



ARTICLE

Automatic Detection and Classification of Grape Leaf Diseases based on Deep Learning and Enhanced Chameleon Swarm Algorithm

Aya Sami,^{*} Reham R. Mostafa, and Sherif I. Barakat

Department of Information Systems, Faculty of Computers and Information, Mansoura University, Mansoura, 35516, Egypt

^{*}Corresponding author: ayasami@mans.edu.eg

(Received: 20 May 2024; Accepted: 18 July 2024; Published: 23 July 2024)

Abstract

Grape diseases and pest infestations threaten the economic viability of grape production, necessitating early detection and intervention. Leveraging advancements in machine learning and computer vision, researchers are developing automated systems that accurately identify and classify grape diseases, contributing to improved disease management strategies. This study proposes an automated framework for classifying and detecting grape leaf diseases, integrating an enhanced metaheuristic optimization algorithm with deep learning techniques. To address the class imbalance present in the Grape dataset, a Conditional Generative Adversarial Network (CGAN) is employed as a data augmentation technique, generating synthetic images to balance the representation of each class. Two pre-trained convolutional neural network (CNN) models, AlexNet and ResNet18, are then utilized to extract deep features from the augmented images. A fusion method aggregates the extracted feature vectors, which are subsequently optimized using an improved metaheuristic optimization algorithm for feature selection (FS). Metaheuristic algorithms, known for their dynamic search behavior and global search capabilities, offer promising solutions for FS. This study introduces the Enhanced Chameleon Swarm Optimizer (ECSA) method, a novel variant of the metaheuristic Chameleon Swarm Algorithm (CSA), to address the FS problem. The ECSA, with its use of chaotic maps during the exploration phase and integration of Levy flight distribution into the exploitation phase, represents a significant advancement in metaheuristic optimization. The final set of selected features is then classified using the K-Nearest Neighbors (KNN) algorithm for grape leaf disease identification. The performance of the proposed framework is assessed on a real-world dataset of grape diseases, employing multiple evaluation criteria. The proposed framework demonstrates superior performance, achieving a peak accuracy of 97.76% on the grape disease dataset.

Keywords: Chameleon Swarm algorithm (CSA); Metaheuristics; chaotic map; levy flight; Convolutional Neural Networks (CNNs); Feature selection (FS); Grape disease prediction.

1. Introduction

Grape diseases and pests pose a significant economic threat to grape production, potentially causing substantial financial losses for farmers if not detected and treated promptly [1]. While pesticide application is a common practice to mitigate these risks, frequent use can lead to detrimental environmental consequences, including developing pesticide resistance to pathogens and potential harm to human health through long-term exposure. Consequently, early disease detection emerges as a critical strategy for minimizing yield losses and ensuring sustainable grape production. For this reason, automated detection and recognition related to artificial intelligence have been thoroughly studied [2].

The rapid advancement of computer technology has enabled the integration of novel deep learning-based visual recognition techniques, including Convolutional Neural Networks (CNNs) and Mask Region-based Convolutional Neural Networks (Mask R-CNNs), into disease detection applications [3]. While traditional machine learning methods, such as Support Vector Machine (SVM) and K-Nearest Neighbor (KNN), have demonstrated utility in plant disease detection and treatment, their reliance on a pre-defined pipeline of image segmentation, feature extraction, and pattern recognition [4] limits their effectiveness in analyzing complex visual data.

Traditional approaches to feature selection and extraction for plant disease detection often require the involvement of highly qualified engineers and experienced specialists, rendering the process both time-consuming and resource-intensive. This manual process is subjective and necessitates significant financial and human resources. Deep learning methods, in contrast, offer a more automated and efficient solution. They eliminate the need for human intervention in feature extraction and classifier design by automatically learning hierarchical features of diseases from image data [5]. This capability has made deep learning techniques highly promising in agricultural applications, empowering small-scale farmers and gardeners with tools for addressing various challenges, including weed detection and plant disease diagnosis [6], [7]. The application of deep neural networks, particularly CNNs in plant disease identification, has seen significant growth since the emergence of deep learning in 2012.

CNNs [8] are artificial neural networks specifically developed for image identification and analysis. Mimicking the human brain's visual processing capabilities, CNNs utilize artificial neurons to observe, process, and predict input data, effectively identifying complex patterns within images. Unlike traditional neural network architectures, CNNs can learn directly from input images, extracting hierarchical features and applying a weight distribution strategy that ensures computational efficiency. To facilitate the classification of diseases across various crops, several CNN-based architectures have been developed, including AlexNet, GoogLeNet, and ResNet18.

Training CNNs on larger datasets generally leads to improved model performance. However, training on limited data can result in poor generalization performance during validation and testing due to overfitting. This occurs when the model learns to recognize specific patterns in the training data that are not generalizable to new data. Data augmentation [9] is a crucial technique for addressing this issue by synthetically generating additional training data from existing samples. It increases the diversity of the training set, exposing the model to a broader range of viewpoints, scales, rotations, and other transformations. As a result, it allows the model to learn robust features invariant to these variations, reducing the risk of memorizing specific training examples and improving its ability to generalize to unseen data. Therefore, data augmentation effectively bridges the gap between training, validation, and test samples by enhancing the model's ability to learn generalizable features of new data, leading to better overall performance.

Deep neural networks often perform better classification tasks than other traditional algorithms.

Nevertheless, training deep neural networks from scratch on a large dataset poses a significant time constraint, potentially requiring days or weeks. Transfer learning offers a solution that leverages pre-trained networks. Therefore, researchers prefer to start with a pre-trained model that can classify patterns and learn features such as color, texture, shapes, and edges in an image [10]. When applying transfer learning, the model trained on one task can be applied to another to improve the performance and progress of the second task. A significantly higher level of accuracy can be achieved when transfer learning is applied.

Pre-trained CNN models, including AlexNet, ResNet50, ResNet18, VGG-19, Inception V1, Inception V2, and VGG-16, were built from scratch using high-end GPUs and trained on over a million images that can be divided into a thousand distinct categories. These models acquired numerous low-level feature representations, including edges, shapes, colors, textures, and rotations, and disseminated the knowledge to other networks. Therefore, these models acted as feature extractors for new images in different areas. Based on the transfer-learning principle [11], a trained model can extract the relevant features from new images of distinct categories.

Optimizing the performance of learning algorithms can be challenging when the dataset contains many features because they may be irrelevant or redundant. Feature selection (FS) attempts to overcome this problem by eliminating irrelevant and redundant features, effectively reducing the complexity of the data without affecting the accuracy of the trained model's predictions [12]. In FS, finding the optimal set of features is crucial. However, due to the complexity of the time, traditional exhaustive techniques such as breadth-first search and depth-first search are considered impractical for identifying the best feature subset.

The computational effort required to create and evaluate the 2^N subset of features is significant. Therefore, FS is a challenging optimization problem with competing objectives [13], and metaheuristic algorithms offer promising solutions. Metaheuristics efficiently explore the search space to find near-optimal solutions. Recent research has demonstrated the efficiency of different metaheuristic algorithms in solving FS problems and their potential for optimizing feature subsets [13, 14, 15].

An effective metaheuristic algorithm is characterized by its ability to strike a delicate balance between exploration and exploitation, resulting in diverse solutions and rapid convergence. When developing a metaheuristic, the selection of search operators is based on the source of inspiration and the compatibility of the operators. Improving metaheuristics often involves enhancing existing algorithms by integrating different search methods, introducing additional computational steps, or even integrating with other metaheuristics [16]. These improvements go beyond the initial inspiration and aim to achieve superior optimization performance. Furthermore, specific optimization problems require the hybridization of algorithms to effectively address particular constraints and capture the unique characteristics of each issue. The motivation behind such hybridization comes from the No Free Lunch (NFL) theorem [14], which states that all metaheuristics perform comparably on all optimization problems, highlighting the need for dedicated algorithms to solve constrained problems efficiently. Consequently, the theorem catalyzes the development of novel metaheuristics tailored to various optimization problems and provides competitive results compared to previous approaches.

In 2021 [17], Braik developed the Chameleon Swarm Algorithm (CSA), one of the most modern and resourceful metaheuristic algorithms that fall into the category of swarm-based algorithms. CSA models the dynamic behavior of chameleons as they hunt and forage in their natural environment. As stated in [17], the effectiveness of CSA was confirmed by validating 67 publicly benchmark functions. The relevance of the CSA is evaluated using five constrained optimization problems and several designs for engineering difficulties. According to the authors, CSA demonstrated strong and increasing performance compared to traditional and current competitors. This achievement justified

the effectiveness of its global search capabilities. However, the exploration-exploitation trade-off in CSA is often imbalanced, leading to premature convergence to local optima and consequently affecting its overall optimization performance. This highlights the need for strategies that better balance exploration and exploitation to improve convergence.

1.1 Motivation

Agriculture is crucial in influencing nations' economies. Therefore, precise and prompt detection of grape leaf diseases is essential for plant health and productivity. Traditional disease classification methods often need help with high-dimensional datasets, especially those with noisy data, resulting in suboptimal solutions.

This research aims to propose an efficient framework for detecting diseases in grape leaves using pre-trained CNN models and an enhanced Chameleon swarm algorithm. This paper first uses a CGAN model as a data augmentation method on the selected Grape dataset to avoid the overfitting problem. Then, deep features are obtained from the images using two pre-trained CNN models, ResNet18 and AlexNet. A fusion technique is subsequently applied to merge and enhance these feature vectors. After that, an enhanced version of the CSA metaheuristic algorithm was proposed to optimize feature vectors, incorporating chaotic maps for global exploration and Levy flights for local optimization. Then, the selected features were classified using the KNN algorithm. The contributions of the work are summarized as follows:

- The CGAN is used in data augmentation to produce synthetic samples within a particular category.
- Deep features are obtained from the images using two pre-trained CNN models: ResNet18 and AlexNet.
- A fusion method is employed to combine and enhance the feature vectors.
- The ECSA is presented as a solution to the FS problem.
- The original CSA is enhanced by integrating levy flight and chaotic maps.
- The ECSA is evaluated for FS on a practical challenge related to the classification of Grape disease.
- Diverse analytical metrics are employed to evaluate the ECSA and establish comparisons with different methods.
- The KNN classifier is used to classify the selected features to identify grape leaf diseases.

The following structure guides the subsequent sections of this work: Section 2 Provides a concise overview of the state-of-the-art techniques employed in previous research, while Section 3 elaborates on each method used. Section 4 discusses further details of the proposed framework, while Section 5 contains the experimental results and discussions. The study is concluded in Section 6, which also provides an overview of potential areas for further research.

2. Related works

Many researchers have contributed their efforts to develop various leaf disease classification techniques. An overview of contemporary research within this domain is provided below.

Lauguico *et al.* [18] proposed a methodology to distinguish three grape leaf diseases from healthy leaves. The method also considered the confidence level to classify the diseases accurately. A comparative analysis is conducted to find the most suitable network among the pre-trained CNN networks AlexNet, GoogLeNet, and ResNet-18, which can be combined with Regions with CNN (RCNN) to detect multiple objects in an image. After evaluating the models, AlexNet was the most effective

previously trained network for use with the RCNN, achieving an accuracy score of 95.65%. The ResNet-18 and GoogLeNet models achieved accuracy values of 89.49% and 92.29%, respectively.

Jin et al. [19] proposed an architecture called GrapeGAN to overcome the limitations of traditional Generative Adversarial Networks (GANs) in generating clear and structurally integral grape leaf images. Initially, the generator of the proposed technique was built using a U-Net-like structure, the Reorg method, and a multi-feature fusion mechanism. This enabled the preservation of detailed features in the images of grape leaf diseases. A discriminator was then constructed using a combination of convolution blocks and capsule structures. This facilitated the extraction of more complex features and ensured the completeness of the grape leaves' structure. Finally, four convolutional neural networks (CNNs) detection models, namely VGG16, ResNetV1, MobileVIT, and InceptionV1, were used to classify the diseases in the generated grape leaf images. The images generated by the suggested model displayed remarkable robustness in the recognition task, achieving an accuracy of 96.13%.

Ansari et al. [20] developed a methodology for disease identification and classification in grape leaves, which consists of image acquisition, noise reduction, image enhancement, segmentation, feature extraction, classification, and detection. The noise reduction process was achieved using the averaging function, while the contrast-limited Adaptive Histogram Equalization (CLAHE) approach was applied to improve the visibility of details in images. The fuzzy C-means algorithm was used for image segmentation, and principal component analysis (PCA) was used for feature extraction. Finally, the classification of the images was performed using Particle Swarm Optimization -Support Vector Machine (PSO-SVM), Backpropagation Neural Network (BPNN), and Random Forest algorithms. For the classification and detection of grape leaf diseases, the PSO-SVM approach has the best accuracy of 95%.

Kaur and Devendran [21] introduced a methodology that covers the critical stages of automated leaf disease identification, including segmentation, feature extraction, and classification. This work focuses on integrating various feature extraction algorithms and machine learning classifiers to improve the accuracy of leaf disease prediction. They used K-Means clustering, optimized by Grey Wolf Optimization, to segment the regions affected by disease in images. They used various feature extraction techniques, including Law's mask, Grey Level Co-occurrence Matrix, Local Binary Pattern, and Gabor features, to develop a hybrid approach for robust leaf disease identification. In addition, they employed a novel ensemble classifier to improve the accuracy of disease classification. Four categories of grape diseases from the Plant-Village dataset were used for the research: leaf blight, black measles, black rot, and healthy leaves. This method is effective compared to previous approaches, achieving an accuracy of 95.69%.

Hernandez et al. [22] applied deep learning to RGB images to detect diseases in grape leaves. This study used color space conversion to transform RGB images into HLS color space to enhance the ability to differentiate between leaf discs. Subsequently, two filters were employed to improve the quality of HLS images: mean shift segmentation and median blur filter. Next, the Hough Circle Transform was used to isolate the leaf discs. Crisp and fuzzy masks are then used to identify areas of high intensity. A pre-trained VGG16 model was used to extract features from disc images automatically. After that, the extracted features were subjected to 10-fold cross-validation to evaluate the network's overall performance and verify the automatic discrimination between positive and negative discs. The Gradient Weighted Class Activation Mapping (Grad-CAM) was used to determine which region of an image is essential for classification. The developed method provided efficient, precise, and accurate results, with an accuracy rate of 81%.

Table 1 summarizes several characteristics of the context in which the studies were conducted.

Table 1. Related work of grape diseases prediction

Study	year	Dataset	Methods	DL/ML Model	Accuracy
auguico et al. [18]	2020	Grape leaf diseases from Plant Village dataset.	<ul style="list-style-type: none"> Regions with CNN (RCNN). Transfer Learning. Montage image creation. Manual Data Annotation. 	AlexNet, ResNet-18, and GoogleNet.	<ul style="list-style-type: none"> AlexNet-RCNN: 95.65%. ResNet18-RCNN: 89.49%. GoogleNet-RCNN: 92.29%.
Ansari et al. [20]	2022	400 images were created for the study.	<ul style="list-style-type: none"> The mean function The Contrast Limited Adaptive Histogram Equalization (CLAHE) method. The fuzzy C Means algorithm. Principal Component Analysis (PCA). Particle Swarm Optimization Support Vector Machine. Backpropagation Neural Network (BPNN). Random forest algorithms. 	Backpropagation Neural Network (BPNN), and Random Forest.	95%
Jin et al. [19]	2022	Grape leaf diseases from Plant Village dataset.	<ul style="list-style-type: none"> U-Net-Like generator. Convolutional block. Capsule structure. The reorg method. Multi-feature fusion mechanism. 	VGG16, ResNetV1, MobileViT, and InceptionV1.	96.13%
Kaur and Devendran [21]	2023	Grape leaf diseases from Plant Village dataset.	<ul style="list-style-type: none"> K-means clustering. Grey Wolf Optimization. Law's mask. Grey Level Co-occurrence Matrix. Local Binary Pattern. Gabor features. 	Ensemble learning.	95.69%
Hernandez et al. [22]	2024	1125 images of a grape leaf were created for the study.	<ul style="list-style-type: none"> Color space conversion. Mean shift segmentation and a median blur filter. Crisp and fuzzy masks. VGG16 model used for transfer learning. 10-fold cross-validation. Grad-CAM. 	VGG16	81%

3. Methods

3.1 Convolutional Neural Network (CNN)

Deep learning methods, specifically Convolutional Neural Networks (CNNs), have effectively extracted relevant features from complex data sources, such as images and videos. This capability has made them a valuable tool in agricultural applications, enabling the extraction of meaningful features from large datasets of plant images [23]. This technique achieves high accuracy and reduces computational effort, resulting in higher yields. The architectural design allows these networks to identify complicated features that simpler networks cannot detect. The basic principle of CNNs is to capture local features in the first layers and combine them into more complex features in the last layers.

Transfer learning [10] is an efficient approach that uses the knowledge acquired from a previously learned model to solve a related task, thereby minimizing the need for extensive retraining or fine-tuning. Deep learning algorithms rely on two critical requirements: a considerable amount of labeled data and significant computing power to operate effectively. Developing a large, high-quality dataset is a challenging and complex undertaking. In addition, implementing deep learning techniques requires access to powerful computing resources, which may be limited to laboratories or organizations with significant financial resources. Deep transfer learning aims to address these challenges by enabling knowledge transfer from one domain to another, thereby reducing the burden on data collection and computing resources.

The initial form of transfer learning involves using a pre-trained CNN as a static feature extraction mechanism. This approach preserves the primary network architecture and learns parameters. Once CNN has extracted the relevant features, these features are used as input to a subsequent network that performs the necessary classification tasks. The second and more advanced approach, fine-tuning, involves making specific changes to the pre-trained CNN to improve performance. These adjustments include architectural changes and parameter optimizations. Additionally, incorporating

new parameters into the network requires training on significant data to maximize effectiveness.

3.2 Chameleon swarm algorithm (CSA)

The recently developed metaheuristic CSA algorithm is inspired by chameleons' foraging behavior [17]. This algorithm effectively seeks optimal solutions by employing a structured three-stage process that emulates the chameleon's hunting behavior, encompassing stages of prey search, target acquisition through eye rotation, and hunting prey. The CSA algorithm's implementation is described in detail below: The CSA algorithm begins by initializing solutions using a uniform random distribution, as described in Eq. 1.

$$X_j = lb_j + \text{rand} \times (ub_j - lb_j) \quad (1)$$

Where ub_j and lb_j denote the higher and lower limits of the search region for the j th dimension, while rand represents a random number that follows a uniform distribution between 0 and 1.

■ Searching for prey:

Chameleons use a mathematically modeled position-updating approach, described by Eq. 2, to look for prey during their foraging behavior. Chameleons can adaptively modify their positions in the search region while searching for optimal solutions.

$$X_{ij}^{t+1} = \begin{cases} X_{ij}^t + p_1 \times (P_{ij}^t - G_j^t) \times r_2 + p_2 \times (G_j^t - X_{ij}^t) \times r_1 & r_i \geq P_p \\ X_{ij}^t + \mu \times (lb_j + r_3 \times (ub_j - lb_j)) \times \text{sgn}(\text{rand} - 0.5) & r_i < P_p \end{cases} \quad (2)$$

The variable X_{ij}^{t+1} denotes the position of the i th Chameleon in the j th dimension for the subsequent iteration; X_{ij}^t represents the position of the i th individual and j th dimension in the current iteration; Positive integers control exploration p_1 and p_2 , Which are 0.25 and 1.50, respectively. P_{ij}^t is the optimal position for the i th Chameleon, while G_j^t is the optimal position for all Chameleons. r_1 , r_2 , and r_3 are random numbers that are uniformly distributed between 0 and 1.

The function $\text{sgn}(\text{rand} - 0.5)$ produces either 1 or -1 to decide the direction of exploration. μ is inversely proportional to the number of iterations and can be determined using Eq. 3.

$$\mu = \gamma e^{(\alpha t/T)^\beta} \quad (3)$$

The variables t and T represent the current and maximum iterations, whereas γ , α , and β are control parameters with values of 1.0, 3.5, and 3.0, respectively.

■ The chameleon's eye rotation:

Chameleons exhibit an extraordinary capacity to detect prey by rotating their eyes, giving them a complete 360° view of their search area. The 'Chameleon's eye rotation phase' is a distinctive optical adaptation that involves four specific steps to determine the location of prey successfully.

1. Creation of a Reference Frame: The chameleons' initial positions are translated to the origin, creating a standard reference frame.
2. Rotation Matrix calculation: The algorithm calculates a rotation matrix to transform the prey's coordinates to align with the chameleon's current spatial orientation.
3. Position Transformation: The chameleons update their positions using the rotation matrix, allowing the exploration of different orientations within the search area.
4. Repositioning to Original Coordinates: The transformed positions are translated back to their original locations, maintaining their initial relative positions.

Eq. 4 concisely represents the mathematical expression for this location update mechanism, which utilizes Chameleon's eye rotation technique.

$$X_i^{t+1} = XR_i^t + XC_i^t \quad (4)$$

The variable X_i^{t+1} represents the updated positions of the i th Chameleon for the following iteration. XR_i^t , determined by evaluating Eq. 5, denotes the rotated center positions in the search space.

$$XR_i^t = m \times XT_i^t \quad (5)$$

where m stands for a rotation matrix. For each Chameleon, the coordinates of its center are denoted by XT_i^t . To determine the XT_i^t and m , we use Eqs. 6 and 7, correspondingly.

$$XT_i^t = X_i^t - XC_i^t \quad (6)$$

where the location of the i th Chameleon for the current iteration is represented by X_i^t , and the center of the Chameleon for the current positions is represented by XC_i^t .

$$m = R(\theta, z_1, z_2) \quad (7)$$

The orthogonal vectors z_1 and z_2 have n dimensions, the rotation angle is denoted by θ , and the rotation matrices in the respective axes are represented by R . Eqs. 8 and 9 are utilized in order to compute the values of θ and R , respectively. The rotation matrices concerning the x and y axes are represented by R_x and R_y , respectively.

$$\theta = \text{rand} \times \text{sgn}(\text{rand} - 0.5) \times 180^\circ \quad (8)$$

$$R_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta \\ 0 & \sin \theta & \cos \theta \end{bmatrix}, R_y = \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix} \quad (9)$$

■ Hunting prey:

During the hunting phase, chameleons employ a pursuit strategy by extending their tongues toward nearby prey, reaching a distance of twice their height. This phase facilitates search space exploitation, and the chameleon's positions are updated according to Eq. 10.

$$X_{i,j}^{t+1} = X_{i,j}^t + \left(V_{i,j}^t - V_{i,j}^{t-1} \right) / 2a \quad (10)$$

where the value of the Chameleon that has been changed for the following iteration is represented by $X_{i,j}^{t+1}$, $V_{i,j}^t$ is the velocity of the i th Chameleon in the current iteration for the j th dimension; the acceleration rate of the Chameleon's tongue projection is represented by a . The number grows to a maximum of 2590 ms per second squared. The value of the variable "a" is determined by applying Eq. 11.

$$a = 2590 \times \left(1 - e^{-\log(t)} \right) \quad (11)$$

The velocity values for the chameleon in the current iteration are determined using Eq. 12.

$$V_{ij}^{t+1} = \omega V_{ij}^t + c_1 \times (G_j^t - X_{ij}^t) \times r_1 + c_2 \times (P_{ij}^t - X_{ij}^t) \times r_2 \quad (12)$$

The positive constants c_1 and c_2 determine the impact of G_j^t and P_{ij}^t on the current Chameleon. The variable ω represents the inertia weight, which decreases linearly as the number of iterations increases. The parameter ρ controls the level of exploitation, and its value is set to a constant of 1.

$$\omega = (1 - t/T)^{(\rho \sqrt{t/T})} \quad (13)$$

The CSA algorithm simulates chameleons' foraging behavior using Equations 2, 4, and 10. These formulas aid the algorithm in choosing the most optimal solutions for the optimization problem.

3.3 K-nearest neighbors (KNN)

The K-Nearest Neighbors (KNN) algorithm [24] is a widely used machine learning method for classification and prediction tasks. It is based on the principle that data points with similar properties tend to have comparable labels or target values. The KNN algorithm retains the complete training dataset as a future reference during its learning phase. Subsequently, when predicting the class of a new data point, the Euclidean distance metric is employed to calculate the distance between data points and each training instance. The algorithm determines the K nearest neighbors of the input data point based on their distances. When classifying, the algorithm predicts the label for the input data point based on the most prevalent class label among the K neighbors. To predict the value for the input data point in the regression, the average or weighted average of the target values of the K neighbors is calculated. Due to its simplicity and easy to understand, the KNN algorithm is widely used in various applications. Careful parameter tuning is required to achieve optimal results, as the choice of K and distance metric can affect system performance.

4. Proposed framework for Grape diseases prediction based on ECSA method

This section presents the proposed framework to achieve accurate grape disease prediction. The framework consists of five stages: a) image capture, b) image augmentation, c) deep feature extraction, d) feature selection (FS), and e) classification. Figure 1 illustrates the proposed grape disease prediction framework, with each stage described as follows:

4.1 Image acquisition

The dataset included in this work is Plant Village [23], a publicly available dataset containing various images illustrating various grape diseases. Figure 2 shows an illustrative example for each disease category. This dataset includes 4062 images showing various diseases affecting grapes. These images were then divided into four groups: Grape Blackrot, Grape Measles, Grape Healthy, and Grape Leaf Blight. The dataset consists of 1,180 images of Grape Blackrot, 1,383 images of Grape Measles, 423 images of Grape Healthy, and 1,076 images of Grape Leaf Blight. The above categorization allows evaluation of the effectiveness of the proposed strategy on different disease categories and healthy samples.

4.2 Image augmentation using Conditional Generative Adversarial Network (CGAN)

Data augmentation techniques are often combined with conventional or deep learning approaches to mitigate the problem of bias towards a particular class and improve classification accuracy [25, 26].

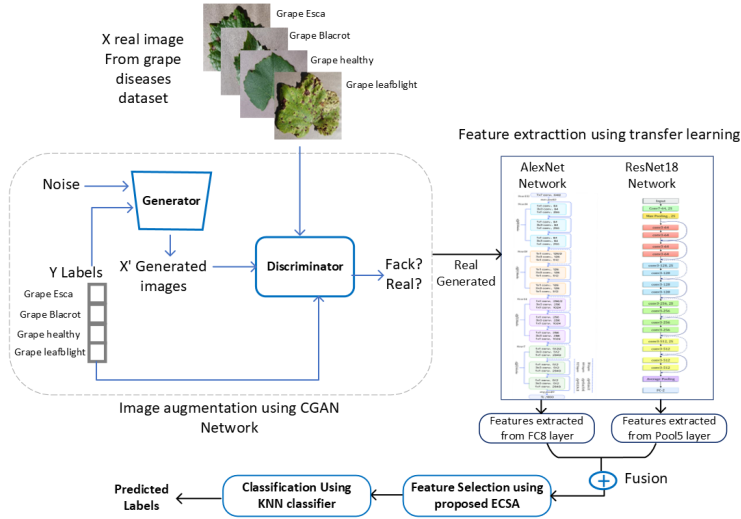


Figure 1. The proposed grape disease prediction framework

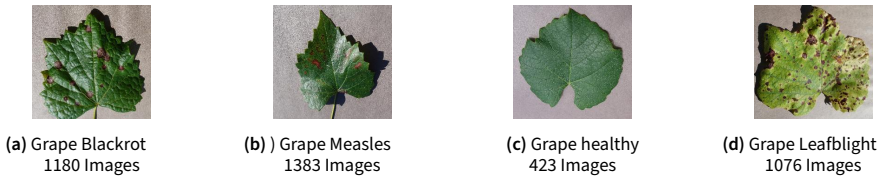


Figure 2. Categorization of grape leaf diseases (Total Images 4062)

This study used a CGAN as a data augmentation strategy to balance the number of images in each class, as shown in Figure 1. Goodfellow et al. [27] first introduced the term GANs in 2014. GANs consist of two neural networks: a generator network, which is responsible for creating images, and a discriminator network, which distinguishes between authentic and fraudulent inputs. The CGAN proposed by Mirza and Osindero [28] is an extended version of this technique that includes explicit class labels. CGANs use a generator to produce images that fall into one of the n categories, allowing for more targeted and diverse data augmentation.

4.3 CNN feature extraction using Transfer Learning

According to Khan et al. (2020) [29], feature extraction is an essential and challenging aspect of pattern recognition and machine learning when it comes to representing images in various fields such as engineering, medical imaging, and agriculture [30, 31, 32]. Classical learning and DL are two of several feature extraction methods described in the literature. Deep learning (DL) is an innovative field of study within machine learning (ML) that focuses on achieving precise classification [32, 33]. Agricultural imaging identifies and exploits the most important features to achieve the desired classification result. This research obtained deep features using two pre-trained models: ResNet18 [34] and AlexNet [35].

- The ResNet18 neural network architecture is famous for its 72 layers, with 18 of them being deep layers. The architecture of this network facilitated the optimal operation of many convolutional layers. The input size for images in ResNet18 is defined as 224×224 pixels.
- AlexNet is a famous neural network architecture. Alex Krizhevsky introduced this architecture

for the ImageNet Large Scale Visual Recognition Challenge, primarily focusing on convolutional neural networks. The fundamental components of this include convolutions, max pooling, and fully connected layers. The dimensions of the input image in AlexNet are defined at 227×227 pixels.

The following steps comprise the feature extraction stage in this study:

- The input images were resized to 224×224 pixels before being fed into the ResNet18 model.
- The input images were resized to 227×227 pixels and transmitted to the AlexNet model.
- The final activation layers, Pool5 and FC8 from ResNet18 and AlexNet, respectively, were used for deep-feature extraction.
- The extracted features from each model were then fused.

4.4 Enhanced Chameleon Swarm Algorithm (ECSA) for Feature Selection

This section provides a detailed description of the proposed ECSA approach to solve FS-related problems. Selecting the most meaningful features can be difficult, especially when redundant data is available. The main goal of extending the original CSA is to address its weaknesses and improve its search capabilities, especially for complicated real-world situations such as FS. This section first discusses the limitations of the original CSA. The procedures and improvements implemented in the proposed ECSA approach are then explained to increase the population evolution process and achieve higher solution quality.

4.4.1 Limitations of the original CSA

Despite its effectiveness in industrial and global optimization problems, the traditional CSA optimizer faces a challenge during exploration. Its inherent tendency to focus on the region indicated by the chameleon swarm can restrict the population to local optimal, resulting in premature convergence and potentially limiting the discovery of more optimal solutions.

This work presents an enhanced version of the CSA algorithm called ECSA to address these weaknesses. The ECSA approach aims to address the limitations of the original CSA by incorporating adaptations that increase the speed of convergence and strike an appropriate balance between the exploration and exploitation of new opportunities. By resolving these limitations, the ECSA approach improves the overall performance and effectiveness of the optimization process.

4.4.2 The architecture of the ECSA method

This work presents an enhanced approach for optimizing the CSA, called the Enhanced CSA (ECSA) method. The ECSA method effectively addresses the limitations of the original CSA by leveraging chaotic maps and Levy flight. Chaotic maps are used during the global search phase to improve the population diversity and exploration capabilities of the CSA algorithm. At the same time, the local search phase uses the Levy flight distribution component to enhance the ability to search locally and avoid being stuck in local solutions. The flowchart in Figure 3 illustrates the proposed ECSA approach for FS. The following sections provide detailed explanations of each phase of the ECSA.

- A. **Updating positions in the searching for prey phase based on chaotic maps:** Optimization algorithms often rely on random values to guide the search strategy within the search space. However, using chaotic maps in optimization algorithms offers an advantage by using chaotic values instead of random ones. Additionally, incorporating chaotic maps enables the algorithm to evade the local optimum trap.

Chaotic maps can conduct more efficient searches than random numbers due to their ability to explore the entire search space non-repetitively. Consequently, combining chaotic maps with

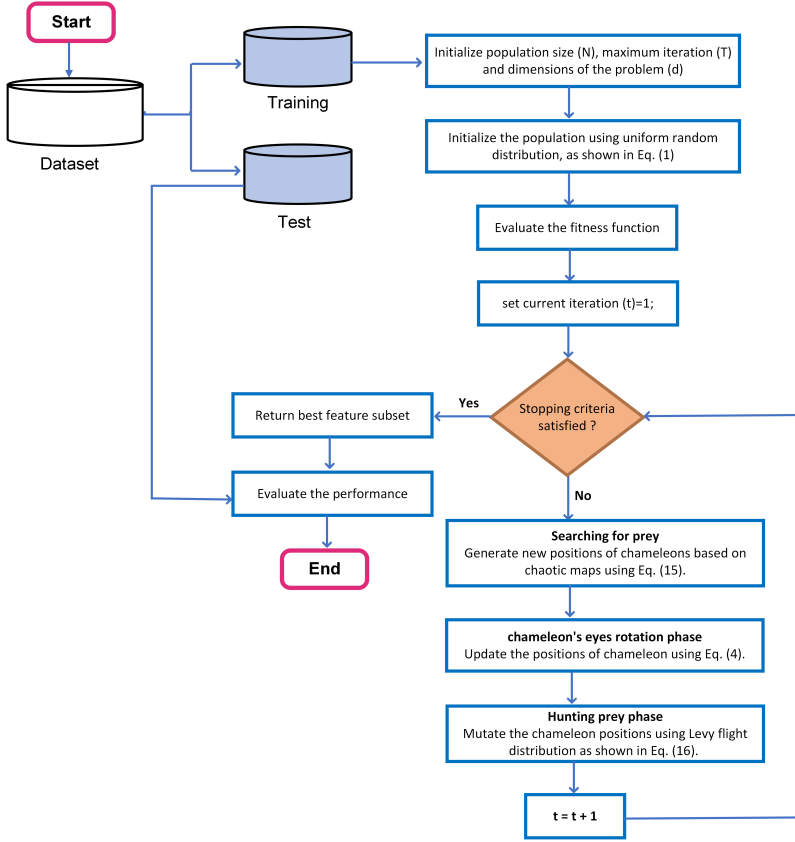


Figure 3. The proposed ECSA method for FS

optimization techniques has shown promising results. In this work, ten chaotic maps are integrated with the CSA algorithm. The Chaotic maps control each random parameter. The primary task of chaotic maps is to replace these random values in the exploration phase of CSA with a chaotic map value. Throughout the iterations, the parameters r_1 , r_2 and r_3 in Eq. 2 are replaced by ch_{k+1} , One of the ten chaotic maps, as illustrated in Table 2. Multiple chaotic mapping functions are employed to enhance the overall efficiency of the optimization strategy. The mathematical expression is shown in Eq. 14.

$$X_{i,j}^{t+1} = \begin{cases} X_{i,j}^t + p_1 \times (P_{i,j}^t - G_j^t) \times ch_{k+1} + p_2 \times (G_j^t - X_{i,j}^t) \times ch_{k+1} & \text{rand} \geq P_p \\ X_{i,j}^t + \mu \times (lb_j + ch_{k+1} \times (ub_j - lb_j)) \times \text{sgn}(\text{rand} - 0.5) & \text{rand} < P_p \end{cases} \quad (14)$$

where ch_{k+1} is a value acquired from well-known chaotic maps, as given in Table 2.

- B. Updating positions in the Hunting prey phase using Levy distribution:** In this phase, the ECSA approach was employed to adjust the location of every individual in the overall population and identify the optimal position. Eq. 10 which represents the exploitation phase, is adjusted using the Levy-based mutation operator, as shown in Eq. 16. The Levy distribution exhibits the collective movement patterns observed in insects and birds, characterized by short-range movements interspersed with occasional long-range leaps, as they forage for food. It improves

Table 2. Ten chaotic maps.

#Map	Name	Definition	Range
1	Chebyshev	$ch_{k+1} = \cos(k \cos^{-1}(ch_k))$	(-1, 1)
2	Circle	$ch_{k+1} = \text{mod} \left(ch_k + r - \left(\frac{l}{2\pi} \right) \sin 2\pi ch_k, 1 \right), l = 0.5 \text{ and } r = 0.2$	(0, 1)
3	Gauss/mouse	$ch_{k+1} = f(x) = \begin{cases} 1, & ch_k = 0 \\ \frac{1}{\text{mod}(ch_k, 1)}, & \text{otherwise} \end{cases}$	(0, 1)
4	Iterative	$ch_{k+1} = \sin \left(\frac{l\pi}{ch_k} \right), l = 0.7$	(-1, 1)
5	Logistic	$ch_{k+1} = lch_k(1 - ch_k), l = 4$	(0, 1)
6	Piecewise	$ch_{k+1} = f(x) = \begin{cases} \frac{ch_k}{l}, & 0 \leq ch_k < 1 \\ \frac{ch_k - l}{0.5 - l}, & 1 \leq ch_k < 0.5, \quad , l = 0.4 \\ \frac{1 - l - ch_k}{0.5 - l}, & 0.5 \leq ch_k < 1 - l \\ \frac{1 - ch_k}{l}, & 1 - l \leq ch_k < 1 \end{cases}$	(0, 1)
7	Sine	$ch_{k+1} = \frac{1}{4} \sin(\pi ch_k), l = 4$	(0, 1)
8	Singer	$ch_{k+1} = \mu \left(7.86ch_k - 23.31ch_k^2 + 28.75ch_k^3 - 13.302875ch_k^4 \right), \mu = 1.07$	(0, 1)
9	Sinusoidal	$ch_{k+1} = lch_k^2 \sin(\pi ch_k), l = 2.3$	(0, 1)
10	Tent	$ch_{k+1} = \begin{cases} \frac{ch_k}{0.7}, & ch_k < 0.7 \\ \frac{10}{3}(1 - ch_k), & ch_k \geq 0.7 \end{cases}$	(0, 1)

the optimization algorithm’s ability to exploit new possibilities. The Levy distribution function, denoted by $L(\beta)$, is determined according to Eq. 15.

$$L(\beta) = \frac{\mu \times \sigma}{|\nu|^{1/\beta}} \tag{15}$$

Assuming that μ and ν are both distributed according to a regular normal distribution. $\sigma = \left[\frac{\Gamma(1+\beta) \times \sin(\pi\beta/2)}{\Gamma[(1+\beta)/2] \beta \times 2^{\beta-1/2}} \right]^{1/\beta}$, $\beta = 1.5$, and Γ is a standard gamma function.

$$X_i^L = X_i \times (1 + L(\beta)) \tag{16}$$

A mutation boosts the optimizer’s accuracy and aids the original algorithm in avoiding a local solution. Furthermore, it increases the population diversity, leading to a better search for prospective regions, enhancing the search speed, and accelerating the optimizer’s convergence tendency.

- C. **Terminal Stage:** The proposed approach involves iterative evaluation and updating until a specified termination condition is satisfied. Within the context of this investigation, “stopping” refers to attaining the maximum number of iterations. This stopping criterion aims to evaluate the efficiency of the proposed approach in generating the optimal subset of characteristics within a specific timeframe. By iteratively assessing and updating the feature subset, this method strives to find the optimum solution that captures the optimal features for the given problem. Algorithm 1 shows the ECSA method’s pseudocode.

4.5 Classification

This study uses a KNN classifier to classify images based on their features. Because of its simplicity, applicability, and lack of tuning factors, KNN was chosen as the classifier. The dataset was partitioned into two groups: the training and testing sets. 80% of the data was allocated for training, whereas the remaining 20% was allocated for testing.

Algorithm 1 A pseudocode of ECSA's optimization process.

-
- 1: Initialize the population set (X) using uniform random distribution as shown in Eq. 1.
 - 2: Evaluate the fitness of random initialized chameleons (X).
 - 3: **for** $t = 1$ to t_{max} **do**
 - 4: Perform the Searching for prey phase using Eq. 14.
 - 5:
$$X_{i,j}^{t+1} = \begin{cases} X_{i,j}^t + p_1 \times (P_{i,j}^t - G_j^t) \times ch_{k+1} + p_2 \times (G_j^t - X_{i,j}^t) \times ch_{k+1} & \text{rand} \geq P_p \\ X_{i,j}^t + \mu \times (lb_j + ch_{k+1} \times (ub_j - lb_j)) \times \text{sgn}(\text{rand} - 0.5) & \text{rand} < P_p \end{cases}$$
 - 6: Perform the Chameleon's eyes rotation phase using Eq. 4.
 - 7: Perform hunting prey phase using Eq. 16.
 - 8: $X_i^L = X_i \times (1 + L(\beta))$
 - 9: Perform elitism by selecting the Chameleon with minimum fitness among (X_i^L), (X_i^G), (X_i^C), and (X_i^{t+1})
 - 10: $t=t+1$
 - 11: **end for**
 - 12: Return the optimal solution obtained so far
-

5. Experimental results and discussion

This study conducted an experiment to evaluate the efficiency of the proposed framework for grape disease prediction. The experiment focused on predicting grape diseases using the plant village datasets [23]. We divide this dataset into an 80:20 ratio, allocating 80% for training and 20% for testing. The experiment used MATLAB 2020a on a desktop with a Core-i7 processor and 16 GB of RAM. In the experiment, the effectiveness of the proposed framework using the ECSA algorithm was evaluated by comparing it with that optimized using various existing optimization metaheuristic algorithms, including the original CSA algorithm, Chimp optimization algorithm (ChOA) [36], Archimedes optimization algorithm (AOA) [37], Bird Swarm Algorithm (BSA) [38], HHO [39], and WOA [40].

The algorithms were tested using the same parameter settings to ensure a thorough comparison. Each algorithm was consistently evaluated for the feature selection process with a population size of 30 and a maximum of 100 iterations. The algorithms' dimensionality was aligned with the number of features in the original dataset. Detailed parameter configurations for all algorithms can be found in Table 3.

Table 3. Metaheuristic algorithms parameters 2

Algorithms	Parameters	Values
Common Settings	Population size (N)	30
	Maximum iterations (T)	100
	Problem Dimensions (D)	Dataset features
CSA	P_1	0.850
	P_2	0.850
	α	3.50
	γ	1.0
	β	3.0
CHOA	a	Decreases linearly from 2 to 0
AOA	$C1$	2
	$C2$	6
	$C3$	1
	$C4$	2
	U	0.9
BSA	L	0.1
	$C1$	1.5
	$C2$	1.5
	$a1$	1
	$a2$	1
HHO	β	1.5
WOA	a	[0, 2]

5.1 Performance measures

This study evaluates the efficiency of the proposed framework by calculating the subsequent performance metrics: classification accuracy, sensitivity, and specificity. While accuracy indicates the proportion of correctly classified instances within a dataset, sensitivity measures the test's ability to recognize the presence of a disease in grape leaves. The specificity metric evaluates the test's ability to detect grape leaves unaffected by any disease precisely [41]. This study calculates the number of chosen features in each dataset to ensure the algorithm can efficiently eliminate the number of features across all iterations. The impact of selected features on classification accuracy is an essential metric for determining a model's efficiency. An algorithm with a limited number of features may provide low accuracy.

$$\text{Accuracy} = \frac{TP+FN}{TP+FN+FP+TN}$$

$$\text{Sensitivity} = \frac{TP}{TP+FN}$$

$$\text{Specificity} = \frac{TN}{TN+FP}$$

where TP is a true positive, FP is a false positive, TN is a true negative, and FN is a false negative.

5.2 Results of Grape Diseases Prediction

This study describes an experiment for evaluating the proposed framework. In the first step, we applied the CGAN model to balance the number of grape disease images in each class, resulting in 400 images per class. In the second step, we applied transfer learning, using two pre-trained CNN models, ResNet18 and AlexNet, to extract features from the grape disease images. We then fused the extracted features from each model, resulting in 1512 features from the two CNN models.

We then optimized the extracted features from the CNNs using the proposed ECSA method for feature selection, resulting in a new feature set of 242 features. The KNN classifier then performs the classification process using the 242 selected features from the ECSA method, achieving the best results with an accuracy of 97.76%, a recall of 90.91%, and a specificity value of 97.46%, as shown in Table 4.

Finally, we compare the performance of the proposed framework, optimized using the ECSA method, with that optimized using various metaheuristic methods, such as the original CSA, ChOA, AOA, BSA, HHO, and the WOA methods. Table 4 displays the number of selected features and the classification accuracy results. The ChOA algorithm achieved the highest rank regarding the number of selected features, while the proposed ECSA method ranked second. The set of features chosen does not primarily determine the efficiency of an FS method, but their influence on the classification accuracy does. An algorithm can provide limited features and still achieve low accuracy. Using the best set of features from the ECSA method, we achieve the best classification rate of 97.76% in this step.

Figure 4(a) illustrates the convergence curve obtained for the grape disease datasets using the proposed ECSA and other competing methods. The results demonstrate that the ECSA achieved a certain level of stability across all datasets. The suggested ECSA algorithm yielded the most optimal

Table 4. Performance comparison of the proposed framework using the ECSA method and other competitors on the Grape disease dataset

Model	#No. of features	Accuracy using KNN classifier %.	Recall	Specificity
Proposed-ECSA	242	97.76%	90.91%	97.46%
Proposed-CSA	575	97.44%	89.65%	97.29%
Proposed-ChOA	88	97.46%	90.22%	97.16%
Proposed-AOA	358	95.52%	85.51%	96.30%
Proposed-BSA	710	96.19%	87.01%	96.86%
Proposed-HHO	530	97.13%	87.81%	97.16%
Proposed-WOA	392	97.16%	85.63%	96.82%

average global solutions for the Grape dataset compared to the other methods. Figure 4(b) demonstrates that the boxplots produced by the proposed ECSA method are narrower than those created by the different techniques. Hence, the ECSA surpassed its rivals.

The experimental results suggest that the proposed framework using the ECSA method outperforms the seven previous metaheuristic algorithms in solving a real-world problem.

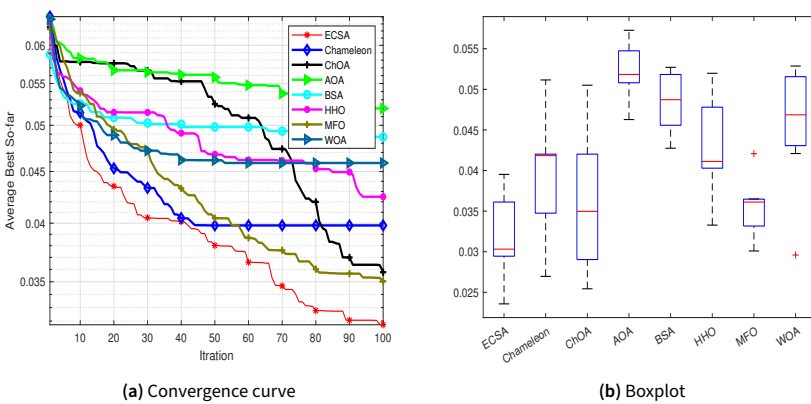


Figure 4. Convergence curve and Boxplot for ECSA against other competitors – Grape diseases prediction

Figure 5 presents a comparative analysis of the proposed framework, incorporating the ECSA method, against other competitive approaches. The comparison is based on accuracy, sensitivity, and specificity. These performance metrics are crucial for evaluating the efficiency of the proposed framework in recognizing grape diseases. As evident from the figure, the proposed framework, optimized using the ECSA method, achieved a peak accuracy rate of 97.76%, establishing it as the most reliable model for identifying positive cases. The ChOA method ranked second, while the AOA method exhibited the lowest accuracy.

Regarding sensitivity, the proposed framework excelled with a sensitivity rate of 90.91%, highlighting its ability to identify true positive cases effectively. The ChOA method achieved the second rank,

while AOA had the last rank. Regarding specificity, the proposed framework optimized using the ECSA method showed the highest specificity score of 97.46%. This metric indicates the negative cases. The CSA method ranked second in terms of specificity, while the AOA method had the lowest specificity value. The results indicate that the proposed framework is the more effective choice for detecting diseases in grape leaves.

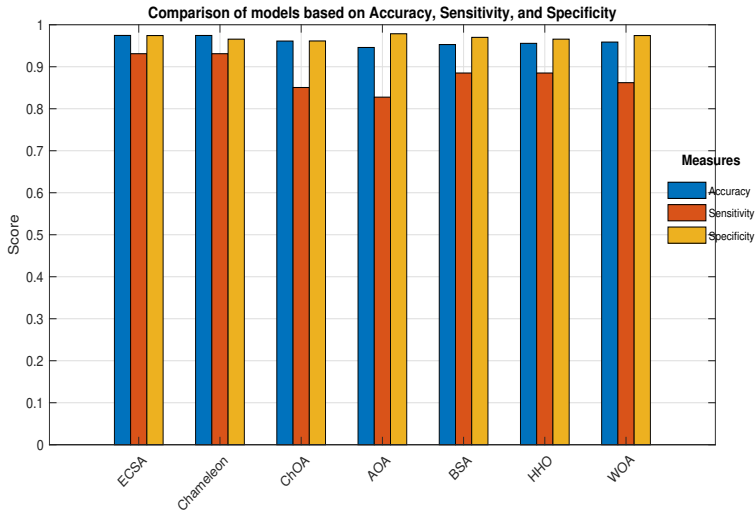


Figure 5. Performance comparison of the proposed framework using the ECSA method and other competitors based on accuracy, sensitivity, and specificity.

Table 5. Comparison with some previous studies of grape disease prediction

study	Dataset	Methodology	#No. of classes	Accuracy (%)
Kaur and Devendran [21]	Grape leaf diseases from Plant Village dataset.	<ul style="list-style-type: none"> Image segmentation: K-means clustering and Grey Wolf Optimization. Feature extraction: Law's mask, Grey Level Co-occurrence Matrix, Local Binary Pattern, and Gabor features. Classification: Ensemble Learning. 	4	95.69%
Jin et al. [19]	Grape leaf diseases from Plant Village dataset.	<ul style="list-style-type: none"> Image Pre-processing: U-Net-Like generator. Feature extraction: Convolutional block. Spatial preservation: Capsule structure. Feature compression: The reorg method. Multi-feature fusion mechanism. Classification: VGG16, ResNetV1, MobileViT, and InceptionV1. 	4	96.13%
Lauguico et al. [18]	Grape leaf diseases from Plant Village dataset.	<ul style="list-style-type: none"> Image pre-processing: Montage image creation and Manual Data Annotation. Feature extraction: Transfer Learning using AlexNet, ResNet18, and GoogLeNet. Classification: Regions with CNN (RCNN). 	4	<ul style="list-style-type: none"> AlexNet-RCNN: 95.65%. ResNet18-RCNN: 89.49%. GoogLeNet-RCNN: 92.29%.
Proposed framework	Grape leaf diseases from Plant Village dataset.	<ul style="list-style-type: none"> Image pre-processing: Data augmentation using the CGAN model. Feature extraction: Transfer learning using ResNet18 and AlexNet. Feature fusion mechanism. Feature selection: Enhanced Chameleon Swarm Algorithm (ECSA) method. Classification: KNN classifier. 	4	97.76%

5.3 Discussion

In this study, the proposed framework achieved the best classification accuracy of 97.76% on grape diseases from the Plant Village dataset involving 4 classes, with an improvement of 1.63% compared to the previous work of Jin *et al.* [19], which achieved an accuracy of 96.13% using the same dataset and the same number of classes. This improvement in classification accuracy is due to our model architecture. First, our model used the CGAN model, which was employed as a data augmentation technique on the selected Grape dataset to address the overfitting problem. Deep features were then extracted from the augmented images using ResNet18 and AlexNet, followed by a fusion method to combine and strengthen the resulting feature representations. After extracting robust features, the final feature vectors are optimized using the ECSA method to obtain a more robust subset of features that provides higher accuracy than other studies, as shown in Table 5. For a fair comparison, this study used the same dataset with the same number of images for grape blackrot, grape measles, healthy, and grape leaf blight.

6. Conclusions and future work

The main aim of this study is to establish an automatic framework that uses pre-trained CNN models and an enhanced Chameleon swarm algorithm (ECSA) to detect diseases in grape leaves. Using a conditional generative adversarial network (CGAN), data augmentation was employed to overcome the class imbalance within the Grape dataset. Deep features were then obtained from the augmented images using two pre-trained convolutional neural networks (CNNs): ResNet18 and AlexNet. A fusion method was then applied to combine the extracted feature vectors. This study employed an enhanced Chameleon Swarm Optimizer (ECSA) to optimize the extracted feature vectors. The ECSA method incorporates chaotic maps during the global search to enhance population diversity and exploration ability. At the same time, leveraging Levy flights in the local search phase improves local exploration and prevents premature convergence. Finally, the selected features were classified using the K-Nearest Neighbors (KNN) algorithm to identify grape leaf diseases. The effectiveness of the proposed framework is assessed on a real-world dataset of grape diseases, employing multiple evaluation criteria. The proposed framework consistently achieves the best accuracy of 97.76% on the grape disease dataset. The experimental findings demonstrate that the ECSA method works better than other advanced comparative methods to obtain optimal solutions. Nevertheless, it is essential to admit that the ECSA requires significant computational time, particularly for high-dimensional datasets. Future research can explore the application of the ECSA in optimizing benchmark issues and solving real-world problems in several domains, including engineering, image segmentation, and clustering.

Open data statement

The datasets analyzed during the current study are available in the UCI repository at <https://archive.ics.uci.edu/datasets>, and plant village datasets at <https://www.kaggle.com/datasets/emmarex/plantdisease>.

References

- [1] Rajeev Sharma and Gurpreet Singh. "Access to modern agricultural technologies and farmer household welfare: Evidence from India". In: *Millennial Asia* 6.1 (2015), pp. 19–43.
- [2] JRN Taylor. "Grain production and consumption: Africa". In: *Encyclopedia of Food Grains* 4 (2016), pp. 359–366.

- [3] Guan Wang, Yu Sun, and Jianxin Wang. "Automatic image-based plant disease severity estimation using deep learning". In: *Computational intelligence and neuroscience 2017.1* (2017), p. 2917536.
- [4] Chaowalit Khitthuk, Arthit Srikaew, Kitti Attakitmongcol, and Prayoth Kumsawat. "Plant leaf disease diagnosis from color imagery using co-occurrence matrix and artificial intelligence system". In: *2018 International electrical engineering congress (iEECON)*. IEEE. 2018, pp. 1–4.
- [5] RajinderKumar M Math and Nagaraj V Dharwadkar. "Early detection and identification of grape diseases using convolutional neural networks". In: *Journal of Plant Diseases and Protection* 129.3 (2022), pp. 521–532.
- [6] Lawrence C Ngugi, Moataz Abelwahab, and Mohammed Abo-Zahhad. "Recent advances in image processing techniques for automated leaf pest and disease recognition–A review". In: *Information processing in agriculture* 8.1 (2021), pp. 27–51.
- [7] YA Nanekaran, Defu Zhang, Junde Chen, Yuan Tian, and Najla Al-Nabhan. "Recognition of plant leaf diseases based on computer vision". In: *Journal of Ambient Intelligence and Humanized Computing* (2020), pp. 1–18.
- [8] Manjunath Jogin, MS Madhulika, GD Divya, RK Meghana, S Apoorva, et al. "Feature extraction using convolution neural networks (CNN) and deep learning". In: *2018 3rd IEEE international conference on recent trends in electronics, information & communication technology (RTEICT)*. IEEE. 2018, pp. 2319–2323.
- [9] Panissara Thanapol, Kittichai Lavangnananda, Pascal Bouvry, Frédéric Pinel, and Franck Leprevost. "Reducing overfitting and improving generalization in training convolutional neural network (CNN) under limited sample sizes in image recognition". In: *2020-5th International Conference on Information Technology (InCIT)*. IEEE. 2020, pp. 300–305.
- [10] Karl Weiss, Taghi M Khoshgoftaar, and DingDing Wang. "A survey of transfer learning". In: *Journal of Big data* 3 (2016), pp. 1–40.
- [11] Manali Shaha and Meenakshi Pawar. "Transfer learning for image classification". In: *2018 second international conference on electronics, communication and aerospace technology (ICECA)*. IEEE. 2018, pp. 656–660.
- [12] Dipti Theng and Kishor K Bhoyar. "Feature selection techniques for machine learning: a survey of more than two decades of research". In: *Knowledge and Information Systems* 66.3 (2024), pp. 1575–1637.
- [13] Fatma A Hashim, Nabil Neggaz, Reham R Mostafa, Laith Abualigah, Robertas Damasevicius, and Abdelazim G Hussien. "Dimensionality reduction approach based on modified hunger games search: case study on Parkinson's disease phonation". In: *Neural Computing and Applications* 35.29 (2023), pp. 21979–22005.
- [14] Reham R Mostafa, Marwa A Gaheen, Mohamed Abd ElAziz, Mohammed Azmi Al-Betar, and Ahmed A Ewees. "An improved gorilla troops optimizer for global optimization problems and feature selection". In: *Knowledge-Based Systems* 269 (2023), p. 110462.
- [15] Gehad Ismail Sayed, Alaa Tharwat, and Aboul Ella Hassanien. "Chaotic dragonfly algorithm: an improved metaheuristic algorithm for feature selection". In: *Applied Intelligence* 49 (2019), pp. 188–205.
- [16] Prachi Agrawal, Hattan F Abutarboush, Talari Ganesh, and Ali Wagdy Mohamed. "Metaheuristic algorithms on feature selection: A survey of one decade of research (2009-2019)". In: *Ieee Access* 9 (2021), pp. 26766–26791.
- [17] Malik Shehadeh Braik. "Chameleon Swarm Algorithm: A bio-inspired optimizer for solving engineering design problems". In: *Expert Systems with Applications* 174 (2021), p. 114685.
- [18] Sandy Lauguico, Ronnie Concepcion, Rogelio Ruzcko Tobias, Argel Bandala, Ryan Rhay Vicerra, and Elmer Dadios. "Grape leaf multi-disease detection with confidence value using transfer learning integrated to regions with convolutional neural networks". In: *2020 IEEE region 10 conference (TENCON)*. IEEE. 2020, pp. 767–772.

- [19] Haibin Jin, Yue Li, Jianfang Qi, Jianying Feng, Dong Tian, and Weisong Mu. “GrapeGAN: Unsupervised image enhancement for improved grape leaf disease recognition”. In: *Computers and Electronics in Agriculture* 198 (2022), p. 107055.
- [20] Arshiya S Ansari, Malik Jawarneh, Mahyudin Ritonga, Pragti Jamwal, Mohammad Sajid Mohammadi, Ravi Kishore Veluri, Virendra Kumar, and Mohd Asif Shah. “[Retracted] Improved Support Vector Machine and Image Processing Enabled Methodology for Detection and Classification of Grape Leaf Disease”. In: *Journal of Food Quality* 2022.1 (2022), p. 9502475.
- [21] Navneet Kaur and V Devendran. “A novel framework for semi-automated system for grape leaf disease detection”. In: *Multimedia Tools and Applications* 83.17 (2024), pp. 50733–50755.
- [22] Inés Hernández, Salvador Gutiérrez, and Javier Tardaguila. “Image analysis with deep learning for early detection of downy mildew in grapevine”. In: *Scientia Horticulturae* 331 (2024), p. 113155.
- [23] Andreas Kamilaris and Francesc X Prenafeta-Boldú. “A review of the use of convolutional neural networks in agriculture”. In: *The Journal of Agricultural Science* 156.3 (2018), pp. 312–322.
- [24] Gongde Guo, Hui Wang, David Bell, Yaxin Bi, and Kieran Greer. “KNN model-based approach in classification”. In: *On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE: OTM Confederated International Conferences, CoopIS, DOA, and ODBASE 2003, Catania, Sicily, Italy, November 3-7, 2003. Proceedings*. Springer, 2003, pp. 986–996.
- [25] Sharada P Mohanty, David P Hughes, and Marcel Salathé. “Using deep learning for image-based plant disease detection”. In: *Frontiers in plant science* 7 (2016), p. 1419.
- [26] Pierre Barré, Ben C Stöver, Kai F Müller, and Volker Steinhage. “LeafNet: A computer vision system for automatic plant species identification”. In: *Ecological Informatics* 40 (2017), pp. 50–56.
- [27] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. “Generative adversarial networks”. In: *Communications of the ACM* 63.11 (2020), pp. 139–144.
- [28] Mehdi Mirza and Simon Osindero. “Conditional generative adversarial nets”. In: *arXiv preprint arXiv:1411.1784* (2014).
- [29] Muhammad Irfan Sharif, Jian Ping Li, Muhammad Attique Khan, and Muhammad Asim Saleem. “Active deep neural network features selection for segmentation and recognition of brain tumors using MRI images”. In: *Pattern Recognition Letters* 129 (2020), pp. 181–189.
- [30] Muhammad Attique Khan, Muhammad Sharif, Tallha Akram, Mudassar Raza, Tanzila Saba, and Amjad Rehman. “Hand-crafted and deep convolutional neural network features fusion and selection strategy: An application to intelligent human action recognition”. In: *Applied Soft Computing* 87 (2020), p. 105986.
- [31] Muhammad Attique Khan, Seifedine Kadry, Yu-Dong Zhang, Tallha Akram, Muhammad Sharif, Amjad Rehman, and Tanzila Saba. “Prediction of COVID-19-pneumonia based on selected deep features and one class kernel extreme learning machine”. In: *Computers & Electrical Engineering* 90 (2021), p. 106960.
- [32] Muhammad Attique Khan, Tallha Akram, Muhammad Sharif, Kashif Javed, Mudassar Raza, and Tanzila Saba. “An automated system for cucumber leaf diseased spot detection and classification using improved saliency method and deep features selection”. In: *Multimedia Tools and Applications* 79 (2020), pp. 18627–18656.
- [33] Karen Simonyan and Andrew Zisserman. “Very deep convolutional networks for large-scale image recognition”. In: *arXiv preprint arXiv:1409.1556* (2014).
- [34] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. “Deep residual learning for image recognition”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

- [35] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "ImageNet classification with deep convolutional neural networks". In: *Communications of the ACM* 60.6 (2017), pp. 84–90.
- [36] Mohammad Khishe and Mohammad Reza Mosavi. "Chimp optimization algorithm". In: *Expert systems with applications* 149 (2020), p. 113338.
- [37] Fatma A Hashim, Kashif Hussain, Essam H Houssein, Mai S Mabrouk, and Walid Al-Atabany. "Archimedes optimization algorithm: a new metaheuristic algorithm for solving optimization problems". In: *Applied intelligence* 51 (2021), pp. 1531–1551.
- [38] Elif Varol Altay and Bilal Alatas. "Bird swarm algorithms with chaotic mapping". In: *Artificial Intelligence Review* 53.2 (2020), pp. 1373–1414.
- [39] Ali Asghar Heidari, Seyedali Mirjalili, Hossam Faris, Ibrahim Aljarah, Majdi Mafarja, and Huiling Chen. "Harris hawks optimization: Algorithm and applications". In: *Future generation computer systems* 97 (2019), pp. 849–872.
- [40] Seyedali Mirjalili and Andrew Lewis. "The whale optimization algorithm". In: *Advances in engineering software* 95 (2016), pp. 51–67.
- [41] Ahmed Reda, Sherif Barakat, and Amira Rezk. "A transfer learning-enabled optimized extreme deep learning paradigm for diagnosis of covid-19". In: *Computers, Materials & Continua* 70.1 (2022), pp. 1381–1399.