International Journal of Computer Science, Engineering and Information Technology (IJCSEIT),
Vol.14, No.5, October 2024

MODIFIED SAND CAT SWARM OPTIMIZATION (MSCSO) BASED FEATURE SELECTION FOR CLASSIFICATION OF DISEASES IN TOMATO LEAVES

Chetan Singh Negi, H. L. Mandoria and Sunita Jalal

College of Technology, GBPUAT, Pantnagar

ABSTRACT

Food obtained from the plants is one of the most basic requirements for the survival of humanity. However, plant diseases produce low-quality products that adversely affect the country's economy. Machine learning based detection methods for such diseases can reduce these effects. However, these methods involve plant images and usually suffer the curse of dimensionality in terms of features. Features selection improves the performance of machine learning algorithms. Feature selection method aims to find optimal number of features for a machine learning algorithm such as classification, clustering and many more. This work proposed a Modified Sand Cat Swarm Optimization (MSCSO) algorithm based feature selection to find the optimal number of features. The algorithm performs well over other standard algorithm such as GA, PSO, and GWO. The performance of different classifiers are analysed for classification of diseases in tomato leaves based on features selected by MSCSO.

KEYWORDS

Feature Selection, Modified Sand Cat Swarm Optimization (MSCSO), Naïve Bayes (NB), K Nearest Neighbour (KNN), Support Vector Machine (SVM), Random Forest (RF).

1. INTRODUCTION

In India, most people depend on agriculture as a primary source of income. Plant diseases have significant effect over the economy and ecology of the crops which decreases the quality and magnitude of the crops drastically. According to statistics from the United Nations Food and Agriculture Organization (FAO), around 40% of global crop production is impacted due to plant diseases, resulting in $ 220Bn loss in the economy worldwide. Tomato is one of the leading vegetable species with overall contribution of 16% in total crop production globally. Plant diseases are observable in the leaf. Therefore, it is adequate to detect plant diseases early using images of leaves. Several widespread tomato plant diseases are Early Blight (EL), Late Blight (LB), Leaf Mold (LM), Bacterial Spot (BS), Mosaic Virus (MV), Septoria (S), Spider Mite (SM), Target Spot (TS) and Yellow Leaf (YL). Some of the critical issues and challenges during leaf disease analysis are as follows:

- Requirement of high-quality images to detect the image features for ensuing predictions
- Requirement of balanced dataset to overcome Class imbalance problem
- Efficient methods for selection of relevant features

Machine learning techniques are widely being utilised in agriculture for the detection and classification of plant diseases. Several methods have been proposed in the past to address these problems. The deep learning methods have high accuracy but the computational cost is very high. On the other hand conventional machine learning approaches suffer from dimensionality curse. Although there have been different attempts in reducing the dimensionality during feature selection, however there is still the scope of dimensionality reduction during feature selection. In this work, we present the following contributions:

- Use of image processing techniques to increase the quality of leaf images and k-means clustering for image segmentation.
- Utilize synthetic minority oversampling technique (SMOTE) to address class imbalance issue.
- Proposed Modified Sand Cat Swarm Optimization (MSCSO) for Feature Selection.
- Performance Analysis of MSCSO based feature selection.

The rest of the paper is organized as follows. Section 2 presents the related concepts. The review of literature is described in section 3. Section 4 explains the proposed methodology. The experimental results are presented and discussed in section 5. Section 6 concludes the proposed work.

2. PRELIMINARY

This section provides the brief overview of tomato plant disease, the feature details with respect to the images, and fundamental Sand Cat Swarm Optimization (SCSO) along with the standard classifiers like Naïve Bayes, KNN, SVM and Random Forest.

2.1. Tomato Leaf Diseases

India is the second largest producer of tomatoes in the world. Diseases in plants can be detected early by observing leaves. Fungi, bacteria, viruses, etc., are disease-causing agents responsible for various diseases in various plants. Understanding plant diseases is vital for ensuring farming productivity, maintaining a healthy ecosystem, and promoting sustainable agricultural practices. Some of the most common tomato leaves diseases are as follows:

- Bacterial spot: Bacterial spot on tomato leaves is caused by Xanthomonas vesicatoria. At the initial stage, the leaf has small dark lesions surrounded by a yellow halo. The spots grow on the leaf boundary and tip, and eventually, foilage becomes yellow and finally dies.
- Early Blight: Early blight tomato disease is caused by the fungus Alternaria solani. It occurs as significant, irregular areas with yellow halos on tomato leaves. Eventually, the spots widen into dark brown with concentric black rings.
- Late Blight: The fungus, Phytophthora infestans, induces late blight in tomatoes. The initial symptom of late blight is water soaked spots on older leaves. Leaves have large, dark brown spots with greyish edges that change to large areas of dry brown leaves.

- Leaf Mold: The fungus Passalora fulva causes leaf mold disease in tomato plants. It infects the oldest leaves first. The upper parts of the leaf contain light greenish-yellow spots. Spots on the leaf expand together and turn brown.

- Mosaic Virus: Tomato Mosaic Virus is a viral type spread by infected seeds or insects. It significantly damages tomato plants' leaves, stems, and fruits. The foliage of infected tomato plants displays mottling, with yellow and green parts. The leaves tend to be stunted and twisted with pointed tips.

- Septoria Leaf Spot: A fungus, Septoria lycopersici, causes septoria leaf spot disease in tomato plants. This disease is severe in prolonged humid and wet weather. Spots generally occur on the lower leaves of the plant after fruits set. The spots are small and circular or irregular in shape with grey centres.

- Spider Mite: Tomato red spider mite (Tetranychus evansi) is a tiny mite species. It sucks cell contents from the leaf that shows white or yellow dots on the leaf. Severely infested leaves can become yellow and fall off.

- Target Spot: The target spot (Corynespora cassicola) in tomato plants is one of the most significant diseases in tropical and subtropical areas. Its symptoms can be matched with bacterial spot and early blight. Plants are most sensitive to disease as seedlings and during fruiting. Initially, pinpoint-sized and water-soaked spots appear on the upper leaf part. The spots grow into little, necrotic lesions with pale brown centres and dark margins.

- Yellow leaf: Viruses in the Geminivirus family are the primary source of Tomato yellow leaf curl disease. It is carried in infected host plants. Whiteflies spread this disease. The most prominent symptom is that the leaves of the tomato plants are small and yellow between the veins. The leaflets also curl upwards and inwards. Plants become stunted.

2.2. Image Features

In image-based classification, an image feature is a component of information about the content of an image. Feature extraction is a critical step in classification that involves identifying and representing distinctive features within an image. Some of the common image features are colour, texture, and shape features.

2.2.1. Colour Features

The Colour features are most significant in the application of computer vision. They are strong descriptors often simplifying the detection and extraction of objects from an image. A colour space is a model representing as many colours as the human vision system can recognize. In this work, five colour spaces namely, RGB, LAB, HSV, HLS and LUV are defined. Each colour space is divided into its corresponding components. The extraction of colour features is represented by computing mean (m), standard deviation (sd), kurtosis (kt) and skewness (sk) of individual components of each colour space. Suppose the size of the image is $M \times N$ and P_{xy} is the pixel value at (x,y).

Mean (m) gives the average value of the component in a colour space. Standard deviation defines the variation between pixel value and mean value of a colour component. Skewness measures the asymmetric distribution of the intensity levels about the mean. Skewness is positive or negative.

The positive value shows that the many intensity values are on the left of the mean, while the negative value shows that many intensity values lie on the right. The zero skewness value indicates that the distribution of intensities is approximately equal on both sides of the mean. Kurtosis gives the shape of the distribution of intensities of the component of the colour space. The high kurtosis value specifies that the distribution has a sharp peak and a long and fat tail, while the low kurtosis value specifies that the distribution has a rounded peak and a shorter and thinner tail. Mean, standard deviation, kurtosis and skewness are defined using equations (1), (2), (3) and (4).

$$mean(m) = \frac{1}{\mathrm{M\ x\ N}} \sum_{x=1}^{M} \sum_{y=1}^{N} P_{xy} \qquad (1)$$

$$\text{standard Deviation (sd)} = \sqrt{\frac{1}{\mathrm{M\ x\ N}} \sum_{x=1}^{M} \sum_{y=1}^{N} (P_{xy} - m)} \qquad (2)$$

$$\text{kurtosis (kt)} = \frac{1}{\mathrm{M\ x\ N\ x\ sd^4}} \sum_{x=1}^{M} \sum_{y=1}^{N} (P_{xy} - m)\left(P_{xy} - m\right)^4 - 3 \qquad (3)$$

$$\text{skewness (sk)} = \frac{1}{\mathrm{M\ x\ N\ x\ sd^3}} \sum_{x=1}^{M} \sum_{y=1}^{N} (P_{xy} - m)^3 \qquad (4)$$

2.2.2. Shape Features

The shape of an object in the image is characterized using Zernike moments that are powerful and accurate descriptors. Zernike moments hold orthogonality and rotational invariant characteristics, providing no redundant information between moments. The computation of Zernike moments requires two parameters: the radius of the disc used to cover the object's region and the degree of the polynomial defined for the area.

2.2.3. Texture Features

Texture features help to get image intensities at different pixel positions and compute the irregularity between pixels. They are computed from Gray Level Co-Occurrence Matrix (GLCM) and arrange the neighbouring pixels with the orientation of 0°, 45°, 90°, and 135°. GLCM analyses the occurrence of grey levels together within an image, especially considering neighbouring pixels. GLCM describes the spatial arrangement of textures by estimating co-occurrences in a distinctive pixel offset (usually 1 or 2) and particular directions such as horizontal, vertical, and diagonal. Each cell (i, j) in the final GLCM represents a value Pij, which is the summation of the number of times the pixel with value i appeared in the defined spatial relationship to a neighbouring pixel with value j in the given image. The number of grey levels N in the image decides the dimensions of the GLCM. Five types of texture feature statistics: Correlation, Energy, Homogeneity, Angular Second Moment, and Contrast are relevant for plant images.

Correlation determines the linear dependencies between the grey tones of the pixels in the images.

$$Correlation\ (cr) = \sum_{i=0}^{N-1}\sum_{j=0}^{N-1} \frac{(i-m_i)(j-m_j)}{\sqrt{\sigma i\ \sigma j}} P_{ij} \qquad (5)$$

where m_i and m_j are mean values.

Entropy measures the irregularity or disorder of an image. A large value of entropy indicates the non-uniform texture of an image.

$$Entropy(et) = -\sum_{i=0}^{N-1}\sum_{j=0}^{N-1} P_{ij} \log P_{ij} \qquad (6)$$

Homogeneity measure is also known as the inverse difference moment. The homogeneity value will be high when the image has the same pixel values.

$$Homogeneity(hg) = \sum_{i=0}^{N-1}\sum_{j=0}^{N-1} \frac{P_{ij}}{1} + (i-j)^2 \qquad (7)$$

Angular Second Moment is also called energy. Angular second moment computes the textural uniformity of the image. It has a maximum value equal to one.

$$Angular\ Second\ Moment(asm) = \sum_{i=0}^{N-1}\sum_{j=0}^{N-1} {P_{ij}}^2 \qquad (8)$$

Contrast computes the local variations in the given image. The contrast shows the difference between the values of the neighbouring set of pixels.

$$Contrast(ct) = \sum_{i=0}^{N-1}\sum_{j=0}^{N-1} P_{ij}(i-j)^2 \qquad (9)$$

2.3. Sand Cat Swarm Optimization (SCSO)

The Sand Cat Swarm Optimization (SCSO) [1] algorithm depicts the behaviour of sand cats in nature. Sand cat performs two main activities: foraging for the prey and attacking the prey. Sand cats have a unique characteristic that detects low-frequency noises. In an optimization problem, each sand cat represents a solution, indicated through an array (X_1, X_2, . . . , X_d), where d is the number of dimensions in the problem. Mathematical modeling of SCSO to search for a prey is given by the following equations:

$$r_G = S_M - \left(\frac{2 \times S_M \times iter_c}{iter_{Maxc} + iter_c}\right) \qquad (10)$$

$$R = 2 \times r_G \times rand(0,1) - r_G \qquad (11)$$

$$r = r_G \times rand(0,1) \qquad (12)$$

$$pos(t+1) = r.(pos_{bs}(t) - rand(0,1) \times pos_c(t) \qquad (13)$$

The value of r_G describes the general sensitivity range and decreases from 2 to 0 over the iteration, and the value of S_M is considered to be 2. Parameter R controls the balance between exploration and exploitation. Parameter r illustrates the sensitivity range of each cat and is used to avoid the local optima pitfall. Parameters $iter_c$ and $iter_{Max}$ indicate the current iteration and maximum number of iterations, respectively. Equation (13) presents the update of the position of each sand cat (search agent) in iteration $t+1$, which is based on its current position pos_c and the position of the best sand cat pos_{bs} in the iteration t.

Search for prey exhibits the exploration phase while attacking the prey demonstrates the exploitation phase. Equations (14) and (15) show the exploitation phase of SCSO mathematically.

$$pos_{rnd} = |rand(0,1).pos_b(t) - pos_c(t)| \qquad (14)$$

$$pos(t+1) = pos_b(t) - r.pos_{rnd}.cos\theta \qquad (15)$$

The best and current positions of sand cat are represented by pos_b and pos_c, respectively. Angle θ determines the direction of movement for hunting. The SCSO uses the Roulette Wheel selection algorithm to choose a random angle for each sand cat. Thus, the position update of each sand cat in the population during the exploration and exploitation phases is presented using equation (16).

$$pos(t+1) = \begin{cases} pos_b(t) - r.pos_{rnd}.cos\theta & |R| \le 1 \\ r\left(pos_{bs}(t) - rand(0,1) \times \text{pos}_c(t)\right) & |R| > 1 \end{cases} \qquad (16)$$

2.4. Naïve Bayes (NB) Classification Model

Naïve Bayes [2] is a classification model using Bayes' Theorem with a presumption of class conditional independence. In simple words, a Naive Bayes model presumes that each attribute in a class is conditionally independent of any other attribute. The Naive Bayes model uses Bayes' theorem in equation (17) to define the probability that a tuple X belongs to class C_j.

$$P(C_j|X) = \frac{P(X|C_j)P(C_j)}{P(X)} \qquad (17)$$

Tuple X belongs to the class for which the value $P(Cj \mid X)$ will be maximum. The value of $P(X \mid Cj)$ is computed according to the class conditional independence assumption in equation (18).

$$P(X|C_j) = \prod_{r=1}^{n} P(x_r|C_j) \qquad (18)$$

Where n is the number of attributes and x_k indicates the value of the k^{th} attribute in tuple X.

2.5. K Nearest Neighbour (KNN) Classification Model

KNN [3] is a supervised machine learning method. It is a lazy learner classifier because it simply stores training data points and predicts a class label for a given test data point by computing its

similarity with training data. KNN learns by comparing a test data point with training data close to it. The distance metric is used to calculate closeness. K is the number of training data points closest to the test data points. KNN has slow speed and storage issues for large dataset. Prediction results depend on the choice of K and distance metric.

2.6. Support Vector Machine (SVM) Classification Model

SVM [4] is a supervised machine learning algorithm used to solve classification and regression tasks. It tries to determine a hyperplane that maximizes the separation between data points depending on their defined classes. Support vectors are the data points closest to the hyperplane. SVM works for binary classification. It performs multi-class classification by dividing the problem into different binary classification problems.

2.7. Random Forest (Rf) Classification Model

Random Forest [5] is an ensemble algorithm for classification and regression problems. It uses a bagging ensemble technique that takes different sample training data and feature subsets and constructs different decision trees. The random forest output is based on majority voting by combining the outputs of constructed decision trees. Random Forest is a scalable and robust machine learning algorithm.

3. RELATED WORK

Bolón-Canedo et al. [6] classified feature selection methods into three categories: filters, wrappers and embedded. Filters based method retain relevant features and eliminate redundant and irrelevant features from the datasets. They do not use any machine learning algorithm to find relevant feature set. Thus the selected features may or may not be optimal for applied machine learning model [7]. Some examples of filter methods are correlation, variance thresholds, and mutual information. Wrapper methods depend on machine learning algorithms to assess the effectiveness of selected features. They produce better quality subsets of features. In the embedded method, feature selection is done during the training phase of the machine learning algorithm. Some machine learning algorithms, such as gradient boosting embeds feature selection as an integral part of their learning. Nature-inspired meta-heuristic algorithms are accepted in wrapper methods rather than embedded methods for searching for an optimal number of relevant features in a reasonable time [8].

Soleimanian and Mousavi [9] proposed a hybrid optimization algorithm using Particle Swarm Optimization (PSO) and Fruit Fly Optimization (FFO) for feature selection to reduce dimensionality in email spam classification. In [10], the authors presented the use of chaotic crow search and PSO for feature selection. The chaotic crow search optimization algorithm offers a better searching method, and PSO finds the global optimal feature set. Ansari et al. [11] applied the Genetic Algorithm (GA) for feature selection to classify natural scene texts. GA uses the f-score of the SVM classifier as the fitness function. Ewees et al. [12] use salp swarm and grasshopper optimization for feature selection. The salp swarm accelerates the search capability of grasshopper optimization by evolving the population by utilizing a crossover operator. In [13], the authors used GA and binary PSO for dimensionality reduction in the Parkinson's disease dataset. They compare the performance of different classifiers based on features selected by GA and binary PSO. In [14], the authors described the random grey wolf optimization algorithm for feature selection and demonstrated its use in selecting relevant features of various chronic diseases. In [15], the authors proposed a computer-aided diagnosis system to analyze mammogram images. They used grey wolf optimization with rough set theory to determine significant features from images.

Sarhani and Voß [16] used the concept of chunking and cooperative learning with PSO to address the problem of selecting relevant features. Sehgal et al. [17] presented a modified grasshopper optimization method for selecting significant features to analyze and identify Parkinson's disease. In [18], the authors described the use of deep learning to detect diseases in paddy leaves automatically. They used adaptive rain optimization and SVM for the feature selection phase. In [19], the authors described disease classification in grapes leaves using a support vector machine that uses an artificial bee colony (ABC) optimization algorithm to select the optimal number of image features to enhance the performance of the classifier. In [20], the ensembling of naive bayes and logistics algorithms with Particle Swarm Optimization is used for feature selection and classification of plant diseases. Reddy et al. [21] proposed a modified red deer optimization (RDO) algorithm to select an optimal number of features extracted by the ResNet50 deep learning model. The optimization algorithm enhances the classification accuracy and f-score of the deep learning model. Pham et al. [22] applied the Artificial Neural Network method to classify mango leaf diseases for early identification. They proposed Adaptive Particle Grey Wolf Optimization (APGWO) for feature selection to select the most valuable features. In [23], the Salp Swarm Optimization algorithm is utilized for feature selection from plant disease images. The optimization algorithm improves the performance of the KNN classification method in detecting diseases in plants. Jain and Dharavath [24] introduced an automated disease identification system for crop diseases such as maize, rice and grapes. They presented a memetic salp swarm optimization method to find the most relevant features with the best classification accuracy. In [25], an optimized algorithm called slime mould optimization with SVM is described to analyze apple tree diseases.

Khan et al. [26] suggested the selection of the most suitable features using a novel method of entropy and rank-based correlation. In [27], the authors combined KNN with ReliefFAttributeEval to select the right features for classifying and forecasting plant diseases. In [28], the feature selection method, the binary Greylag Goose Optimization, is used to enhance the performance of the machine learning models by determining feature sets appropriate to the models. Outcomes revealed that the multi-layer perceptron classifier, with feature selection, obtained an accuracy of 98.3%, highlighting the vital function of feature selection in enhancing machine learning model performance. Farhanah and Al Maki [29] explained the application of Binary PSO for feature selection to identify diseases in Hop plants. Ferdinand and Al Maki [30] presented the application of PSO with SVM for feature selection in identifying diseases in broccoli leaves. The summary of the research work discussed in this section is given in Table 1.

Table 1. Summary of Existing Feature Selection Research Work

Ref	Feature Selection Algorithm	Application Area	Classification Algorithm	Fitness function
[9]	PSO and Fruit Fly Optimization	Email spam classification	KNN, DT, MLP ANN,	Distance criterion
[10]	Chaotic crow search and particle swarm optimization algorithm	Different diseases such as Heart, Lung cancer, Parkinson	KNN	Classification error rate
[11]	Genetic Algorithm	Natural scene texts	SVM	f-score of classifier
[12]	Crossover-salp swarm with grasshopper optimization algorithm	Different datasets from UCI repository such as Wine, Breast cancer, etc.	KNN	Classification error rate

[13]	Genetic Algorithm and binary PSO	Parkinson disease	K Nearest Neighbour (KNN), SVM, ANN, Decision Tree (DT)	Classification accuracy
[14]	Random walk grey wolf optimizer	Chronic disease datasets from UCI repository	KNN	Classification error rate
Ref	**Feature Selection Algorithm**	**Application Area**	**Classification Algorithm**	**Fitness function**
[15]	Grey wolf optimization with rough set theory	Mammogram image analysis	NB, J48	Dependency measure
[16]	PSO	Disease datasets from UCI repository	SVM	Classification error rate and % of selected number of features
[17]	Grasshopper optimization algorithm	Parkinson Disease	KNN, DT, Random Forest	Classification error rate and % of selected number of features
[18]	Adaptive rain optimization	Paddy plant leaf disease	Adaptive bi-long short term memory	Classification accuracy
[19]	Artificial Bee Colony Optimization	Grapes leaves diseases	Support Vector Machine (SVM)	Classification accuracy
[20]	Particle Swarm Optimization (PSO)	Plant leaves diseases	Logistic Regression, Naive Bayes	Classification accuracy
[21]	Red deer optimization	Plant leaf disease	ResNet50	Distance
[22]	Adaptive Particle Grey Wolf Optimization	Mango leaves diseases	Artificial Neural Network (ANN)	Error rate
[23]	Salp Swarm Algorithm	Plant diseases	KNN	Classification error rate and % of selected number of features
[24]	Memetic salp swarm optimization	Crop diseases	KNN, SVM	Classification accuracy
[25]	Slime mould optimization	Apple tree diseases	SVM	Classification accuracy
[26]	Entropy and rank based correlation	Fruit diseases	SVM	Entropy-correlation
[27]	ReliefF Algorithm	Plant Disease	KNN	Classification accuracy
[28]	binary Greylag Goose Optimization	Potato leaf disease	KNN, SVM, Logistic Regression	Classification accuracy
[29]	Binary PSO	Hop Plant Disease	SVM	Classification accuracy
[30]	PSO	Broccoli leaf disease	SVM	Classification accuracy

It has been observed during the literature review that several meta-heuristic approaches have been proposed for the feature selection. However there is still scope of improvisation and therefore we have explored the SCSO which is originally proposed for the continuous domain and modified it to implement over discrete feature selection domain. One of the major issues with the proposed approaches is the number of parameters that are to be tuned to obtain the optimal results. In comparison to these approaches MSCSO has only one parameter i.e. sensitivity range to be tuned.

We have also explored the reduction in the number of features in comparison to other approaches.

4. PROPOSED METHODOLOGY

Figure 1 illustrates the overview of the research work presented in this paper. Features are extracted from preprocessed tomato leave images. The proposed nature-inspired optimization algorithm Modified Sand Cat Swarm Optimization (MSCSO) selects the optimal number of features. Different classification models such as KNN, Naïve Bayes, SVM, Random Forest are used for classification of diseases in tomato leaves using selected features. The effectiveness of these classification models is measured using performance parameters such as accuracy, precision, recall and f1-score.

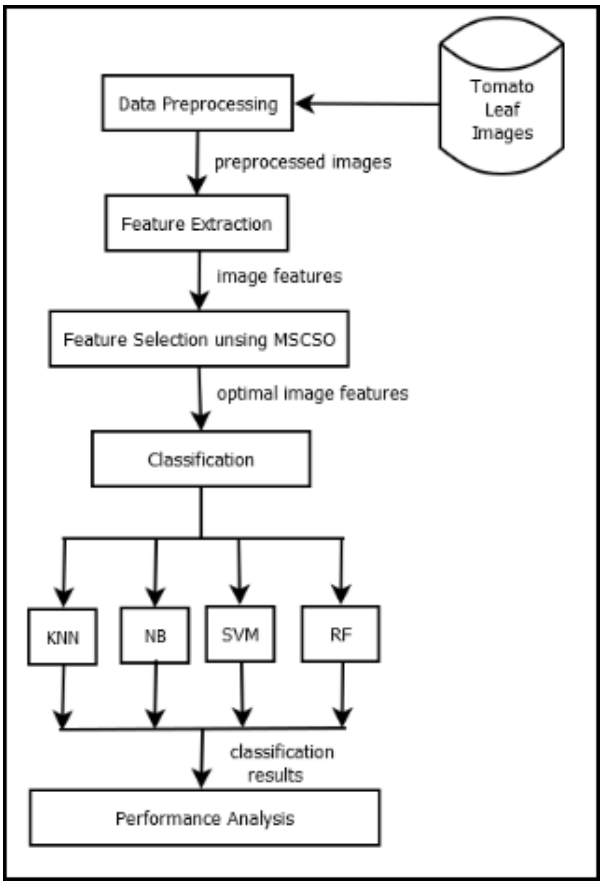

Figure 1. The Proposed Methodology

4.1. Dataset and Data Preprocessing

Plant village database [31] of tomato leaf is considered. We consider 10 categories of tomato leaves that contain 9 diseases and 1 healthy. Figure 2 shows samples of disease categories. Each leaf image is denoised using fast NL-Means method [32] and Size of the image is 256 × 256. The k-means clustering [33] is a simple and computationally efficient algorithm, so it is employed on

tomato leaf images to determine the region of interest. The preprocessing steps of bacterial spot disease are shown in Figure 3.

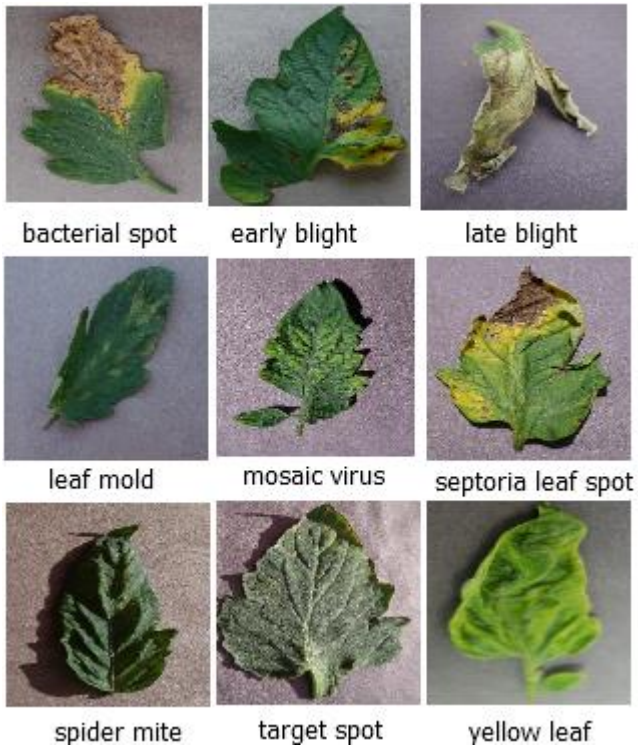

Figure 2. Sample Images of infected tomato leaves

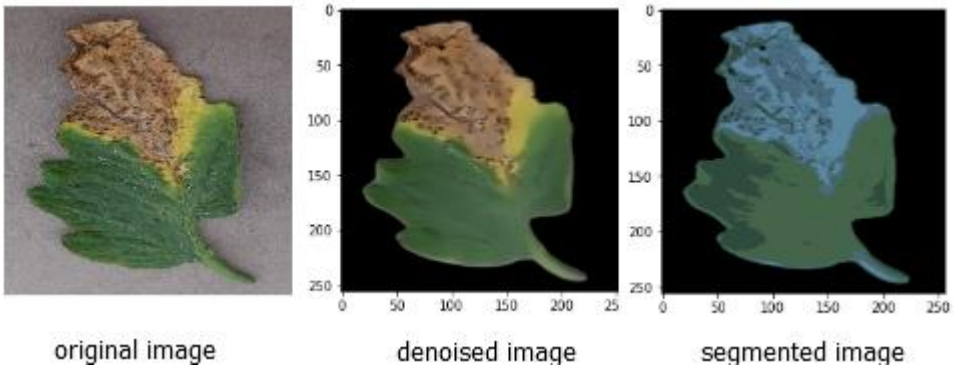

Figure 3. Preprocessing of Bacterial Spot Disease Image

4.2. Feature Extraction

Feature extraction is used to characterize the inputs for a classifier. We consider colour features, Zernike moments, texture features of an image. We consider five colour spaces namely LAB, RGB, HLS, HSV and LUV. These colour features are useful in classifying the infected area of an image. The colour features are represented by computing mean (m), standard deviation (SD), kurtosis (KT) and skewness (SK) for each colour space [34]. The total number of colour features is 60.

Zernike moments are shape descriptor of an image. This shape descriptor holds orthogonality and rotational invariant characteristics. Zernike moments [35] are more robust due to orthogonality as it provides non redundant information between moments. There are 25 values for zernike moments. Texture features help to get image intensities at different pixel positions and compute the irregularity between pixels. They are computed from Gray Level Co-Occurrence Matrix (GLCM) and arranges the neighbouring pixels with the orientation of 0°, 45°, 90°, 135°. Five types of texture feature statistics: Contrast, Correlation, Energy, Homogeneity, and Angular Second Moment are relevant for plant images [36]. The total numbers of texture features are 20 for four orientations of pixels.

4.3. Feature Selection Using Modified Sand Cat Swarm Optimization (MSCSO)

Feature selection method enhances the quality of the features in many application areas such as classification, clustering, regression, etc. Many features are needed in machine learning to solve real-time problems. However, only few features are essential for finding the solution, because some are irrelevant and redundant, and affects the accuracy of overall system performance. Nature-inspired algorithms come under meta-heuristic approaches that are generic and solve a variety of optimization problems. Nature-inspired algorithms cover a wide range of applications and find optimal solutions in polynomial time. The feature selection procedure aims to find the optimal number of features representing the original feature set and yield results with high classification accuracy. Selection of set of relevant features is a combinatorial problem and is considered as a global optimization problem. Brute force approach takes exponential time to find optimal solution. Nature-inspired algorithms give optimum results in polynomial time.

Sand Cat Swarm Optimization is a new and efficient swarm intelligent algorithm. It works well for problems having continuous search space. However, some problems have discrete search space. For example, in selecting an optimal number of features, each solution vector represents discrete values 1 and 0, where 1 represents the inclusion of the feature and 0 means that the feature is not considered.

4.3.1. Solution Encoding and Initial Population

Each sand cat represents the solution to the feature selection problem, shown by an array of 1 × d. Figure 4 shows the solution encoding for i^{th} sand cat. The value of d gives the total number of features of the image dataset. Each value in the array is either 1 or 0. The initial population is generated randomly.

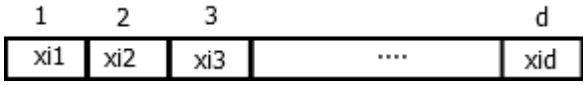

Figure 4. Solution Encoding of i^{th} Sand Cat

4.3.2. Fitness Measure

The fitness function measures the quality of solutions in the population. In this work, the fitness is computed on the basis of classification accuracy, and number of selected features values of Naïve Bayes classifier. Naïve Bayes classifier is simple and effective and takes linear time in training and testing. The classification accuracy and fitness are computed using following equations:

$$Overall\_accuracy = \frac{\sum True\ Positives\ for\ each\ class}{\text{Total samples}} \quad (19)$$

$$fitness = w1.Overall\_accuracy + \text{w2}.\frac{|d-m|}{|d|} \quad (20)$$

The sum of weights must be equal to 1. We have assigned the weightage of 0.85 to the accuracy and 0.15 to the impact of reduced features. The weights are assigned to check if the same accuracy can be obtained from fewer features. Accordingly the fitness function is defined in terms of weighted sum. Value *m* represents the size of the selected feature subset, and *d* is the total number of features. The value $|d - m| / |d|$ emphasizes considering solutions with high accuracy, and fewer features.

4.3.3. Exploring and Exploiting the Prey

Sand cat performs search for the prey. In optimization problem, searching shows exploration for the optimal solution. Searching behaviour of sand cat in Modified Sand Cat Swarm Optimization is given mathematically as follows:

$$pos(t+1) = pos_b(t)\ crossover\ \left(mutation\ pos_c(t)\right) \quad |R| > 1 \quad (21)$$

The values of *R* are calculated using Equations (1) and (2). Two point crossover operator is used. The mutation and crossover operator support diversification of the search directions and avoid early convergence to local optima. In this work, a dynamic multipoint mutation operator is used. At each of the iteration, numbers of mutation points are chosen randomly. The sensitivity range of a sand cat is considered a circle. Thus, random angle θ determines the direction of movement. The value of $cos\theta$ is between -1 and 1. The following equations describe the exploitation phase in Modified Sand Cat Swarm Optimization.

$$pos_{rnd} = (mutation\ pos_{bs}(t))crossover\ (pos_c(t)) \quad (22)$$

$$pos(t+1) = \begin{cases} pos_b(t)crossover\ (mutation\ pos_{rnd}) & cos\theta \in [-1,0)\ and\ |R| \le 1 \\ pos_b(t)crossover\ pos_{rnd} & cos\theta \in (0,1]\ and\ |R| \le 1 \\ pos_b(t) & cos\theta = 0\ and\ |R| \le 1 \end{cases} \quad (23)$$

The algorithm of MSCSO is given in Algorithm 1.

Algorithm 1: FS_MSCSO
Input candidate feature sets
Output optimal feature set
1. Initialize population P of candidate feature sets (sand cats) randomly using binary encoding.
2. Calculate the fitness of each sand cat using Equation (20).
3. Initialize $S_M = 2$, $iter_c = 1$, $iter_{Max}$
4. **while** $iter_c <= iter_{Max}$ **do**
5. Compute r_G and R using Equation (10) and Equation (11).
6. **for** each sand cat **do**
7. Get a random angle θ ($0^o \le \theta \le 360^o$) using roulette wheel selection.
8. **if** |R| <= 1 **then**
9. Update the position of sand cat using Equation (23).

```
10.     else
11.         Update the position of sand cat using Equation (21).
12.     Find best feature set (sand cat) based on fitness.
13.     iter_c = iter_c + 1
```

Time complexity of proposed algorithm is $O\ (iter_{Max} \times n_1 \times d \times n_2)$ where n_1 is population size, d is number of features, and n_2 is training size for Naïve Bayes classifier. Thus the proposed algorithm finds optimal feature set in polynomial time.

5. EXPERIMENTAL RESULTS

The experiments are performed using Python programming environment on a workstation with Windows 10 platform, 16GB of RAM and Intel Core i7 CPU @ 2.3 GHz.

5.1. Training Set and Testing Set

The preprocessed tomato image dataset was divided into two parts: training set (70%) and test set (30%). 70% of the dataset is used as the training set for classification model training. The large size of the training set helps the classification model to analyze the data accurately and provide precise predictions. The performance is evaluated using the test set. The size of the training set is 7000. The size of the test set is 3000. Each category of leaf image in the dataset contains a different number of images that create class imbalance and affect the performance of the classification model. This work uses the Synthetic Minority Oversampling Technique (SMOTE) [37] to equalize the number of samples in all classes to solve the class imbalance problem. Table 2 presents the number of images in each class in the dataset.

Table 2. Dataset Description

Class	Images in Dataset	Images after SMOTE	Training Images	Testing Images
Healthy	750	1000	700	300
Bacterial spot	850	1000	700	300
Early blight	600	1000	700	300
Late blight	764	1000	700	300
Leaf mold	550	1000	700	300
Mosaic virus	250	1000	700	300
Septoria leaf spot	708	1000	700	300
Spider mite	670	1000	700	300
Target spot	650	1000	700	300
Yellow leaf	1000	1000	700	300

5.2. Comparative Analysis of MSCSO with Standard Optimization Algorithms

The proposed nature-inspired algorithm Modified Sand Cat Swarm Optimization (MSCSO) is compared with Genetic Algorithm (GA), Particle Swarm Optimization (PSO) and Grey Wolf

Optimization (GWO) to show its effectiveness. The population size and max iteration are 200 and 40, respectively. Other parameter values are shown in Table 3.

Table 3. Parameters for optimization algorithms

Optimization Algorithm	**Parameter**	**Value**
GA	Crossover rate	0.85
	Mutation rate	0.15
PSO	Inertia weight	[0.9, 0.4]
	Coefficient C1	1
	Coefficient C2	2
GWO	A	[2, 0]
	A	[-2a, 2a]
	C	[0, 2]
MSCSO	Sensitivity range (r_G)	[2, 0]

The results of different optimization algorithms using Naïve Bayes classification model are presented in Table 4.

Table 4. Performance results of different optimization algorithms

Measure	**GA**	**PSO**	**GWO**	**MSCSO**
Optimal features	61	55	49	44
Overall Accuracy of Naïve Bayes (%)	78.20	82.17	86.10	89.23

The optimization algorithms are stochastic in nature, and require undergoing statistical tests to confirm the results. Wilcoxon signed rank test is a statistical test to validate the significant results obtained by applying Naïve Bayes classifier over the results obtained from GA, PSO, MGWO and MSCSO for 20 executions independently. The significance level, $\alpha = 0.05$ is kept uniform for each of the approaches during the statistical test. Tables 5 exhibit the results for the independent executions considering optimal features and overall accuracy for Naïve Bayes classifier. The results of MSCSO are compared with GA, PSO, and GWO. R+ represents the cumulative rank for all the independent executions in which MSCSO performs better and R- represents the cumulative ranks in which compared method performs better than MSCSO. It can be observed from the Table 5 that MSCSO has superior performance with respect to both overall accuracy and optimal features. Thus MSCSO is claimed to have better performance than GWO, PSO and GA.

Table 5: Wilcoxon signed rank test results MSCSO with GA, PSO, GWO

Comparison	**Optimal Features**			**Overall Accuracy of NB**		
	R+	**R-**	**p value**	**R+**	**R-**	**p value**
MSCSO vs. GA	210	0	0.8e-04	210	0	0.8e-04

MSCSO vs. PSO	210	0	0.8e-04	210	0	0.8e-04
MSCSO vs. GWO	209	1	0.1e-03	209	1	0.1e-03

5.3. Comparative Analysis of Performance of Classification Models with MSCSO

The performance measures of a classification model for a dataset are computed using a confusion matrix of N x N, where N is the number of classes in the dataset. The confusion matrix compares the actual class values against the values predicted by the model. It presents a holistic view of the performance of the model. The confusion matrix gives true positives (tp), false positives (fp), true negatives (tn), and false negatives (fn) for each class in the dataset. Figure 5 shows the confusion matrix generated for the random forest classification model with features selected by MSCSO.

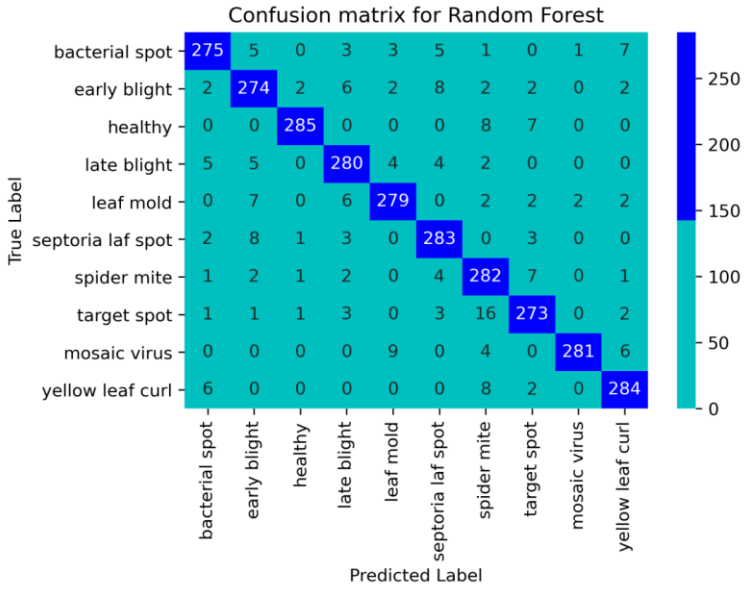

Figure 5. Confusion matrix for random forest with MSCSO

Table 6 specifies the values of different performance measures concerning each class of tomato leaf data set for the random forest classification model.

Table 6. Class-wise Performance results of Random Forest with MSCSO

Class	Accuracy (%)	Precision	Recall	F-Score
Bacterial Spot	98.60	0.9418	0.9167	0.9291
Early Blight	98.20	0.9073	0.9133	0.9103
Healthy	99.33	0.9828	0.9500	0.9661
Late Blight	98.57	0.9241	0.9333	0.9287
Leaf Mold	98.70	0.9394	0.9300	0.9347
Septoria Leaf Spot	98.63	0.9218	0.9433	0.9325
Spider Mite	97.97	0.8677	0.9400	0.9024
Target Spot	98.33	0.9223	0.9100	0.9161
Mosaic Virus	99.27	0.9894	0.9367	0.9623
Yellow Leaf Curl	98.80	0.9342	0.9467	0.9404

It can be observed from the Table 6 that the Random Forest performs well with the accuracy above 98% for each of the disease class. The results are also validated through a good f-score for each class. Table 7 represents the overall performance of different classifiers applied to the

features selected from MSCSO and it is evident from the results that the Random Forest outperforms the other classifiers.

Table 7. Performance results of different classification models with MSCSO

Classification Model	Overall Accuracy(%)	Precision	Recall	F-Score
KNN	87.50	0.8754	0.8750	0.8752
NB	89.23	0.8932	0.8923	0.8927
SVM	91.20	0.9124	0.9120	0.9122
RF	93.20	0.9330	0.9320	0.9325

The results are validated by considering the primary measures like Accuracy, precision, recall and f-score. It can be observed that all these measures are well balanced and thus strongly support the performance of the MSCSO.

6. CONCLUSION

In this work, we have proposed feature selection algorithm and studied its impact in the performance of machine learning algorithms in detection of tomato leaf diseases. We used the PlantVillage dataset and considered nine different tomato leaf diseases. The image quality is increased using denoising, and k-means clustering performs image segmentation to find the region of interest. Synthetic Minority Oversampling Technique (SMOTE) handles class imbalance issues. Selecting an optimal number of features is vital for the excellent performance of machine learning-based classifiers. Thus, Modified Sand Cat Swarm Optimization (MSCSO) is proposed to select features from images. MSCSO improves the performance of KNN, Naïve Bayes, SVM and Random Forest models. Machine learning models are compared using performance measures such as accuracy, precision, recall, and f-score.

REFERENCES

[1] A. Seyyedabbasi and F. Kiani, (2023) “Sand cat swarm optimization: A nature-inspired algorithm to solve global optimization problems,” Engineering with Computers, Vol. 39, No. 4, pp. 2627–2651.

[2] I. Rish, (2001) “An empirical study of the naive bayes classifier,” in IJCAI 2001 workshop on empirical methods in artificial intelligence, Vol. 3, No. 22. Citeseer, pp. 41–46.

[3] E. Fix, (1985) Discriminatory analysis: nonparametric discrimination, consistency properties. USAF school of Aviation Medicine, Vol. 1.

[4] V. Vapnik, S. Golowich, and A. Smola,(1996) “Support vector method for function approximation, regression estimation and signal processing,” Advances in neural information processing systems, Vol. 9.

[5] L. Breiman, (2001) “Random forests,” Machine learning, Vol. 45, pp. 5–32.

[6] V. Bol´on-Canedo, N. S´anchez-Maro˜no, and A. Alonso-Betanzos, (2016) “Feature selection for high-dimensional data,” Progress in Artificial Intelligence, Vol. 5, pp. 65–75.

[7] J. Li, K. Cheng, S. Wang, F. Morstatter, R. P. Trevino, J. Tang, and H. Liu, (2017) “Feature selection: A data perspective,” ACM computing surveys (CSUR), Vol. 50, No. 6, pp. 1–45.

[8] M. S. Raza and U. Qamar, (2017) Understanding and using rough set based feature selection: concepts, techniques and applications. Springer.

[9] F. Soleimanian Gharehchopogh and S. K. Mousavi, (2020) “A new feature selection in email spam detection by particle swarm optimization and fruit fly optimization algorithms,” Computer and Knowledge Engineering, Vol. 2, No. 2, pp. 49–62.

[10] A. Adamu, M. Abdullahi, S. B. Junaidu, and I. H. Hassan, (2021) “An hybrid particle swarm optimization with crow search algorithm for feature selection,” Machine Learning with Applications, Vol. 6, p. 100108.

[11] G. J. Ansari, J. H. Shah, M. C. Farias, M. Sharif, N. Qadeer, and H. U. Khan, (2021) "An optimized feature selection technique in diversified natural scene text for classification using genetic algorithm," IEEE Access, Vol. 9, pp. 54 923–54 937.
[12] A. A. Ewees, M. A. Gaheen, Z. M. Yaseen, and R. M. Ghoniem, (2022) "Grasshopper optimization algorithm with crossover operators for feature selection and solving engineering problems," IEEE Access, Vol. 10, pp. 23 304–23 320.
[13] A. Pasha and P. H. Latha, (2020) "Bio-inspired dimensionality reduction for parkinson's disease (pd) classification," Health information science and systems, Vol. 8, No. 1, p. 13.
[14] Preeti and K. Deep, (2022) "A random walk grey wolf optimizer based on dispersion factor for feature selection on chronic disease prediction," Expert Systems with Applications, Vol. 206, p. 117864.
[15] A. Sathiyabhama, S. U. Kumar, J. Jayanthi, T. Sathiya, A. Ilavarasi, V. Yuvarajan, and K. Gopikrishna, (2021) "A novel feature selection framework based on grey wolf optimizer for mammogram image analysis," Neural Computing and Applications, Vol. 33, No. 21, pp. 14 583– 14 602.
[16] M. Sarhani and S. Voß, (2022) "Chunking and cooperation in particle swarm optimization for feature selection," Annals of Mathematics and Artificial Intelligence, Vol. 90, No. 7, pp. 893–913, 2022.
[17] S. Sehgal, M. Agarwal, D. Gupta, S. Sundaram, and A. Bashambu, (2020) "Optimized grass hopper algorithm for diagnosis of parkinson's disease," SN Applied Sciences, Vol. 2, pp. 1–18.
[18] R. K. Dubey and D. K. Choubey, (2024) "An efficient adaptive feature selection with deep learning model-based paddy plant leaf disease classification," Multimedia Tools and Applications, Vol. 83, No. 8, pp. 22 639–22 661.
[19] A. D. Andrushia and A. T. Patricia, (2020) "Artificial bee colony optimization (abc) for grape leaves disease detection," Evolving Systems, Vol. 11, No. 1, pp. 105–117.
[20] Chaudhary, R. Thakur, S. Kolhe, and R. Kamal, (2020) "A particle swarm optimization based ensemble for vegetable crop disease recognition," Computers and Electronics in Agriculture, Vol. 178, p. 105747.
[21] S. R. Reddy, G. S. Varma, and R. L. Davuluri, (2023) "Resnet-based modified red deer optimization with dlcnn classifier for plant disease identification and classification," Computers and Electrical Engineering, Vol. 105, p. 108492.
[22] T. N. Pham, L. Van Tran, and S. V. T. Dao, (2020) "Early disease classification of mango leaves using feed-forward neural network and hybrid metaheuristic feature selection," IEEE access, Vol. 8, pp. 189 960–189 973.
[23] X. Xie, F. Xia, Y. Wu, S. Liu, K. Yan, H. Xu, and Z. Ji, (2023) "A novel feature selection strategy based on salp swarm algorithm for plant disease detection," Plant Phenomics, Vol. 5, p. 0039.
[24] S. Jain and R. Dharavath, (2023)"Memetic salp swarm optimization algorithm based feature selection approach for crop disease detection system," Journal of Ambient Intelligence and Humanized Computing, Vol. 14, No. 3, pp. 1817–1835.
[25] S. M. Javidan, A. Banakar, K. A. Vakilian, and Y. Ampatzidis, (2022)"A feature selection method using slime mould optimization algorithm in order to diagnose plant leaf diseases," in 8th Iranian Conference on Signal Processing and Intelligent Systems (ICSPIS). IEEE, pp. 1–5.
[26] M. A. Khan, T. Akram, M. Sharif, M. Alhaisoni, T. Saba, and N. Nawaz, (2021)"A probabilistic segmentation and entropy-rank correlation based feature selection approach for the recognition of fruit diseases," EURASIP Journal on Image and Video Processing, Vol. 2021, No. 1, pp. 14.
[27] I. Imran, R. H. Ali, S. M. Jameel, and R. A. Jaleel, (2024) "A proposed model based on k-nearest neighbour classifier with feature selection techniques to control and forecast plant disease," International Journal of Grid and Utility Computing, Vol. 15, No. 3-4, pp. 306–313.
[28] M. Radwan, A. A. Alhussan, A. Ibrahim, and S. M. Tawfeek, (2024) "Potato leaf disease classification using optimized machine learning models and feature selection techniques," Potato Research, pp. 1–25.
[29] A. Farhanah and W. F. Al Maki, (2022) "Hops plants disease detection using feature selection based bpso-svm," in 9th International Conference on Electrical Engineering, Computer Science and Informatics(EECSI). IEEE, pp. 389–393.

[30] Y. Ferdinand and W. F. Al Maki, (2022) "Broccoli leaf diseases classification using support vector machine with particle swarm optimization based on feature selection," International Journal of Advances in Intelligent Informatics, Vol. 8, No. 3, pp. 337–348.

[31] A. Hughes, M. Salath´e, (2015) "An open access repository of images on plant health to enable the development of mobile disease diagnostics," arXiv preprint arXiv:1511.08060.

[32] A. Buades, B. Coll, and J.-M. Morel, (2005) "Image denoising by nonlocal averaging," in Proceedings.(ICASSP'05). IEEE International Conference on Acoustics, Speech, and Signal Processing, Vol. 2. IEEE, pp. ii–25.

[33] Macqueen, (1967) "Some methods for classification and analysis of multivariate observations," in Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability/University of California Press.

[34] W. K. Mutlag, S. K. Ali, Z. M. Aydam, and B. H. Taher, (2020) "Feature extraction methods: a review," in Journal of Physics: Conference Series, Vol. 1591, No. 1. IOP Publishing, pp. 012028.

[35] M. R. Teague, (1980) "Image analysis via the general theory of moments," Journal of Optical Society of America, Vol. 70, No. 8, pp. 920–930.

[36] A. Kadir, (2014) "A model of plant identification system using glcm, lacunarity and shen features," arXiv preprint arXiv:1410.0969.

[37] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, (2022) "Smote: synthetic minority over-sampling technique," Journal of artificial intelligence research, Vol. 16, pp. 321–357.

AUTHORS

Chetan Singh Negi is working as Associate Professor in the College of Technology, GBPUAT, Pantnagar. He has completed his B. Tech. from UPTU, and M. Tech. from IIT Kharagpur. He is currently pursuing his Ph.D. from College of Technology, Pantnagar.

H. L. Mandoria is working as Professor in the College of Technology, GBPUAT, Pantnagar. He has done his B.Tech. from Pantnagar and Master's & PhD from MANIT, Bhopal. Currently he is on deputation as Director, APKKIT, Tanakpur.

Sunita Jalal is working as Associate Professor in the College of Technology, GBPUAT, Pantnagar. She has completed her B. Tech. from UPTU, M. Tech. from Banasthali Vidyapeeth and Ph.D. from MNNIT, Allahabad.