSO, ACO, CS, and MSCS) in terms of their performance in “Fitness” and “Feature” over the iterations. It has been noted that CS achieves the greatest fitness value of 0.0046, while MSCS attains the value of 0.0095. Therefore, MSCS does not surpass the CS in terms of fitness value, nor does it surpass the GA, PSO, or ACO. However, MSCS has the lowest standard deviation of 0.00037, indicating it produces very consistent results in terms of fitness. MSCS has obtained 73 features, the lowest compared to other algorithms. The standard deviation of MSCS is 9, the second smallest value. Figure 6a and b illustrate the convergence curve and the number of selected features for each method, respectively. It is clear from the discussion that CS has the highest fitness performance, but its feature efficiency is only modest. The performance of MSCS is balanced and characterized by the utmost efficacy in feature selection, very low variability in results, and reasonable fitness values.

### Performance analysis of classification on selected features from affected superpixels

An initial analysis is conducted on a collection of 408 original features derived from four superpixel categories. The DT, KNN, SVM, and MLP classifiers achieved average classification accuracy of 87.10%, 91.00%, 94.50%, and 96.80% correspondingly for the category of superpixels, as indicated in Table 6. The MLP has demonstrated superior performance to the other three classifiers when evaluated using multiple metrics on the original feature set. The evaluation measures are outlined in Table 6. The remaining part of this section will investigate classifier behavior following the implementation of five evolutionary feature selection methodologies.

The classifiers DT, KNN, SVM, and MLP are trained and tested using the features obtained by GA, PSO, ACO, CS, and MSCS. In addition, the classification is conducted on the components generated by the PCA when applied to the original feature set. The classification result is assessed using the following performance metrics: Accuracy, Error, Sensitivity, Specificity, and Precision. The average performance of the DT classifier on all selected feature sets for classifying four categories of superpixels is shown in Fig. 7a. DT obtains 87.8% accuracy based on the ACO selected feature set, indicating it is the most effective in correctly classifying instances. At the same time, the lowest accuracy obtained by DT is based on PCA components. Hence, the error rate is lowest in the case of ACO and highest in PCA. The DT obtains 87.8%, 95.93%, and 87.87% for Sensitivity, Specificity,

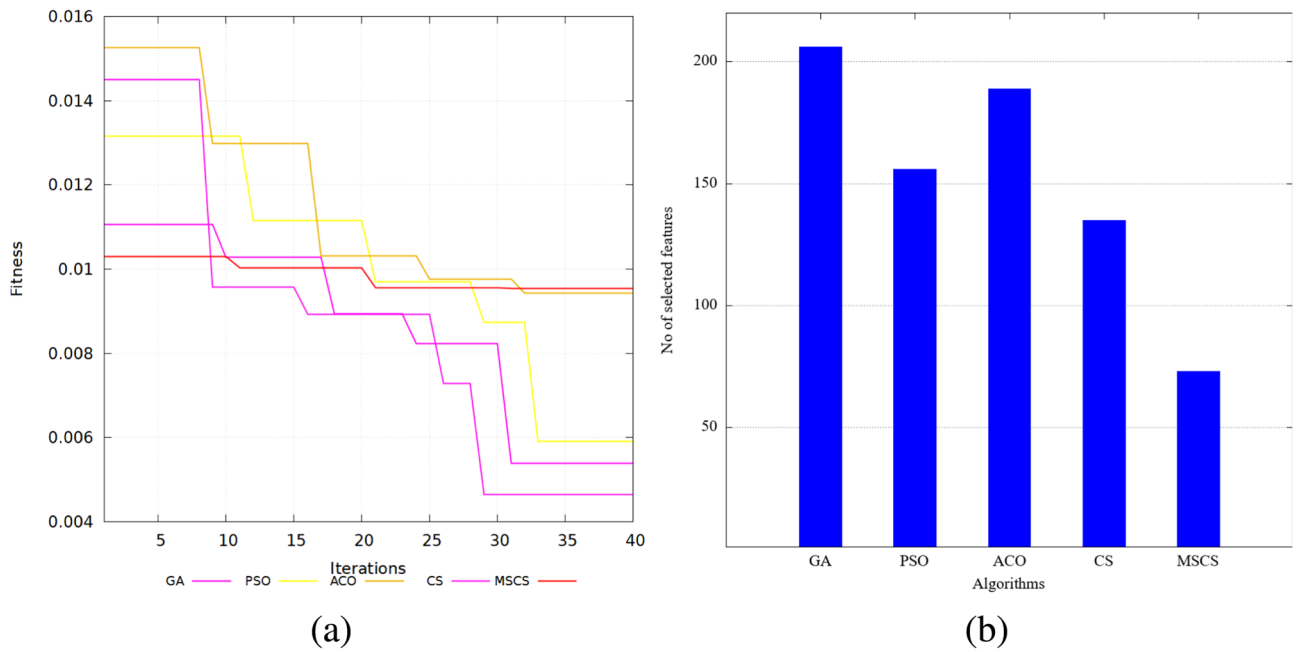


**Fig. 5.** Convergence plots (#iterations vs fitness value) of optimized superpixels **a** (for the Sample-06 shown in Fig. 4) for various methods and **b** (for all the samples shown in Fig. 4) for Cuckoo search.

Algorithms	Fitness			#Feature		
	Mean	STD	Best	Mean	STD	Best
GA	0.0081	0.0016	0.0054	210	8	206
PSO	0.0096	0.0019	0.0059	142	49	156
ACO	0.0105	0.0015	0.0094	232	17	189
CS	0.0088	0.0021	0.0046	172	38	135
MSCS	0.0099	0.00037	0.0095	72	9	73

**Table 5.** Performance of MSCS in feature selection with other different algorithms.

and Precision, respectively, based on ACO-obtained features. However, PCA-based performances by DT on the previous three parameters are again the lowest compared to the other methods. Therefore, the ACO selected features set performs very well with the DT classifier. At the same time, DT obtains the lowest performance with PCA. GA, PSO, CS, and MSCS perform well but do not outperform ACO. GA and CS perform slightly better than PSO and MSCS in most metrics. Figure 7b displays the mean performance of the KNN classifier across all chosen feature sets for categorizing four types of superpixels. The KNN classifier achieves a maximum accuracy of 91.3% when utilizing GA based feature selection. On the other hand, the lowest accuracy of 88.9% is obtained when employing PCA. The performance of KNN using different feature selection methods falls within the range of the lowest and highest accuracy achieved. That suggests the level of accuracy is relatively consistent. The error rate associated with GA-based methods is the lowest, at 8.7%, while PCA-based methods have the highest error rate, at 11.1%. The GA-based and ACO-based models exhibit the highest sensitivity, with values of 91.3% and 91.1%, respectively. The sensitivity of PCA-based methods is the lowest, at 88.9%. The ACO-based technique achieves the maximum specificity, with a value of 97.03%, while the PCA-based method has the lowest specificity, with a value of 96.3%. The remaining technique similarly exhibits a high level of specificity, falling within the performance range of ACO and PCA. When using PCA, the precision of the KNN classifier is lowest at 89.42% and maximum at 91.43% when using a GA-based technique. The GA is the most successful approach for the KNN classifier, achieving the best accuracy and lowest error rates. Additionally, it demonstrates good sensitivity, specificity, and precision. PCA exhibits lower performance in this context. Furthermore, ACO and PSO are also formidable competitors. To categorize four different kinds of superpixels, Fig. 7c shows the average performance of the SVM classifier across all selected feature sets. The obtains the highest accuracy at 95.0% MSCS-based method and the lowest at 93.1% with PCA. Error is also minimal with the MSCS-based method. SVM also achieves a high sensitivity of 95.0%, a specificity of 98.33%, and a precision of 94.99%. Similarly to the MSCS-based method, SVM's efficacy in sensitivity, specificity, and precision is comparable to that of the other evolutionary feature selection techniques. MSCS-based feature selection is the most dependable method for the SVM classifier in this particular condition. MSCS may be the preferred option to maximize the performance of their SVM classifier; however, GA, PSO, and ACO also exhibit comparable performance to

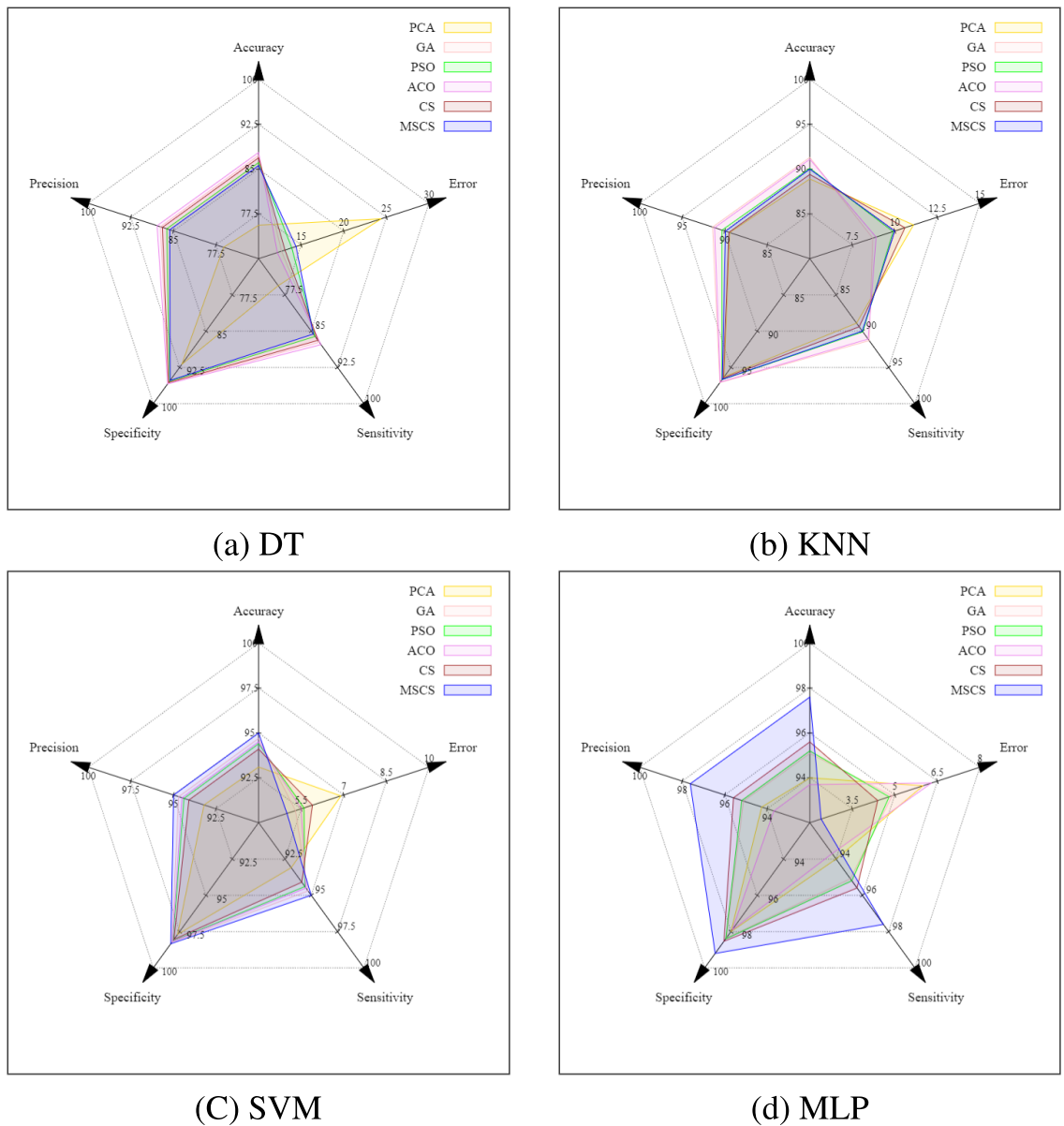


**Fig. 6.** **a** Convergence curve; **b** No of selected features.

Classifier	Type	Acc.(%)	Err.(%)	Sensit.	Specif.	Prec.	FPR	F1-Sc.	MCC
DT	Type-1	90.40	9.60	0.9040	0.9680	0.904	0.0320	0.9040	0.8720
	Type-2	82.00	18.00	0.8200	0.9480	0.8402	0.0520	0.8300	0.7743
	Type-3	88.00	12.00	0.8800	0.9480	0.8494	0.0520	0.8644	0.8184
	Type-4	88.00	12.00	0.8800	0.9640	0.8907	0.0360	0.8853	0.8474
	Avg	87.10	12.90	0.8710	0.9570	0.8711	0.0430	0.8709	0.8280
KNN	Type-1	98.4	1.6	0.984	0.964	0.9011	0.036	0.9407	0.9214
	Type-2	80.4	19.60	0.804	0.9733	0.9095	0.0267	0.8535	0.8112
	Type-3	90.0	10.00	0.9	0.9813	0.9414	0.0187	0.9202	0.8948
	Type-4	95.20	4.80	0.952	0.9613	0.8914	0.0387	0.9207	0.894
	Avg	91.00	9.00	0.91	0.97	0.9109	0.03	0.9088	0.8804
SVM	Type-1	98.00	2.00	0.98	0.9853	0.957	0.0147	0.9684	0.9578
	Type-2	91.6	8.40	0.916	0.976	0.9271	0.024	0.9215	0.8956
	Type-3	92.4	7.60	0.924	0.9773	0.9315	0.0227	0.9277	0.9038
	Type-4	96.00	4.00	0.9600	0.988	0.9639	0.012	0.9619	0.9493
	Avg	94.50	5.50	0.94500	0.9817	0.9449	0.0183	0.9449	0.9266
MLP	Type-1	99.2	0.80	0.992	0.9933	0.9802	0.0067	0.9861	0.9814
	Type-2	94.8	5.20	0.948	0.9813	0.9442	0.0187	0.9461	0.9281
	Type-3	96.4	3.60	0.964	0.9920	0.9757	0.008	0.9698	0.9599
	Type-4	96.8	3.20	0.968	0.9907	0.9719	0.0093	0.9699	0.9600
	Avg	96.8	3.20	0.968	0.9893	0.968	0.0107	0.968	0.9573

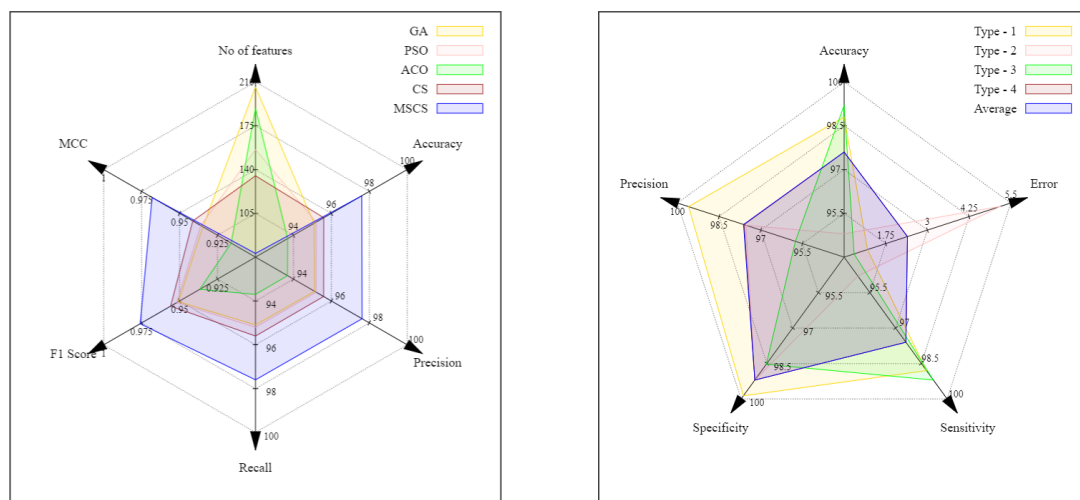
**Table 6.** Performance of different classifiers on complete dataset of 1000 samples of superpixels without feature selection.

MSCS. The average performance of the MLP classifier in categorizing four categories of superpixels across all selected feature sets is illustrated in Fig. 7d. MLP classifier has obtained the highest accuracy of 97.60% and minimum error of 2.4% using the MSCS-based method, whereas using the ACO-based method, MLP has obtained the lowest accuracy of 93.7% and highest error of 6.3%. Using the MSCS-based technique, the MLP classifier regularly achieves the greatest sensitivity value of 97.6%, the highest specificity of 99.2%, and the best precision of 97.62%. The MLP classifier, using the ACO-based technique, obtains the lowest performance in sensitivity of 93.7%, specificity of 97.9%, and precision of 93.69%. Additionally, other methodologies are strong competitors in terms of a few parameters. The above classification results analysis, which includes all feature selection strategies, indicates that the MLP with an MSCS-based strategy consistently performs well



**Fig. 7.** Average classification performance of different classifiers on selected features by various feature selection strategy.

under average conditions in all five parameters: accuracy, error, sensitivity, specificity, and precision. A more comprehensive analysis of the performance of MLP classification has been shown in Fig. 8. The performance of MLP with five evolutionary algorithms-based feature selection strategies for the average case is illustrated in Fig. 8a. The five metrics have been considered: number of features, accuracy, precision, recall, F1 score, and MCC. The MSCS algorithm utilizes the minimum number of features, precisely 73, whereas the GA algorithm employs the maximum number of features, 206. The quantity of features significantly influences the classifier’s performance. The MSCS-based technique has a maximum accuracy of 97.6%, while the CS method achieves an accuracy of 95.6%. The accuracy of ACO is 93.7%, which is the lowest of the options. The average classification accuracy with MSCS is notable, using minimal features, indicating the effectiveness of feature selection. MSCS exhibits the highest precision (97.62%), suggesting that it generates minimal false positives. The precision of ACO is the lowest at 93.69%. Recall signifies the actual positives correctly identified instances. MSCS achieved the maximum recall (97.6%), while ACO achieved the lowest (93.7%). The harmonic mean of precision and recall is the F1 score, which balances the two. MSCS has the highest F1 score of 0.9759, suggesting robust stability between precision and recall. ACO has the lowest F1 score of 0.9367. MCC is a metric that evaluates the quality of classifications in the context of true and false classifications. The MSCS-based method has the highest MCC of 0.97, which generates the most dependable classification results. The lowest MCC is 0.92 for the ACO-based strategy. The performance of MLP for the average case and individual superpixel category is illustrated in Fig. 8b. The MLP classifier demonstrates high accuracy in all categories, with the highest accuracy observed in Type - 3



(a) Different FS strategy with parameters. (b) Superpixel category with parameters.

**Fig. 8.** Specific performance of MLP classifier with various parameters.

(99.2%) and the lowest in Type - 2 (94.8%). The performance is robust, as evidenced by the average accuracy of 97.6%. The error rates are inversely proportional to the accuracy rates. Type 3 has the lowest error rate at 0.8%, while Type-2 has the highest at 5.2%. The average error rate is a mere 2.4%. The sensitivity values demonstrate the classifier's high effectiveness in distinguishing positive instances, which is consistent with the accuracy values. The sensitivity values are highest for Type-3 (99.2%), lowest for Type-2 (94.8%), and average (97.6%). The classifier performs exceptionally well in reliably identifying negative instances, with consistently high specificity values over 98% for all types. Type-1 has the maximum level of specificity, measuring at 99.87%, while the Average demonstrates a specificity of 99.2%. The precision values for all types are consistently high, showing that the MLP classifier has a very low rate of false positive errors. Type-1 exhibits the highest level of precision, at 99.60%. Conversely, Type-3 has the lowest precision, at 95.75%. The average precision value is 97.62%. Furthermore, the comprehensive and detailed information on classifier performances for average and individual categorization with different feature selection strategies for assessing metrics is provided in Table 7. Overall, MSCS-selected features achieve the highest accuracy (97.6%), precision (97.62%), recall (97.6%), F1 score (0.9760), and MCC (0.97) compared to other approaches. This performance implies that MSCS selects features effectively, leading to better classification performance with only 73 features in the average case and for each category. Moreover, MSCS is the most effective and efficient feature selection technique for the MLP classifier in this superpixel categorization task.

#### Ablation study with four classifiers applied to original and MSCS feature sets

A comparative investigation of classification performance was performed across four classifiers: DT, KNN, SVM, and MLP, utilizing both the original and MSCS-selected feature sets. The initial feature set had 408 features; however, the MSCS approach reduced this to only 73 features. The evaluated performance criteria comprise accuracy, sensitivity, specificity, and precision. The original feature set for the DT classifier marginally surpasses the MSCS-selected features, exhibiting a minor decline in all measures when employing MSCS. On the other hand, KNN demonstrates consistent performance across all feature sets, with just a slight decrease in sensitivity and precision when utilizing MSCS, which suggests robustness to feature reduction. The SVM demonstrates a marginal enhancement in accuracy, sensitivity, and specificity when utilizing MSCS-selected features, indicating that the model advantages from a more concise and relevant feature set. MLP consistently attains superior performance across all measures, with enhancements observed when employing MSCS-selected features. Accuracy increased from 96.8% to 97.6%, sensitivity from 96.8% to 97.6%, specificity from 98.93% to 99.2%, and precision from 96.8% to 97.62%. The performance comparison graphs between the original and MSCS-selected feature set for each classifier are illustrated in Fig. 9, with (a) denoting DT, (b) KNN, (c) SVM, and (d) MLP. It indicates that the MSCS method not only reduces computational complexity but also enhances classification performance.

#### Severity score generation

The four categories of superpixels, classified by their degree of severity, are illustrated in Fig. 10. Finally, the severity score is computed using Eq. 30. Figure 11 shows twelve diseased images and their disease severity scores. The computation of the severity score  $\zeta$  using Eq. 30 for the examples shown in Fig. 11 is explained in detail in Table 8. Furthermore, Fig. 12 displays the scores of the initial twenty-five sample leaf pictures on a scale ranging from 1 to 4. The overall severity score  $\zeta$  for all 25 sample leaves, as seen in Fig. 12, has been computed using Eq. 31 and yielded a value of 2.0185.

Method	Classifier	Type	Acc.	Err.	Sensit.	Specif.	Prec.	FPR	F1-Sc.	MCC
PCA	DT	Type-1	81.20	18.80	0.8120	0.9547	0.8565	0.0450	0.8337	0.7807
		Type-2	68.40	31.60	0.6840	0.8573	0.6151	0.1430	0.6477	0.5232
		Type-3	77.60	22.40	0.7760	0.9347	0.7984	0.0650	0.7870	0.7175
		Type-4	75.20	24.80	0.7520	0.9280	0.7769	0.0720	0.7642	0.6875
		Average	75.60	24.40	0.7560	0.9187	0.7617	0.0810	0.7582	0.6772
	KNN	Type-1	97.60	2.40	0.9760	0.9587	0.8873	0.0410	0.9295	0.9064
		Type-2	88.80	11.20	0.8880	0.9280	0.8043	0.0720	0.8441	0.7904
		Type-3	86.00	14.00	0.8600	0.9773	0.9267	0.0230	0.8921	0.8590
		Type-4	83.20	16.80	0.8320	0.9880	0.9585	0.0120	0.8908	0.8614
		Average	88.90	11.10	0.8890	0.9630	0.8942	0.0370	0.8891	0.8543
	SVM	Type-1	99.20	0.80	0.9920	0.9853	0.9575	0.0150	0.9745	0.9660
		Type-2	91.20	8.80	0.9120	0.9547	0.8702	0.0450	0.8906	0.8534
		Type-3	91.60	8.40	0.9160	0.9800	0.9385	0.0200	0.9271	0.9033
		Type-4	90.40	9.60	0.9040	0.9880	0.9617	0.0120	0.9320	0.9110
		Average	93.10	6.90	0.9310	0.9770	0.9320	0.0230	0.9310	0.9084
	MLP	Type-1	97.20	2.80	0.9720	0.9947	0.9838	0.0050	0.9779	0.9706
		Type-2	94.00	6.00	0.9400	0.9480	0.8577	0.0520	0.8969	0.8621
		Type-3	94.00	6.00	0.9400	0.9893	0.9671	0.0110	0.9533	0.9383
		Type-4	90.80	9.20	0.9080	0.9880	0.9619	0.0120	0.9342	0.9137
		Average	94.00	6.00	0.9400	0.9800	0.9426	0.0200	0.9406	0.9212
GA	DT	Type-1	89.60	10.40	0.8960	0.9653	0.8960	0.0350	0.8960	0.8613
		Type-2	84.00	16.00	0.8400	0.9467	0.8400	0.0530	0.8400	0.7867
		Type-3	84.80	15.20	0.8480	0.9520	0.8548	0.0480	0.8514	0.8022
		Type-4	90.80	9.20	0.9080	0.9667	0.9008	0.0330	0.9044	0.8724
		Average	87.30	12.70	0.8730	0.9577	0.8729	0.0420	0.8729	0.8306
	KNN	Type-1	98.80	1.20	0.9880	0.9680	0.9114	0.0320	0.9482	0.9313
		Type-2	79.20	20.80	0.7920	0.9773	0.9209	0.0230	0.8516	0.8109
		Type-3	90.80	9.20	0.9080	0.9813	0.9419	0.0190	0.9246	0.9004
		Type-4	96.40	3.60	0.9640	0.9573	0.8828	0.0430	0.9216	0.8955
		Average	91.30	8.70	0.9130	0.9710	0.9143	0.0290	0.9115	0.8845
	SVM	Type-1	97.60	2.40	0.9760	0.9867	0.9606	0.013	0.9683	0.9576
		Type-2	91.60	8.40	0.9160	0.9773	0.9309	0.0230	0.9234	0.8982
		Type-3	94.00	6.00	0.9400	0.9773	0.9325	0.0230	0.9363	0.9149
		Type-4	95.60	4.40	0.9560	0.9880	0.9637	0.0120	0.9598	0.9465
		Average	94.70	5.30	0.9470	0.9823	0.9469	0.0180	0.9469	0.9293
	MLP	Type-1	98.40	1.60	0.9840	0.9799	0.9425	0.0200	0.9628	0.9504
		Type-2	94.00	6.00	0.9400	0.9772	0.9325	0.0230	0.9363	0.9148
		Type-3	91.60	8.40	0.9160	0.9893	0.9662	0.0110	0.9405	0.9218
		Type-4	96.36	3.64	0.9636	0.9880	0.9636	0.0120	0.9636	0.9516
		Average	95.09	4.91	0.9509	0.9836	0.9512	0.0160	0.9508	0.9346
Continued										

Method	Classifier	Type	Acc.	Err.	Sensit.	Specif.	Prec.	FPR	F1-Sc.	MCC
PSO	DT	Type-1	88.00	12.00	0.8800	0.9680	0.9016	0.0320	0.8907	0.8549
		Type-2	82.00	18.00	0.8200	0.9320	0.8008	0.0680	0.8103	0.7461
		Type-3	85.20	14.80	0.8520	0.9480	0.8452	0.0520	0.8486	0.7979
		Type-4	89.20	10.80	0.8920	0.9667	0.8992	0.0330	0.8956	0.8610
		Average	86.10	13.90	0.8610	0.9537	0.8617	0.0460	0.8613	0.8150
	KNN	Type-1	98.40	1.60	0.9840	0.9587	0.8881	0.0410	0.9336	0.9121
		Type-2	76.40	23.60	0.7640	0.9760	0.9139	0.0240	0.8322	0.7881
		Type-3	89.60	10.40	0.8960	0.9813	0.9412	0.0190	0.9180	0.8921
		Type-4	96.00	4.00	0.9600	0.9520	0.8696	0.0480	0.9125	0.8834
		Average	90.10	9.90	0.9010	0.9670	0.9032	0.0330	0.8991	0.8689
	SVM	Type-1	97.60	2.40	0.9760	0.9840	0.9531	0.0160	0.9644	0.9525
		Type-2	91.60	8.40	0.9160	0.9773	0.9309	0.0230	0.9234	0.8982
		Type-3	92.00	8.00	0.9200	0.9773	0.9312	0.0230	0.9256	0.9010
		Type-4	96.40	3.60	0.9640	0.9867	0.9602	0.0130	0.9621	0.9494
		Average	94.40	5.60	0.9440	0.9813	0.9438	0.0190	0.9439	0.9253
	MLP	Type-1	98.00	2.00	0.9800	0.9827	0.9496	0.0170	0.9646	0.9527
		Type-2	93.60	6.40	0.9360	0.9800	0.9398	0.0200	0.9379	0.9172
		Type-3	92.40	7.60	0.9240	0.9840	0.9506	0.0160	0.9371	0.9167
		Type-4	96.00	4.00	0.9600	0.9867	0.9600	0.0130	0.9600	0.9467
		Average	95.20	4.80	0.9520	0.9840	0.9520	0.0160	0.9519	0.9360
ACO	DT	Type-1	90.80	9.20	0.9080	0.9720	0.9153	0.0280	0.9116	0.8824
		Type-2	86.40	13.60	0.8640	0.9400	0.8276	0.0600	0.8454	0.7927
		Type-3	84.40	15.60	0.8440	0.9533	0.8577	0.0470	0.8508	0.8017
		Type-4	89.60	10.40	0.8960	0.9720	0.9143	0.028	0.9051	0.8739
		Average	87.80	12.20	0.8780	0.9593	0.8787	0.0410	0.8782	0.8377
	KNN	Type-1	98.80	1.20	0.9880	0.9640	0.9015	0.036	0.9427	0.9243
		Type-2	80.80	19.20	0.8080	0.9747	0.9140	0.0250	0.8577	0.8168
		Type-3	88.80	11.20	0.8880	0.9800	0.9367	0.0200	0.9117	0.8839
		Type-4	96.00	4.00	0.9600	0.9627	0.8955	0.0370	0.9266	0.9020
		Average	91.10	8.90	0.9110	0.9703	0.9119	0.0300	0.9097	0.8817
	SVM	Type-1	98.40	1.60	0.9840	0.9840	0.9535	0.0160	0.9685	0.9580
		Type-2	91.60	8.40	0.9160	0.9773	0.9309	0.023	0.9234	0.8982
		Type-3	92.00	8.00	0.9200	0.9760	0.9274	0.0240	0.9237	0.8984
		Type-4	96.00	4.00	0.9600	0.9893	0.9677	0.0110	0.9639	0.9519
		Average	94.50	5.50	0.9450	0.9817	0.9449	0.0180	0.9449	0.9266
	MLP	Type-1	98.40	1.60	0.9840	0.9760	0.9318	0.0240	0.9572	0.943
		Type-2	90.00	10.00	0.9000	0.976	0.9259	0.024	0.9128	0.8844
		Type-3	90.40	9.60	0.9040	0.9773	0.9300	0.0230	0.9168	0.8898
		Type-4	96.00	4.00	0.9600	0.9867	0.9600	0.0130	0.9600	0.9467
		Average	93.70	6.30	0.9370	0.9790	0.9369	0.0210	0.9367	0.9160
Continued										

Method	Classifier	Type	Acc.	Err.	Sensit.	Specif.	Prec.	FPR	F1-Sc.	MCC
CS	DT	Type-1	89.60	10.40	0.8960	0.9573	0.8750	0.0430	0.8854	0.8467
		Type-2	82.80	17.20	0.8280	0.9453	0.8347	0.0550	0.8313	0.7754
		Type-3	86.80	13.20	0.8680	0.9493	0.8510	0.0510	0.8594	0.8120
		Type-4	88.40	11.60	0.8840	0.9733	0.9170	0.0270	0.9002	0.8680
		Average	86.90	13.10	0.8690	0.9563	0.8694	0.0440	0.8691	0.8255
	KNN	Type-1	98.40	1.60	0.9840	0.9627	0.8978	0.0370	0.9389	0.9191
		Type-2	76.00	24.00	0.7600	0.9707	0.8962	0.0290	0.8225	0.7741
		Type-3	88.40	11.60	0.8840	0.9760	0.9247	0.0240	0.9039	0.8732
		Type-4	94.80	5.20	0.9480	0.9493	0.8618	0.0510	0.9029	0.8702
		Average	89.40	10.60	0.8940	0.9647	0.8951	0.0350	0.8920	0.8591
	SVM	Type-1	97.60	2.40	0.9760	0.9813	0.9457	0.0190	0.9606	0.9474
		Type-2	92.00	8.00	0.9200	0.9760	0.9274	0.0240	0.9237	0.8984
		Type-3	90.80	9.20	0.9080	0.9760	0.9265	0.0240	0.9172	0.8900
		Type-4	96.00	4.00	0.9600	0.9880	0.9639	0.0120	0.9619	0.9493
		Average	94.10	5.90	0.9410	0.9803	0.9409	0.0200	0.9409	0.9213
	MLP	Type-1	98.80	1.20	0.988	0.9867	0.9611	0.0130	0.9744	0.9658
		Type-2	92.80	7.20	0.9280	0.9840	0.9508	0.0160	0.9393	0.9195
		Type-3	93.60	6.40	0.9360	0.9813	0.9435	0.0190	0.9398	0.9198
		Type-4	97.20	2.80	0.9720	0.9893	0.9681	0.0110	0.9701	0.9601
		Average	95.60	4.40	0.9560	0.9853	0.9559	0.0150	0.9559	0.9413
MSCS	DT	Type-1	88.80	11.20	0.8880	0.9587	0.8775	0.0413	0.8827	0.8433
		Type-2	78.40	21.60	0.7840	0.9493	0.8376	0.0507	0.8099	0.7500
		Type-3	86.40	13.60	0.8640	0.9333	0.8120	0.0667	0.8372	0.7814
		Type-4	88.80	11.20	0.8880	0.9667	0.8989	0.0333	0.8934	0.8581
		Average	85.60	14.40	0.8560	0.9520	0.8565	0.0480	0.8558	0.8082
	KNN	Type-1	98.40	1.60	0.9840	0.9760	0.9318	0.0240	0.9572	0.9430
		Type-2	81.20	18.80	0.8120	0.9533	0.8529	0.0467	0.8320	0.7782
		Type-3	90.80	9.20	0.9080	0.9747	0.9228	0.0253	0.9153	0.8875
		Type-4	89.60	10.40	0.8960	0.9627	0.8889	0.0373	0.8924	0.8564
		Average	90.00	10.00	0.9000	0.9667	0.8991	0.0333	0.8992	0.8663
	SVM	Type-1	98.00	2.00	0.9800	0.9907	0.9722	0.0093	0.9761	0.9681
		Type-2	92.00	8.00	0.9200	0.9760	0.9274	0.0240	0.9237	0.8984
		Type-3	93.60	6.40	0.9360	0.9800	0.9398	0.0200	0.9379	0.9172
		Type-4	96.40	3.60	0.9640	0.9867	0.9602	0.0133	0.9621	0.9494
		Average	95.00	5.00	0.9500	0.9833	0.9499	0.0167	0.9499	0.9333
	MLP	Type-1	98.80	1.20	0.9880	0.9987	0.9960	0.0013	0.9920	0.9893
		Type-2	94.80	5.20	0.9480	0.9920	0.9753	0.0080	0.9615	0.9490
		Type-3	99.20	0.80	0.9920	0.9853	0.9575	0.0147	0.9745	0.9660
		Type-4	97.60	2.40	0.9760	0.9920	0.9760	0.0080	0.9760	0.9680
		Average	<b>97.60</b>	<b>2.40</b>	<b>0.9760</b>	<b>0.9920</b>	<b>0.9762</b>	<b>0.0080</b>	<b>0.9760</b>	<b>0.9681</b>

**Table 7.** Performance of different classifiers based on the selected features by various bio-inspired algorithms on complete dataset of 1000 samples of superpixels.

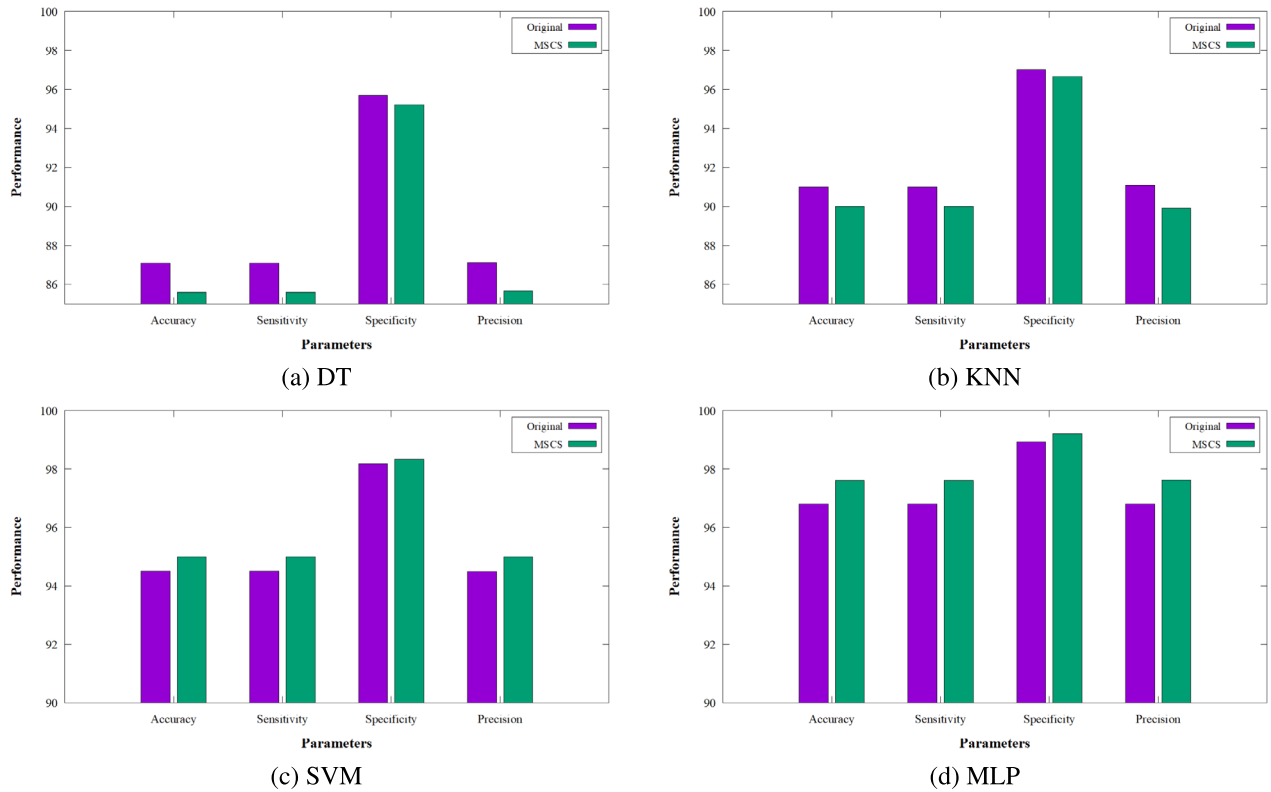
The above investigation is conducted using a dataset that is of high quality and readily accessible. Due to the current unavailability of a substantial dataset for the experiment, the study is grounded in conventional machine-learning techniques. A suitable deep-learning model that functions successfully with a minimal number of datasets may be considered for this problem. The method achieves a high degree of precision with the standard model; it will likely achieve even greater precision with deep learning models. When the strategy is combined with deep learning using deep features, it may be possible to identify more severity grades (more than four).

### Comparative analysis of optimized SLIC and other segmentation methods

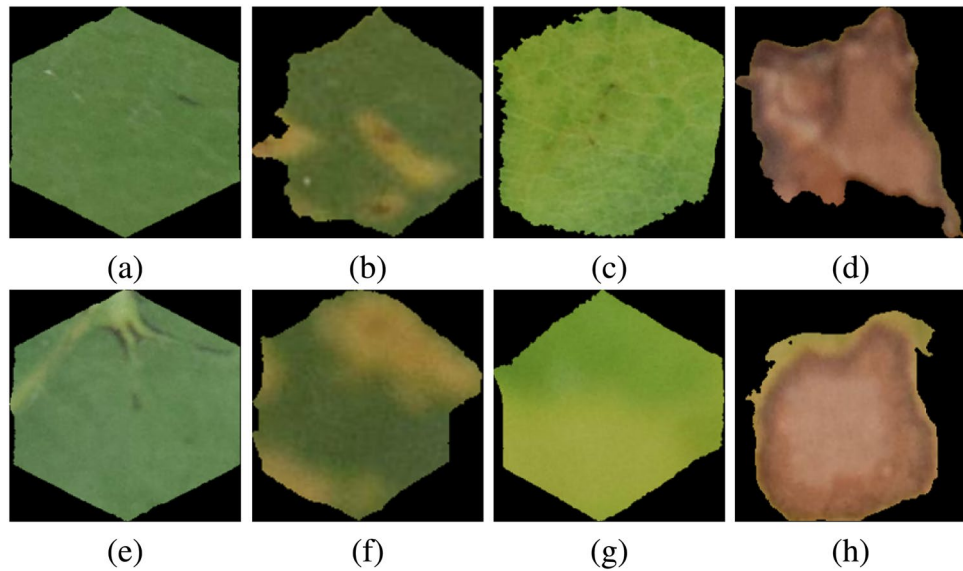
In this section the proposed approach is analyzed with other methods.

#### Comparison with global thresholding based methods

The efficacies of the two methodologies proposed by Golzari Oskouei et al. (2021)<sup>62</sup> and Song et al. (2024)<sup>63</sup> are given herein. Golzari Oskouei et al. (2021) presented the Cluster-weight and Group-local Feature-weight

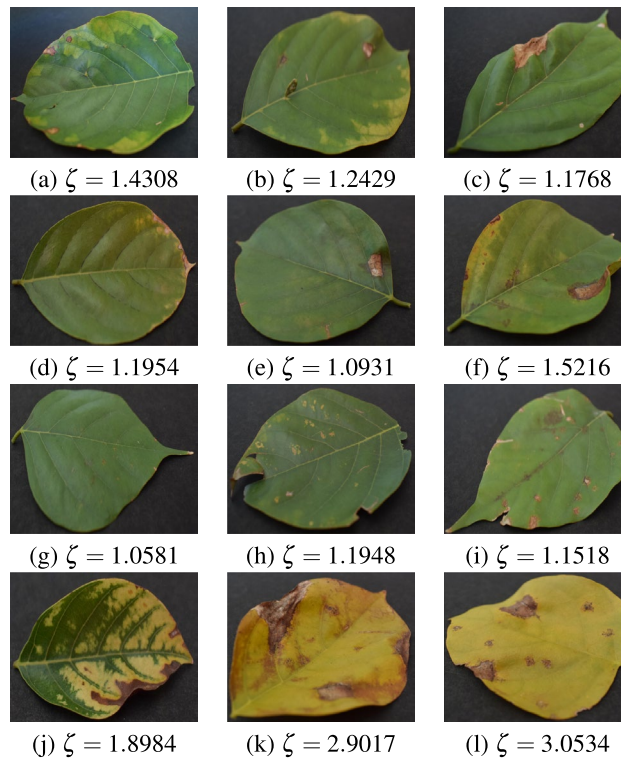


**Fig. 9.** Performance comparison of different classifiers on original and MSCS-selected feature sets.



**Fig. 10.** Superpixel category: **a, e** Type-1, **b, f** Type-2, **c, g** Type-3, **d, h**: Type-4.

learning in the Fuzzy C-Means (CGFFCM) method for color image segmentation. They developed an enhanced fuzzy c-means (FCM) clustering technique for color image segmentation. This study demonstrates that CGFFCM applies automated cluster weighting and group-local feature weighting to address FCM's sensitivity to initial cluster selection and uniform feature weighting. Song et al. (2024) introduced a Modified Snake Optimizer (MSO) to enhance color image segmentation for crop disease leaf, utilizing Kapur's entropy as the objective function. This study presents dynamic parameter modification, enhanced location updates, Levy flight for evading local optima, and a balancing method to increase global and local search efficacy. Figure 13 displays the sample images with their respective ground truth images. The segmentation outcomes of the two approaches



**Fig. 11.** Severity scores of some sample leaves.

Sl No	Sample No	No of superpixels				Weight factor $\omega$				Severity Score $\zeta$
		Type-1	Type-2	Type-3	Type-4	$\omega_1$	$\omega_2$	$\omega_3$	$\omega_4$	
1	Fig. 11a	402	34	80	11	1	2	3	4	1.4308
2	Fig. 11b	230	35	12	3	1	2	3	4	1.2429
3	Fig. 11	213	7	2	10	1	2	3	4	1.1768
4	Fig. 11d	225	20	3	8	1	2	3	4	1.1954
5	Fig. 11e	248	2	2	6	1	2	3	4	1.0931
6	Fig. 11f	167	20	34	11	1	2	3	4	1.5216
7	Fig. 11g	217	3	2	2	1	2	3	4	1.0581
8	Fig. 11h	223	37	6	1	1	2	3	4	1.1948
9	Fig. 11i	205	11	1	7	1	2	3	4	1.1518
10	Fig. 11j	116	31	86	3	1	2	3	4	1.8984
11	Fig. 11k	18	32	206	39	1	2	3	4	2.9017
12	Fig. 11l	2	2	256	21	1	2	3	4	3.0534

**Table 8.** Details of severity score  $\zeta$  calculation of Fig. 11 samples.

are illustrated in Fig. 13. The visual analysis indicates that Golzari Oskouei et al. (2021)<sup>62</sup> outperform Song et al. (2024)<sup>63</sup>. Furthermore, both approaches can segment the lesion region to a considerable extent. This signifies the adequate performance of both methods in segmenting diseased plant images. Further study is conducted based on the various segmentation quality parameters in Table 9. The parameter values also show good segmentation quality, notably for the approaches proposed by Golzari Oskouei et al. (2021)<sup>62</sup>.

Based on the preceding segmentation results in Fig. 13, a severity score can be calculated by determining the Euclidean distances between clusters representing diseased areas and healthy leaves. The distinct clusters in the segmentation output signify the varying levels of severity. Thus, the severity score ( $\zeta$ ) is computed according to Eqs. 32–34. In this context,  $|S_i|$  denotes the number of pixels in cluster  $S_i$ ,  $u$  signifies the total number of clusters, and  $|L_{area}|$  indicates the total leaf area (in pixels). The set  $H$  represents a set of five healthy leaf samples (shown in Fig. 14). The distance between  $S_i$  and  $H$  is denoted by the Euclidean distance ( $\delta_i$ ). Here, we have considered the mean of five distances (concerning five healthy samples) as  $\delta_i$ . Finally,  $\zeta$  is calculated using Eq. 34, which incorporates the individual leaf severity  $d_n$  and the maximum leaf severity  $d_{max}$ .

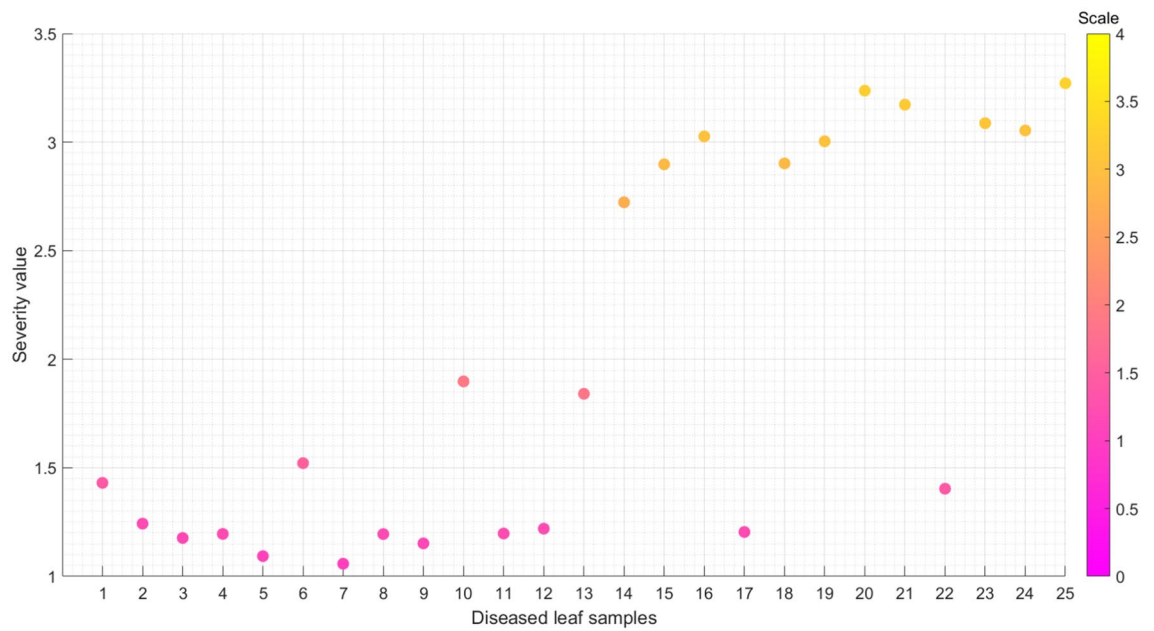


Fig. 12. Severity scores of twenty-five sample leaf images.

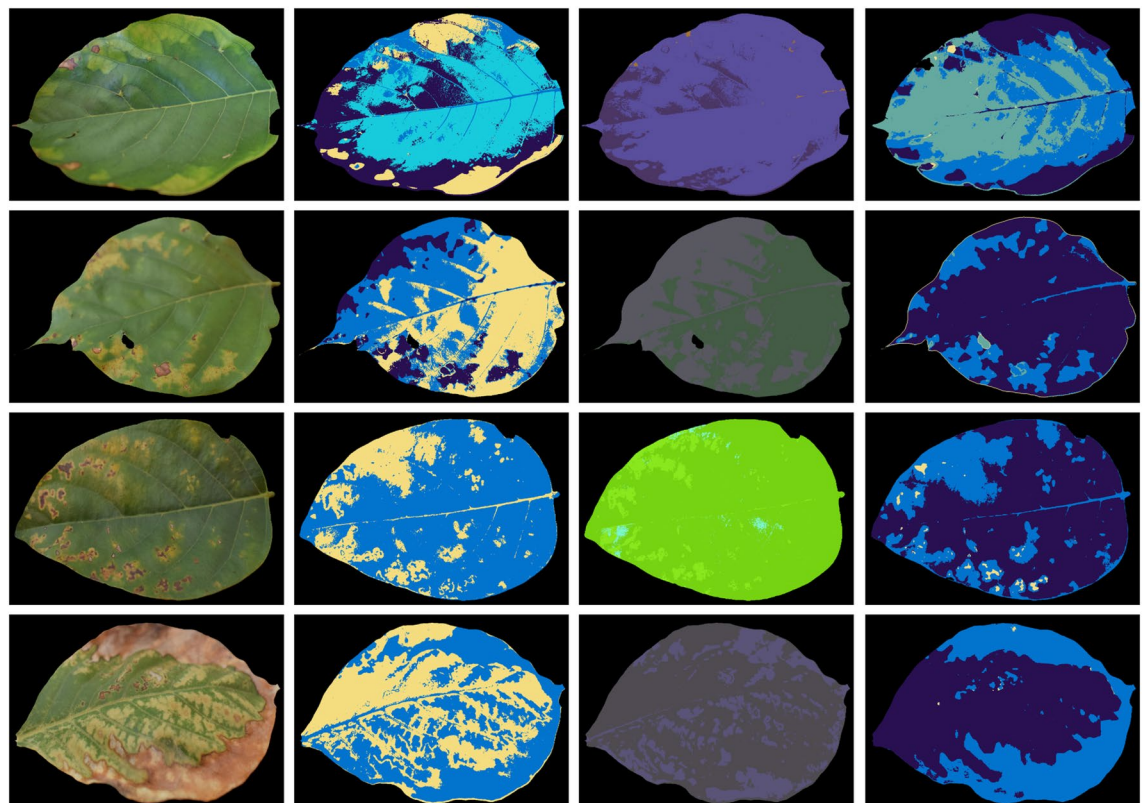
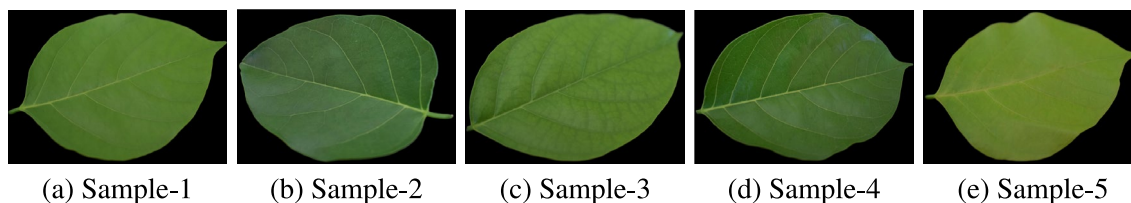


Fig. 13. First column: four sample images; second column: segmentation ground truth; third column: segmentation using Song et al. (2024); fourth column: segmentation using Golzari Oskouei et al. (2021).

$$d_n = \sum_{i=1}^u \delta_i \times \frac{|S_i|}{|L_{area}|} \tag{32}$$

Method	Sample	Accuracy	Precision	Recall	F1 Score	Jaccard	Dice
Song et al.(2024)	1	0.8023	0.2195	0.3935	0.2353	0.2193	0.2353
	2	0.8138	0.2035	0.2500	0.2243	0.2035	0.2243
	3	0.9657	0.6856	0.5886	0.6102	0.5491	0.6102
	4	0.8423	0.4526	0.5033	0.4735	0.4221	0.4735
Golzari Oskouei et al.(2021)	1	0.8470	0.3890	0.3418	0.3432	0.2943	0.3432
	2	0.9091	0.5120	0.5770	0.4937	0.4342	0.4937
	3	0.9725	0.6655	0.6914	0.6744	0.6215	0.6744
	4	0.8600	0.4178	0.4215	0.4182	0.3527	0.4182

**Table 9.** Performance comparison of MSO<sup>63</sup> and CGFFCM<sup>62</sup> on segmentation quality metrics.



**Fig. 14.** Samples of healthy leaf images.

Sample	Euclidean Distance				No of Pixels				Total Pixels	Severity
	Cluster-1	Cluster-2	Cluster-3	Cluster-4	Cluster-1	Cluster-2	Cluster-3	Cluster-4		
1	27.9807	61.9136	36.1406	46.9871	172777	87883	186142	1325	448127	31.99
2	79.7835	31.4777	46.8086	51.4238	77105	308874	3463	4468	393910	34.69
3	48.201	13.2757	38.947	0	132615	319366	5454	0	457435	19.91
4	100.4091	69.5458	36.2661	0	172460	276033	472	0	448965	68.35

**Table 10.** Euclidean distance based severity estimation from the segments.

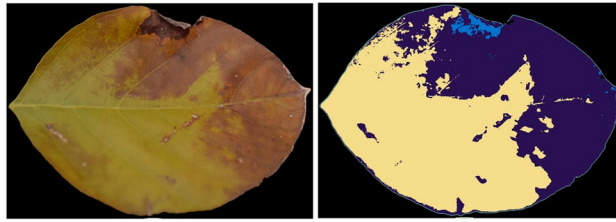
$$\delta_i = \text{distance}(H, S_i) \quad (33)$$

$$\zeta = \frac{d_n}{d_{max}} \times 100\% \quad (34)$$

While this method may be straightforward, it has two limitations: firstly, determining the optimal number of color threshold levels, which can vary according to the severity stages as the color changes, and secondly, the challenge of precise detection, where minor color variations may not accurately reflect changes in severity stages. The detection of tiny color patches is crucial for early disease identification. Segmentation quality for the four samples shown in Fig. 13 are presented in Table 9 and the estimated severities (of the four leaf samples) using these segmentation approaches are shown in Table 10. These leaf image samples show the challenges related to optimal segmentation. The corresponding Jaccard indices show this. Another sample is shown in Fig. 15, which shows that some segmentation strategies may not detect the disease area properly (light-shaded leaf areas are not detected as a separate segment in this example). In this context, we believed that capturing lesion regions with appropriate training-based classification using superpixels obtained from the lesions may enhance understanding of disease severity.

#### Histogram matching based comparison

A further study has been conducted that focuses on histogram matching-based severity estimation. Here, the histograms are prepared from individual color channels of the healthy reference image and diseased leaves. The normalized histograms of healthy image and diseased image are presented  $H_h^c(i)$  and  $H_d^c(i)$  at intensity level  $i \in \{0, \dots, 255\}$ , for color channel  $c \in \{R, G, B\}$ . In this context, three parameters, *Chi-square distance*, *histogram intersection*, and *correlation coefficient*, have been considered for histogram matching. The Chi-Square Distance quantifies the difference between the two histograms; lower values indicate more similarity; histogram intersection measures the overlap between the source and reference histograms; higher values indicate greater similarity in colour distribution and correlation coefficient: evaluates the linear relationship between the histograms; values closer to 1 indicate a substantial similarity. The calculation of the Chi-Square distance, histogram intersection, and correlation coefficient are provided in the Eqs. 35–40 respectively. The  $cov(H_d^c, H_h^c)$



**Fig. 15.** Sample leaf image (with clear colored segments) vs Corresponding segmentation output using Golzari Oskouei et al. (2021).

denotes covariance between the histograms of the diseased and healthy images for channel  $c$ . The  $\sigma_{H_d^c}$  indicates the standard deviation of the histograms of the diseased images for channel  $c$ .

$$\chi_c^2 = \sum_{i=0}^{255} \frac{(H_d^c(i) - H_h^c(i))^2}{H_h^c(i) + \epsilon} \quad (35)$$

$$\overline{\chi^2} = \frac{1}{3} \sum_{c \in \{R, G, B\}} \chi_c^2 \quad (36)$$

$$HI^c = \sum_{i=0}^{255} \min(H_d^c(i), H_h^c(i)) \quad (37)$$

$$\overline{HI} = \frac{1}{3} \sum_{c \in \{R, G, B\}} HI^c \quad (38)$$

$$\rho_c = \frac{\text{cov}(H_d^c, H_h^c)}{\sigma_{H_d^c} \cdot \sigma_{H_h^c}} \quad (39)$$

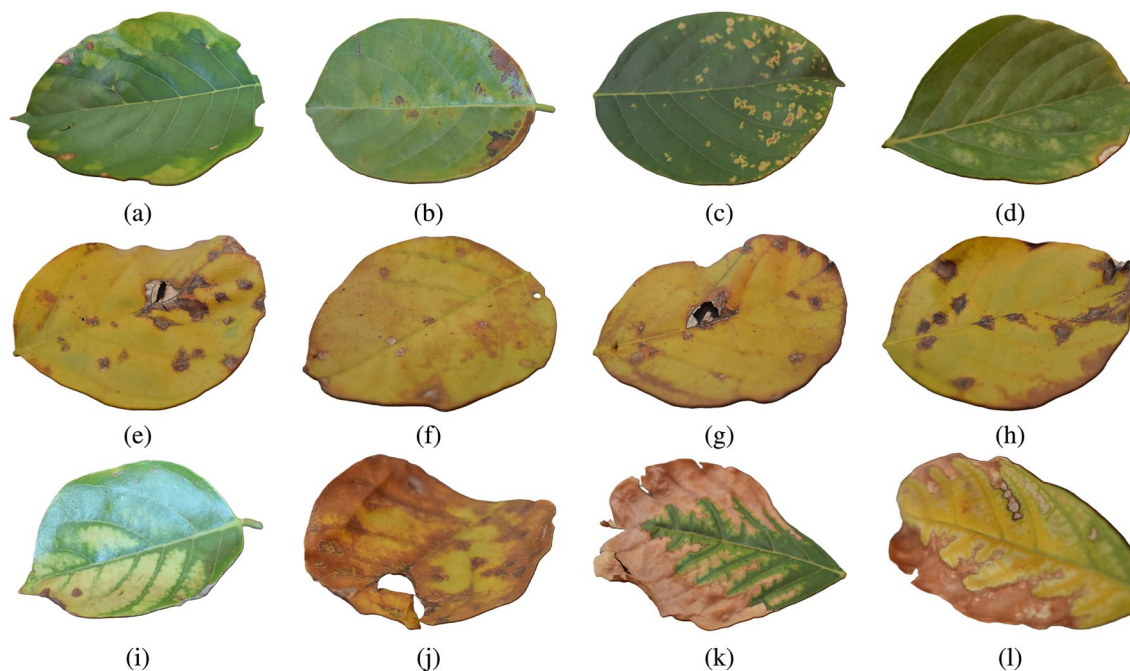
$$\overline{\rho} = \frac{1}{3} \sum_{c \in \{R, G, B\}} \rho_c \quad (40)$$

After determining the distance of each channel, the k-means clustering technique is used to categorize them into visually identifiable three groups according to the disease impact. Four samples from each group are shown in Fig. 16. Visually, it is evident that the images in Fig. 16a–d depict leaves with minimal damage (Grade-1), while those in (e)–(h) exhibit more significant color distortion (Grade-2). The images in (i)–(l) show even more pronounced color changes (Grade-3). Table 11 presents the mean values of the parameters, Chi-square distance, histogram intersection, and correlation coefficient for individual channels for the images shown in Fig. 16. It is worth mentioning that when we use color-profile-distance-based grouping or categorization of the leaves, the image shown in Fig. 16i is grouped into Grade-3. However, this does not match with human perception. Our proposed superpixel-based estimation of severity for the leaf is 1.77 on a scale of 1 (lowest) to 4 (highest), which relates to the human perception with more perfection (see Fig. 17).

#### Comparative study with varying number of superpixels

Figure 18 displays the original sample leaf photos with the results obtained with different numbers of superpixels. The number of superpixels varies with the values of 30, 60, 90, 120, and 150. The outcomes of the superpixels obtained using Cuckoo search optimization are also presented. The results indicate that increased superpixels lead to more precise lesion area detection. The optimal superpixels may identify the lesion more precisely than the superpixel range of 30 to 150. In sample-1 of Fig. 18, when the number of superpixels is 30, there exists a significant number of superpixels showing color variation over the optimal case.

Table 12 provides a comparative analysis of SLIC superpixel segmentation across four example images at varying superpixel counts, utilizing three parameters: average compactness, average superpixel size, and average color variation. An optimal number of superpixels is determined for each sample image, signifying the best trade-off balance among these criteria. Compactness often decreases with increased superpixels, signifying more uniformly formed superpixels. Reduced compactness values in the optimum rows indicate that precise boundary adherence, essential for identifying irregular lesion forms, is most effectively attained with relatively high superpixel counts. The average size of the superpixels decreases as the number of superpixels increases. The optimum results show reduced superpixel sizes, which may enable more accurate differentiation between diseased and healthy regions. The variation in color decreases as the number of superpixels increases, indicating that higher segmentation levels provide more homogeneous regions. The optimum result shows minimal color variation, which is crucial for identifying areas impacted by minor color differences. This level of detail is crucial for detecting the early stage of leaf disease. The optimal values presented in the table provide a balanced



**Fig. 16.** Three grade of severity based on histogram matching: Grade-1: (a)–(d), Grade-2: (e)–(h), Grade-3: (i)–(l).

Severity	Sample	Chi-Square				Histogram Intersection				Correlation Coefficient			
		Red	Green	Blue	Mean	Red	Green	Blue	Mean	Red	Green	Blue	Mean
Grade-1	Fig. 16a	8.1838	10.8936	7.6685	<b>8.9153</b>	0.5251	0.5380	0.6783	<b>0.5805</b>	0.9644	0.9628	0.9815	<b>0.9696</b>
	Fig. 16b	26.492	20.2919	45.8929	<b>30.8922</b>	0.4121	0.4521	0.5512	<b>0.4718</b>	0.9482	0.9489	0.9717	<b>0.9562</b>
	Fig. 16c	9.4183	0.8542	0.9537	<b>3.7421</b>	0.6406	0.7098	0.6902	<b>0.6802</b>	0.9732	0.9728	0.9780	<b>0.9747</b>
	Fig. 16d	7.0883	0.7739	0.9655	<b>2.9426</b>	0.5285	0.8435	0.7611	<b>0.7110</b>	0.9625	0.9951	0.9877	<b>0.9818</b>
Grade-2	Fig. 16e	1895.455	31.6813	3.7159	<b>643.6175</b>	0.3082	0.4195	0.6733	<b>0.4670</b>	0.9003	0.9080	0.9675	<b>0.9253</b>
	Fig. 16f	8182.588	17.2656	1.1276	<b>2733.661</b>	0.3078	0.4396	0.6467	<b>0.4647</b>	0.8973	0.9309	0.9561	<b>0.9281</b>
	Fig. 16g	1184.225	17.2313	1.4309	<b>400.9624</b>	0.3532	0.5027	0.7206	<b>0.5255</b>	0.9231	0.9512	0.9767	<b>0.9503</b>
	Fig. 16h	7308.965	29.1748	1.2446	<b>2446.462</b>	0.3349	0.4795	0.6491	<b>0.4879</b>	0.9110	0.9245	0.9594	<b>0.9316</b>
Grade-3	Fig. 16i	191.9446	1574.727	780.0511	<b>848.9077</b>	0.4071	0.4091	0.4348	<b>0.4170</b>	0.9680	0.9653	0.9773	<b>0.9702</b>
	Fig. 16j	1439.347	6.913	0.7257	<b>482.3284</b>	0.3871	0.8042	0.7626	<b>0.6513</b>	0.9543	0.9937	0.9863	<b>0.9781</b>
	Fig. 16k	177.4627	4.4802	29.5628	<b>70.5019</b>	0.4872	0.6512	0.6839	<b>0.6074</b>	0.9728	0.9814	0.9876	<b>0.9806</b>
	Fig. 16l	1761.208	10.6888	2.2266	<b>591.3745</b>	0.3348	0.5300	0.7262	<b>0.5304</b>	0.9387	0.9627	0.9881	<b>0.9632</b>

**Table 11.** Comparison of metrics across severity grades.

segmentation outcome that provides a precise evaluation of disease severity. This system provides a more accurate, automated assessment of crop disease severity by ensuring precise localization and segmentation.

We observed that as the number of superpixels ( $k$ ) increases (from 30 to 150), the average superpixel size decreases, color variation is often reduced, and compactness improves, as shown in Table 12. When  $k$  is fixed at 300, and the optimization is limited to two parameters ( $m$  and  $r$ ), further improvements are observed in color variation for most instances. We also observe improvements in compactness for many instances. Hence, determining an appropriate value of  $k$  for an arbitrary leaf image is challenging; however, the results suggest that increasing  $k$  may enhance the metrics. However, if we designate  $k$  as too large, it results in over-segmentation. Here, minimal color variations indicate enhanced consistency within each superpixel, which is essential for precisely capturing local image characteristics. Furthermore, decreased superpixel size allows for improved granularity, although decreased compactness values signify more uniform and coherent superpixel configurations, which promote improved alignment with lesion borders and assist in accurate segmentation. As shown in Table 12, optimal performance across all three parameters ( $k$ ,  $m$ , and  $r$ ) consistently yields better results in terms of color variation and compactness. Consequently, the individual superpixels more closely conform to the actual lesion structure, enabling a more accurate and robust representation of disease-affected areas that leads to more precise severity estimation.



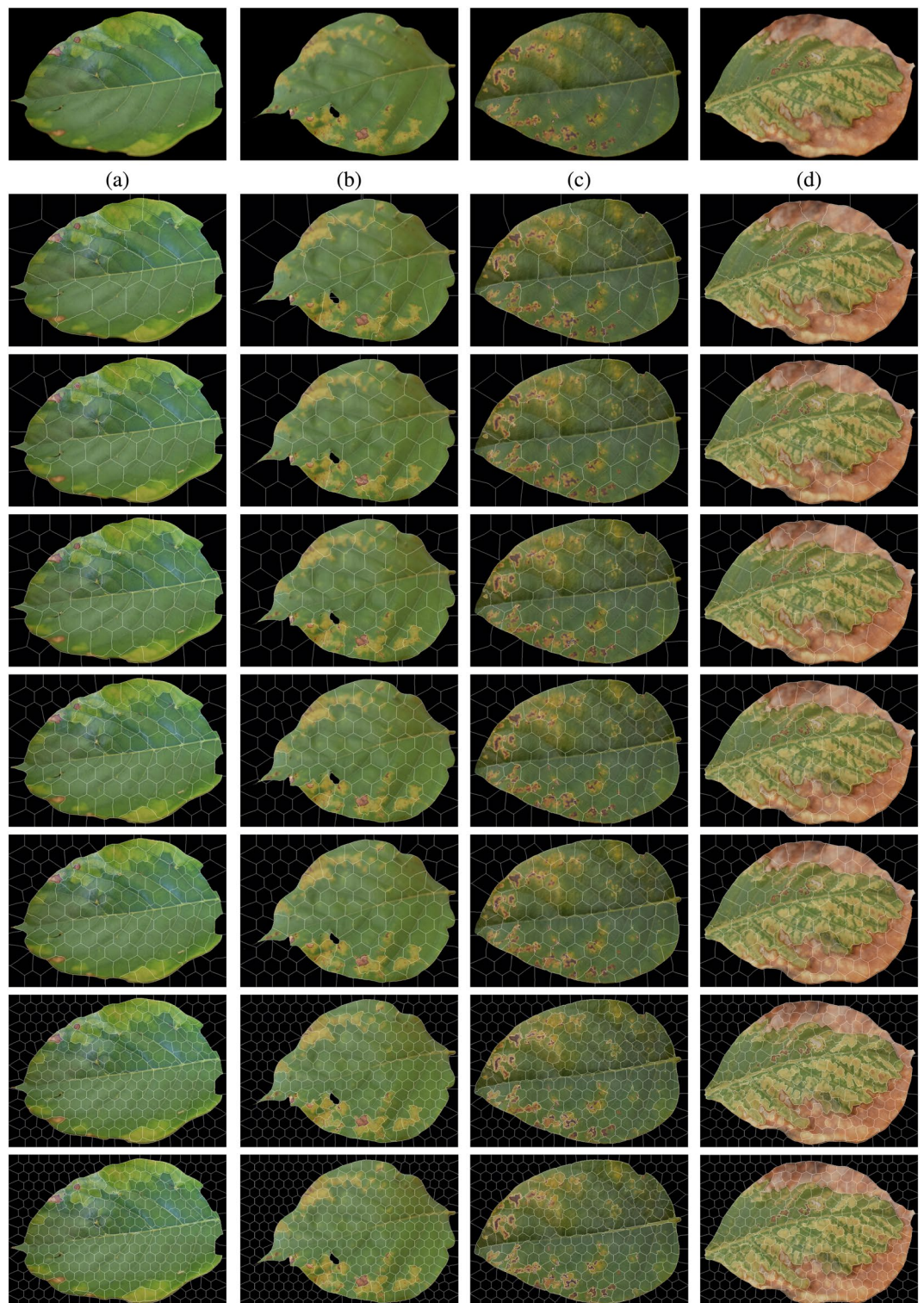
**Fig. 17.** Leaf image shown in Fig. 16i with severity 1.77.

The execution time for the experiments was measured with the image size of  $1000 \times 700$  pixels, and after experimenting with 10 sample images, the average execution time has been reported. The method based on Golzari Oskouei et al. (2021) took an average of 2.5587 minutes to run, while the method based on Song et al. (2024) took about 11.1637 minutes. For the SLIC algorithm, the average execution times with different numbers of superpixels were: 0.0867 mins (30 superpixels), 0.0917 mins (60), 0.0908 mins (90), 0.1003 mins (120), and 0.1051 mins (150). For the proposed method, the optimization step took an average of 21.5637 minutes, while the final execution time using the optimal SLIC parameters was 0.1565 minutes. It is important to note that the time for the feature selection step is not included here, as it is performed only once for the entire dataset.

### Conclusion & future work

In summary, the main objective of this research is to develop a chlorosis detection framework with high precision. PQCSAF measures the degree of chlorosis of the disease leaf. The color co-occurrence matrix-based features were utilized with the newly proposed *multi swarm cuckoo search* based feature selection strategy. It utilizes DT, KNN, SVM, and MLP-based supervised models for classification. Various performance measures validate all the experimental results. It is seen that MLP has obtained the maximum accuracy of 97.60%. The experiment demonstrates that utilizing superpixel-based lesion localization on the leaf results in the development of disease severity scores with a high level of accuracy. Furthermore, using a multi-swarm-based evolutionary feature selection technique enhances the system's ability to adapt to textural changes on the leaf caused by different diseases. The proposed approach has been implemented, and the chlorosis that impacts the leaves of *Pongamia pinnata* has been assessed. This dataset has been selected due to the extensive variation of color caused by the disease. In addition, datasets with other plant leaves and disease conditions may be considered for further analysis to adjudge the method's applicability in general. The model is robust, and it may be applied to monitoring crops as well as forests.

The research may be extended to include more types of leaf diseases by using multiple objective functions that account for disease-specific information. The model's performance may be further analyzed through hyperparameter adjustment and other classifiers. The optimization pipeline is subject to additional enhancement. Further investigations could look into the potential benefits of implementing the proposed framework directly on AI-enabled edge devices. This can broaden the research scope to develop an effective device for the Fog computing layer in on-field agricultural applications with reduced reliance on the Internet.



**Fig. 18.** Output for varying the number of superpixels: row-1: original Sample 1 – 4, row-2: 30, row-3: 60, row-4: 90, row-5: 120, row-6: 150, row-7: 300, row-8: optimum result (357, 416, 351 and 380).

Sample	No of Superpixels	Avg. Compactness	Avg. Superpixel Size	Avg. Color Variation
Fig. 18a	30	28.02370026	20000.0000	0.004352914
	60	27.79024058	10937.5000	0.003778743
	90	25.83830942	8433.73494	0.004010828
	120	24.72599071	5645.16129	0.002074245
	150	24.15277288	4827.58621	0.002528411
	300	22.40486912	2272.727273	0.001615095
	<b>Optimum</b>	<b>23.69421202</b>	<b>1902.17391</b>	<b>0.001280180</b>
Fig. 18b	30	27.91431518	19444.4444	0.003428631
	60	26.42788108	12068.9655	0.003751028
	90	25.51787268	8536.58537	0.003328570
	120	24.02955945	5737.70492	0.001740597
	150	23.92827400	4827.58621	0.001712131
	300	24.05319707	2127.659574	0.000880442
	<b>Optimum</b>	<b>22.54239220</b>	<b>1763.22418</b>	<b>0.000933445</b>
Fig. 18c	30	27.38739749	16279.0698	0.004025582
	60	25.51333984	11475.4098	0.003270565
	90	24.97259465	8139.53488	0.003058355
	120	24.20469426	5785.12397	0.002449244
	150	23.95379726	4666.66667	0.001960294
	300	28.82397349	2017.291066	0.001107312
	<b>Optimum</b>	<b>27.45519085</b>	<b>1846.96570</b>	<b>0.001087340</b>
Fig. 18d	30	30.59146352	20000.0000	0.007559161
	60	28.56969997	11290.3226	0.007259274
	90	25.94246069	8536.58537	0.006045887
	120	24.92690211	5737.70492	0.004857056
	150	24.56879157	4827.58621	0.004365713
	300	29.13128538	2095.808383	0.002057259
	<b>Optimum</b>	<b>23.55861262</b>	<b>1933.70166</b>	<b>0.002486619</b>

**Table 12.** Comparison of SLIC superpixel results across different samples and superpixel counts.

## Data availability

The dataset used for experimentation is publicly available (Shown in Ref. 58). The results or observations may be made available on reasonable request to Mr. Sourav Samanta (sourav.uit@gmail.com).

Received: 6 August 2024; Accepted: 19 September 2025

Published online: 24 October 2025

## References

- Chouhan, S. S., Singh, U. P. & Jain, S. Applications of computer vision in plant pathology: A survey. *Arch. Comput. Methods Eng.* **27**, 611–632. <https://doi.org/10.1007/s11831-019-09324-0> (2019).
- Vishnoi, V. K., Kumar, K. & Kumar, B. Plant disease detection using computational intelligence and image processing. *J. Plant Dis. Prot.* **128**, 19–53. <https://doi.org/10.1007/s41348-020-00368-0> (2020).
- Borkar, S. G. *History of Plant Pathology* (WPI Publishing, NY, 2017).
- Raychaudhuri, S. P., Verma, J. P., Nariani, T. K. & Sen, B. The history of plant pathology in India. *Ann. Rev. Phytopathol.* **10**, 21–36. <https://doi.org/10.1146/annurev.py.10.090172.000321> (1972).
- Strange, R. N. & Scott, P. R. Plant disease: A threat to global food security. *Ann. Rev. Phytopathol.* **43**, 83–116. <https://doi.org/10.1146/annurev.phyto.43.113004.133839> (2005).
- Savary, S. et al. The global burden of pathogens and pests on major food crops. *Nat. Ecol. Evol.* **3**, 430–439. <https://doi.org/10.1038/s41559-018-0793-y> (2019).
- Agrios, G. N. *Plant Pathology* 5th edn. (Academic Press, 2005).
- Jones, J. B., Momol, M. T., Olson, S. M. & Funderburk, J. E. *Diseases of Tomato and Pepper* (Wiley, NY, 2014).
- Zell, R. Global climate change and the emergence/re-emergence of infectious diseases. *Int. J. Med. Microbiol. Suppl.* **293**, 16–26. [https://doi.org/10.1016/s1433-1128\(04\)80005-6](https://doi.org/10.1016/s1433-1128(04)80005-6) (2004).
- Mansfield, J. et al. Top 10 plant pathogenic bacteria in molecular plant pathology. *Mol. Plant Pathol.* **13**, 614–629. <https://doi.org/10.1111/j.1364-3703.2012.00804.x> (2012).
- Gilligan, C. A. Sustainable agriculture and plant diseases: an epidemiological perspective. *Philos. Trans. R. Soc. B: Biol. Sci.* **363**, 741–759. <https://doi.org/10.1098/rstb.2007.2181> (2007).
- Getahun, S., Kefale, H. & Gelaye, Y. Application of precision agriculture technologies for sustainable crop production and environmental sustainability: A systematic review. *Sci. World J.* <https://doi.org/10.1155/2024/2126734> (2024).
- Haggag, W. M. Towards sustainable agriculture: role of modern agricultural technologies in plant disease management. In *Sustainable Agriculture Reviews* Vol. 35 (ed. Lichtfouse, E.) 135–170 (Springer International Publishing, 2021).
- University of Illinois Extension – focus on plant problems. Accessed 08 Feb 2022. Chlorosis | Focus on Plant Problems | U of I Extension (2022).

15. Purdue University Extension – iron chlorosis of trees and shrubs. Accessed 08 Feb 2022.
16. Chlorosis - archive, Accessed 08 Feb 2022.
17. Pankaja, K. & Suma, V. Leaf recognition and classification using Chebyshev moments. In *Smart Intelligent Computing and Applications* 667–678 (Springer Singapore, NY, 2018). [https://doi.org/10.1007/978-981-13-1927-3\\_70](https://doi.org/10.1007/978-981-13-1927-3_70).
18. Islam, S. T. & Mazumder, B. Wavelet based feature extraction for rice plant disease detection and classification. In *2019 3rd International Conference on Electrical, Computer & Telecommunication Engineering (ICECTE)*, 53–56. <https://doi.org/10.1109/icect48615.2019.9303567> (IEEE, 2019).
19. Tampinong, F. F., Herdiyeni, Y. & Herliyana, E. N. Feature extraction of Jabon (anthocephalussp) leaf disease using discrete wavelet transform. *Telecommun. Comput. Electron. Control*, **18**, 740. <https://doi.org/10.12928/telkommika.v18i2.10714> (2020).
20. Dhingra, G., Kumar, V. & Joshi, H. D. Quality assessment of leaves quality using texture and DWT based local feature extraction analysis. *Chemom. Intell. Lab. Syst.* **208**, 104195. <https://doi.org/10.1016/j.chemolab.2020.104195> (2020).
21. Qin, F., Guo, J. & Lang, F. Superpixel segmentation for polarimetric SAR imagery using local iterative clustering. *IEEE Geosci. Remote Sens. Lett.* **12**, 13–17. <https://doi.org/10.1109/lgrs.2014.2322960> (2015).
22. Zhang, S., You, Z. & Wu, X. Plant disease leaf image segmentation based on superpixel clustering and EM algorithm. *Neural Comput. Appl.* **31**, 1225–1232. <https://doi.org/10.1007/s00521-017-3067-8> (2017).
23. Zhang, S., Zhu, Y., You, Z. & Wu, X. Fusion of superpixel, expectation maximization and PHOG for recognizing cucumber diseases. *Comput. Electron. Agric.* **140**, 338–347. <https://doi.org/10.1016/j.compag.2017.06.016> (2017).
24. Khan, S. & Narvekar, M. Novel fusion of color balancing and superpixel based approach for detection of tomato plant diseases in natural complex environment. *J. King Saud Univ. Comput. Inf. Sci.* <https://doi.org/10.1016/j.jksuci.2020.09.006> (2020).
25. Singh, V. Sunflower leaf diseases detection using image segmentation based on particle swarm optimization. *Artif. Intell. Agric.* **3**, 62–68. <https://doi.org/10.1016/j.iaia.2019.09.002> (2019).
26. Sengar, N., Dutta, M. K. & Travieso, C. M. Computer vision based technique for identification and quantification of powdery mildew disease in cherry leaves. *Computing* **100**, 1189–1201. <https://doi.org/10.5555/3288338.3288342> (2018).
27. Pandey, C., Baghel, N., Dutta, M. K., Srivastava, A. & Choudhary, N. Machine learning approach for automatic diagnosis of chlorosis in Vigna mungo leaves. *Multimed. Tools Appl.* **80**, 13407–13427. <https://doi.org/10.1007/s11042-020-10309-6> (2021).
28. Chakraborty, K. K., Mukherjee, R., Chakraborty, C. & Bora, K. Automated recognition of optical image based potato leaf blight diseases using deep learning. *Physiol. Mol. Plant Pathol.* **117**, 101781. <https://doi.org/10.1016/j.pmpp.2021.101781> (2022).
29. Phan, H., Ahmad, A. & Saraswat, D. Identification of foliar disease regions on corn leaves using slic segmentation and deep learning under uniform background and field conditions. *IEEE Access* **10**, 111985–111995. <https://doi.org/10.1109/access.2022.3215497> (2022).
30. Abisha, S., Mutawa, A. M., Murugappan, M. & Krishnan, S. Brinjal leaf diseases detection based on discrete shearlet transform and deep convolutional neural network. *PLOS ONE* **18**, e0284021. <https://doi.org/10.1371/journal.pone.0284021> (2023).
31. Singla, P., Kalavakonda, V. & Senthil, R. Detection of plant leaf diseases using deep convolutional neural network models. *Multimed. Tools Appl.* <https://doi.org/10.1007/s11042-023-18099-3> (2024).
32. Bedi, P., Gole, P. & Marwaha, S. Pdse-lite: lightweight framework for plant disease severity estimation based on convolutional autoencoder and few-shot learning. *Front. Plant Sci.* **14**, 15. <https://doi.org/10.3389/fpls.2023.1319894> (2024).
33. Chouhan, S. S., Singh, U. P., Sharma, U. & Jain, S. Leaf disease segmentation and classification of *Jatropha curcas* L. and *Pongamia pinnata* L. biofuel plants using computer vision based approaches. *Measurement* **171**, 108796. <https://doi.org/10.1016/j.measurement.2020.108796> (2021).
34. Achanta, R. et al. *Slic superpixels* (Technical report, EPFL, 2010).
35. Ortiz, A., Górriz, J. M., Ramírez, J., Salas-González, D. & Llamas-Elvira, J. M. Two fully-unsupervised methods for MR brain segmentation using SOM-based strategies. *Appl. Soft Comput.* **13**, 2668–2682 (2013).
36. Ortiz, A., Palacio, A. A., Górriz, J. M., Ramírez, J. & Salas-González, D. Segmentation of brain MRI using SOM-FCM-based method and 3D statistical descriptors. *Comput. Math. Methods Med.* **2013**, 638563. <https://doi.org/10.1155/2013/638563> (2013).
37. Haralick, R. M., Shanmugam, K. & Dinstein, I. Textural features for image classification. *IEEE Trans. Syst. Man and Cybern.* **3**, 610–621 (1973).
38. Jolliffe, I. T. *Principal Component Analysis: Springer Series in Statistics* (Springer, New York, NY, 2002).
39. Tsai, F. S. Dimensionality reduction for computer facial animation. *Expert Syst. Appl.* **39**, 4965–4971. <https://doi.org/10.1016/j.eswa.2011.10.018> (2012).
40. Ilin, A. & Raiko, T. Practical approaches to principal component analysis in the presence of missing values. *J. Mach. Learn. Res.* **11**, 1957–2000. <https://doi.org/10.5555/1756006.1859917> (2010).
41. Abd-alsabour, N. A review on evolutionary feature selection. In *2014 European Modelling Symposium*. <https://doi.org/10.1109/em.s.2014.28> (IEEE, 2014).
42. Saraç, E. & Özel, S. A. An ant colony optimization based feature selection for web page classification. *Sci. World J.* **2014**, 1–16. <https://doi.org/10.1155/2014/649260> (2014).
43. Yadav, S., Ekbal, A. & Saha, S. Information theoretic-PSO-based feature selection: an application in biomedical entity extraction. *Knowl. Inf. Syst.* **60**, 1453–1478. <https://doi.org/10.1007/s10115-018-1265-z> (2018).
44. Marie-Sainte, S. L. & Alalyani, N. Firefly algorithm based feature selection for Arabic text classification. *J. King Saud Univ. Comput. Inf. Sci.* **32**, 320–328. <https://doi.org/10.1016/j.jksuci.2018.06.004> (2020).
45. Alzaqebah, M. et al. Memory based cuckoo search algorithm for feature selection of gene expression dataset. *Inf. Med. Unlock* **24**, 100572. <https://doi.org/10.1016/j.imu.2021.100572> (2021).
46. Khan, A. H., Sarkar, S. S., Mali, K. & Sarkar, R. A genetic algorithm based feature selection approach for microstructural image classification. *Exp. Tech.* **46**, 335–347. <https://doi.org/10.1007/s40799-021-00470-4> (2021).
47. Breiman, L., Friedman, J., Charles, S. J. & A., O. R. *Classification and regression trees* (Chapman and Hall, NY, 1984).
48. Priyanka, N. & Kumar, D. Decision tree classifier: a detailed survey. *Int. J. Inf. Decis. Sci.* **12**, 246. <https://doi.org/10.1504/ijids.2020.108141> (2020).
49. Cover, T. & Hart, P. Nearest neighbor pattern classification. *IEEE Trans. Inf. Theory* **13**, 21–27. <https://doi.org/10.1109/TIT.1967.1053964> (1967).
50. Mucherino, A., Papajorgji, P. J. & Pardalos, P. M. k-nearest neighbor classification. In *Data Mining in Agriculture Springer Optimization and Its Applications* 83–106 (Springer, New York, 2009). <https://doi.org/10.1007/978-0-387-88615-2>.
51. Vapnik, V. N. Pattern recognition using generalized portrait method. *Autom. Remote Control* **24**, 774–780 (1963).
52. Chakraborty, A. et al. Determining protein-protein interaction using support vector machine: A review. *IEEE Access* **9**, 12473–12490. <https://doi.org/10.1109/access.2021.3051006> (2021).
53. Al-Mejibli, I. S., Abd, D. H., Alwan, J. K. & Rabash, A. J. Performance evaluation of kernels in support vector machine. In *2018 1st Annual International Conference on Information and Sciences (AiCIS)*, 96–101. <https://doi.org/10.1109/AiCIS.2018.00029> (2018).
54. Hastie, T., Tibshirani, R. & Friedman, J. *The Elements of Statistical Learning Data Mining, Inference, and Prediction* 2nd edn. (Springer, New York, 2009).
55. Rosenblatt, F. *Principles of Neurodynamics: Perceptrons and the Theory Of Brain Mechanisms* (Spartan Books, 1962).
56. Cybenko, G. Approximation by superpositions of a sigmoidal function. *Math. Control, Signals Syst.* **2**, 303–314. <https://doi.org/10.1007/bf02551274> (1989).

57. Chouhan, S. S., Singh, U. P., Kaul, A. & Jain, S. A data repository of leaf images: Practice towards plant conservation with plant pathology. In *4th International Conference on Information Systems and Computer Networks (ISCON)*, 700–707. <https://doi.org/10.1109/ISCON47742.2019.9036158> (2019).
58. Chouhan, S. S., Singh, U. P., Kaul, A. & Jain, S. A database of leaf images: Practice towards plant conservation with plant pathology. <https://doi.org/10.17632/hb74ynkjc.1>. <https://data.mendeley.com/datasets/hb74ynkjc/1> (2020).
59. Otsu, N. A Threshold Selection Method from Gray-level Histograms. *IEEE Trans. Syst. Man Cybern.* **9**, 62–66. <https://doi.org/10.1109/TSMC.1979.4310076> (1979).
60. Yang, X.-S. & Deb, S. Cuckoo search via Lévy flights. In *2009 World Congress on Nature & Biologically Inspired Computing (NaBIC)*. <https://doi.org/10.1109/nabic.2009.5393690> (IEEE, 2009).
61. Guerrero-Luis, M., Valdez, F. & Castillo, O. A review on the cuckoo search algorithm. In *Fuzzy Logic Hybrid Extensions of Neural and Optimization Algorithms: Theory and Applications*, 113–124, [https://doi.org/10.1007/978-3-030-68776-2\\_7](https://doi.org/10.1007/978-3-030-68776-2_7) (Springer International Publishing, 2021).
62. Golzari Oskouei, A., Hashemzadeh, M., Asheghi, B. & Balafar, M. A. Cgffcm: Cluster-weight and group-local feature-weight learning in fuzzy c-means clustering algorithm for color image segmentation. *Appl. Soft Comput.* **113**, 108005. <https://doi.org/10.1016/j.asoc.2021.108005> (2021).
63. Song, H., Wang, J., Bei, J. & Wang, M. Modified snake optimizer based multi-level thresholding for color image segmentation of agricultural diseases. *Expert Syst. Appl.* **255**, 124624. <https://doi.org/10.1016/j.eswa.2024.124624> (2024).

### Author contributions

Sourav Samanta: Conceptualization, Data preparation, Formal analysis, Validation, Visualization, Writing – original draft. Sanjoy Pratihar: Conceptualization, Validation, Supervision, Writing – review & editing. Sanjay Chatterji: Supervision, Writing – review & editing.

### Declarations

### Competing interests

The authors declare no competing interests.

### Additional information

**Correspondence** and requests for materials should be addressed to S.S.

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

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

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

© The Author(s) 2025