AIMS Mathematics, 9(5): 11432–11451.
DOI: 10.3934/math.2024561
Received: 15 December 2023
Revised: 21 February 2024
Accepted: 01 March 2024
Published: 25 March 2024

http://www.aimspress.com/journal/Math

Research article

A battle royale optimization with feature fusion-based automated fruit disease grading and classification

S. Rama Sree[1], E Laxmi Lydia[2], C. S. S. Anupama[3], Ramya Nemani[4], Soojeong Lee[5], Gyanendra Prasad Joshi[5,*] and Woong Cho[6,*]

[1] Department of CSE, Aditya Engineering College, Surampalem, India
[2] Department of Computer Science and Engineering, Vignan's Institute of Information Technology, Visakhapatnam, 530049, India
[3] Department of Electronics and Instrumentation Engineering, V.R.Siddhartha Engineering College, Vijayawada 520007, India
[4] GITAM School of Sciences, Visakhapatnam Campus, GITAM (Deemed to be University), Andhra Pradesh, India
[5] Department of Computer Science and Engineering, Sejong University, Seoul 05006, Republic of Korea
[6] Department of Electronics, Information and Communication Engineering, Kangwon National University, Samcheok 25913, Gangwon State, Republic of Korea

* **Correspondence:** Email: joshi@sejong.ac.kr; wcho@kangwon.ac.kr; Tel: +82-2-6935-2481; +82-33-570-6402.

Abstract: Fruit Disease Detection (FDD) using Computer Vision (CV) techniques is a powerful strategy to accomplish precision agriculture. Because, these techniques assist the farmers in identifying and treating the diseased fruits before it spreads to other plants, thus resulting in better crop yield and quality. Further, it also helps in reducing the usage of pesticides and other chemicals so that the farmers can streamline their efforts with high accuracy and avoid unwanted treatments. FDD and Deep Learning (DL)-based classification involve the deployment of Artificial Intelligence (AI), mainly the DL approach, to identify and classify different types of diseases that affect the fruit crops. The DL approach, especially the Convolutional Neural Network (CNN), has been trained to classify the fruit images as diseased or healthy, based on the presence or absence of the disease symptoms. In this background, the current study developed a new Battle Royale Optimization with a Feature Fusion Based Fruit Disease Grading and Classification (BROFF-FDGC) technique. In the presented BROFF-

FDGC technique, the Bilateral Filtering (BF) approach is primarily employed for the noise removal process. Besides, a fusion of DL models, namely Inception v3, NASNet, and Xception models, is used for the feature extraction process with Bayesian Optimization (BO) algorithm as a hyperparameter optimizer. Moreover, the BROFF-FDGC technique employed the Stacked Sparse Autoencoder (SSAE) algorithm for fruit disease classification. Furthermore, the BRO technique is also employed for optimum hyperparameter tuning of the SSAE technique. The proposed BROFF-FDGC system was simulated extensively for validation using the test database and the outcomes established the enhanced performance of the proposed system. The obtained outcomes emphasize the superior performance of the BROFF-FDGC approach than the existing methodologies.

Keywords: battle royale optimization; fruit disease detection; computer vision; agriculture; deep learning
Mathematics Subject Classification: 65D19, 68T07

1. Introduction

Automatic Fruit Disease Detection (FDD) refers to a process of identifying the presence of diseases in fruit crops. Image analysis, sensor technology, and Machine Learning (ML) are some of the techniques through which the diseases can be detected. The ML techniques remain the most common method for automatic detection of fruit disease while this technique analyzes the images of the fruits [1]. For this purpose, a large set of images of both diseased as well as healthy fruits are collected and utilized to train the ML method. Then, the technique detects the patterns in these images that are indicative of diseases [2]. Sensor technology is another method used to collect the data on fruit crops. This includes humidity, other environmental conditions, temperature measurements and readings from sensors that confirm the presence of particular diseases [3]. The collected data can be used for fruit classification i.e., either diseased or healthy. There exist numerous benefits when utilizing an automated FDD system. One of the most important advantages is its accuracy and quick disease detection [4], which helps in the prevention of disease and protection of crop health. Automatic disease detection systems can be utilized for monitoring vast areas of crops, thus reducing the requirement for manual inspection. Also, these automatic systems are unbiased and highly objective than the human analysis, which in turn helps in minimizing the errors and enriching the precision of disease detection process [5]. However, there exists a few difficulties too when utilizing an automatic FDD system [6]. The necessity for high-quality and accurate data is one of the most challenging issues faced in automatic FDD system. The method cannot detect the disease accurately, if the data used to train the model is of poor quality or incomplete [7]. Further, another difficulty is the demand for robust algorithms that can manage variations in the dataset. For instance, the method should be aware of different kinds of fruits that may have diverse features. At last, the automatic systems necessitate reliable software and hardware infrastructure for proper functioning [8].

There has been a significant surge in the application of image processing and machine vision technologies for enriching the quality of fruit surface images. This is attributed to the reason that such technologies provide huge benefits in areas, where the human eye is not sensitive. Thus, the application of image processing and Computer Vision (CV) methods overcomes the challenges through subjective industrial quality control approaches [9]. Although the performance of the ML algorithms developed

earlier is good enough, these methods mostly rely upon the image types (of fruit crops) used during testing and training periods. Further, the outcomes depend on the handcrafted feature extraction techniques, which are again a labor-intensive one. Additionally, such methods have been tested and trained only for smaller datasets, which raises the risk of biased estimations. An alternate method to overcome the abovementioned issues is to leverage the Deep Learning (DL) methods and develop a fruit classification and grading mechanism [10]. Such DL-based technique should be capable of deriving the appropriate features mechanically without any need for manual intervention [11].

In this background, the current research paper introduces a novel Battle Royale Optimization with a Feature Fusion Based Fruit Disease Grading and Classification (BROFF-FDGC) approach. In the proposed BROFF-FDGC approach, the Bilateral Filtering (BF) model is initially implemented for noise elimination. Besides, a compendium of DL methods such as the Inception v3, NASNet, and Xception methods is also utilized for feature extraction. In addition to this, the Bayesian optimization (BO) algorithm is used as a hyperparameter optimizer. Moreover, the BROFF-FDGC model utilizes the stacked sparse-AE(SSAE) technique for classification purpose. Furthermore, the BRO technique is also utilized for optimum hyperparameter tuning of the SSAE technique. The proposed BROFF-FDGC model was extensively experimented through simulation while the outcomes showcase the enriched performance of the model.

The remaining sections of the article are arranged as briefed herewith. Section 2 offers the literature review and Section 3 discusses about the proposed method. Then, Section 4 elaborates on the evaluation results and Section 5 concludes the work.

2. Related works

Majid et al. [12] examined an integrated DL architecture for fruit disease classification. At first, the researchers applied a data increase and two different kinds of features were derived. In the initial feature type, both color and texture features were derived while in the next type, the DL features were extracted through a pre-trained process. In general, the pre-trained process can be reused with Transfer Learning (TL). Then, both the features were fused through the maximal mean value of the serial technique after which a harmonic threshold-based genetic algorithm was used for the optimization of the resultant fused vector.

Shah et al. [13] proposed a new computerized technique featuring Ant Colony Optimization (ACO)-oriented choice with the help of DL technique. The presented technique had four basic steps; a combination of the derived deep features through matrix length, data augmentation for solving the imbalanced data set, selection of better features utilizing ACO and hybrid Neighbourhood Component Analysis (NCA) methods, and fine-tuned pre-trained DL techniques (such as MobileNet-V2 and NasNet Mobile). The best-selective features were then passed onto multiple classifiers for ultimate recognition. Mostafa et al. [14] leveraged a deep CNN (DCNN)-related data enhancement method by utilizing the unsharp masking technique and color-histogram equalization in order to identify distinct guava plant species. In the presented technique, the data was first preprocessed and normalized. The presented study utilized five NN frameworks such as ResNet-50, AlexNet, GoogLeNet, ResNet-101, and SqueezeNet for identifying distinct guava plant species.

The authors in the study conducted earlier [15] employed a two-phase DCNN methodology for citrus disease classification and plant disease detection by exploiting the leaf images. The proposed methodology had two key stages; categorization of the majorly targeted area to a particular disease;

and its classification utilizing a classifier, thus suggesting the effectively-targeted diseased areas in the RPN. In literature [16], the authors examined novel DL-related citrus disease recognition and classification models. An innovative DL-related AlexNet architecture was used in this study for effectual disease detection. The Otsu method was executed in this study for image segmentation after which the application of Alex-Net architecture was executed as an extraction procedure. Afterwards, the Random Forest (RF) method was utilized for categorizing different types of citrus diseases. In addition to this, the Adaptive Gamma Correction (AGC) method was also enforced to enhance the contrast of citrus images. Nikhitha et al. [17] emphasized on formulating a user-friendly device that can detect and grade the disease levels. The inception model utilized CNNs for the classification while the latter was retrained through the TL method. The presented mechanism even ranked the fruit depending on the infection percentage.

In literature [18], the author focused on grape diseases and presented a new structure for identification and classification of selective diseases at the initial phase. A DL-related solution was entrenched into a typical architecture for optimum performance. In this model, three major stages were involved (a) Fusion of stronger features utilizing the presented approach after which the features were subjected to classifier step utilizing the Least Squared Support Vector Machine (LS-SVM), (b) feature extraction after the application of TL on pretrained deep methods such as ResNet101 and AlexNet, and (c) selecting the optimal features through the proposed Yager Entropy along with Kurtosis (YEaK) method. Kejriwal et al. [19] intended to identify and detect the foliar diseases with the help of apple leaf images. Involving a professional to detect the infections is not only time-consuming, but also inefficient, and expensive for large orchards. So, this technique was presented in which a group of three pre-trained DCNNs such as InceptionResNetV2, ResNet101V2, and Xception was used to categorize the apple tree leaves as either infected or healthy based on five disease classes.

3. The proposed model

In the current study, an innovative BROFF-FDGC method has been introduced for identification and classification of the fruit disease. The proposed BROFF-FDGC technique incorporates the following processes namely, BF-based noise removal, fusion-based feature extraction, BO-based hyperparameter tuning, SSAE classification, and BRO-based parameter optimization. Figure 1 represents the working flow of the BROFF-FDGC system.

3.1. BF-based noise removal process

The BF process is used to smoothen the images, while at the same time, preserving the rest of the fine details and edges in the image [20]. BF is used in image processing techniques to reduce the number of details and noise in the image and also preserve the overall structure of the image. The BF functions employ a weighted average for all the pixels in the image while the closest pixel is used as weight. The weight is measured based on the intensity difference and spatial distance between the pixel and its neighbors. This enables the filter to preserve other fine details and edges in the image, as the pixel with the largest spatial distance or highest intensity difference from the neighbor is given lower weight and remains less affected by the filter. BF is wisely applied in medical images for smoothening the images, while preserving the significant structures like tissue boundaries and blood vessels.

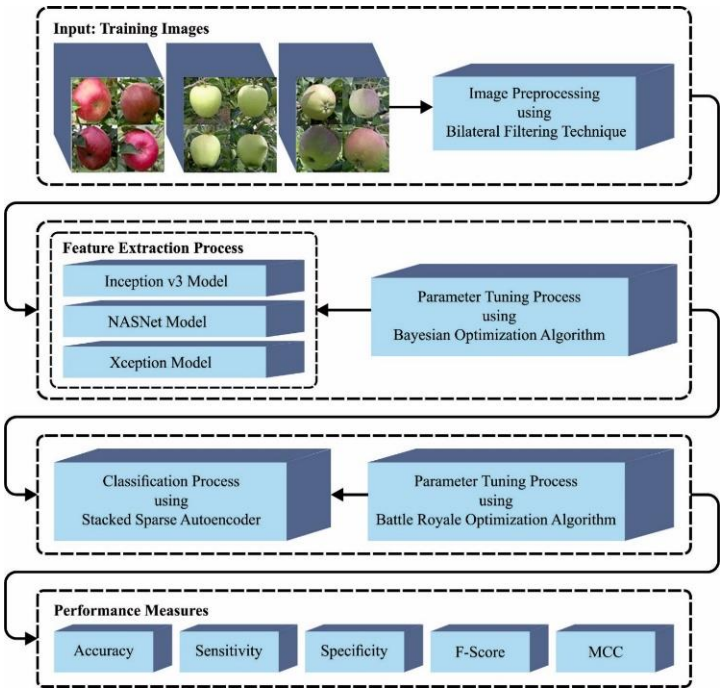

Figure 1. Working procedure of the BROFF-FDGC method.

3.2. Fusion-based feature extraction process

In this study, a fusion of DL architectures like Inception v3, NASNet, and Xception models is executed for the feature extraction process. Data fusion technique has been employed in a diverse range of CV and ML applications [21]. Feature fusion is a crucial task that incorporates multiple feature vectors. This technique depends on feature fusion using the formulated entropy.

The three vectors are defined as follows

$$f_{Inception\times n} = \{Inception_{1\times1}, Inception_{1\times2}, Inception_{1\times3}, \cdots, Inception_{1\times n}\}, \quad (1)$$

$$f_{\mathrm{NNet1}\times m} = \{\mathrm{NNet}_{1\times1}, NNet_{1\times2}, NNet_{1\times3}, \cdots, NNet_{1\times n}\}, \quad (2)$$

$$f_{\mathrm{Xception}\ 1\times p} = \{\mathrm{Xception}_{1\times1}, \mathrm{Xception}_{1\times2}, \mathrm{Xception}_{1\times3}, \cdots, \mathrm{Xception}_{1\times n}\}. \quad (3)$$

Furthermore, the extracted feature is combined in a single vector.

$$Fused(features\ vector)_{1\times q} = \sum_{i=1}^{3}\{fInception_{1\times n}, fNNet_{1\times m}, f\mathrm{Xception}_{1\times p}\}. \quad (4)$$

Here, f refers to the fused vector (1 × 1186). The entropy is exploited to feature the vectors for the optimum Feature Selection (FSs) outcomes as per the score. The FS technique can be expressed mathematically as shown in the Eqs (1) and (4). Entropy is used for selecting 1186 score-assisted features in 7835 features.

$$B_{He} = -NHe_b \sum_{i=1}^{n} p\,(f_i), \quad (5)$$

$$F_{select} = B_{He}\big(\max(f_i, 1186)\big). \quad (6)$$

In these expressions, p denotes the feature probability and He signifies the entropy. The last selected feature is then fed into the classifiers for differentiating the glioma and healthy images.

3.2.1. Inceptionv3 model

InceptionV3 model outperforms the preceding inception architecture through cost-effective computational outcomes [22]. Inception module remains the core component of the inception architecture. The inception model allows for deep networking and effective computation through a reduction dimension with a stacked $1x1$ convolutional layer. The key aim of the proposed technique is to address the problems of computation cost and overfitting among various other problems. The fundamental concept behind the inception model is to generate different filters of various dimensions that can run in parallel mode instead of series mode. The network in the inception module has an additional $1x1$ convolutional layer before the 3×3 and $5\mathrm{x}5$ convolutional layers and this feature makes the procedure, a highly robust and computationally inexpensive one. A dense $128x1$ layer exchanges the classifier part of the models, viz., head of the model, $3x1$ and $128x$, and $12x1$ for ternary and binary classifications, correspondingly. Afterwards, the model is fine-tuned upon the input images for extracting the best features. For training purposes, InceptionV3 model is fed with input images sized at $224x224x3$ while the input passes through different inception models in later stages and it might assist in preventing over-fitting as well as decreasing the computation cost. After being passed over the Inception module, the inputs are then passed to the dense layer of different sizes such as $128x1$ and $3x1$ or $2x1$ for the classifier model.

3.2.2. NASNet model

NASNet is a framework created with the help of a neural structural search method [23]. The search system called the Neural Architecture Search (NAS) as it exploits a control NN to develop a better CNN structure for the given data. The type of NASNet that was exploited in InstaCovNet19 was originally intended for the data named ImageNet. In this design, two classes of convolution cells are exploited, viz., the standard cell, and the reduction cell. Particularly, NASNet has been increased for the ImageNet data that comprises of images from every walk of life. In this case, a pre-trained NASNet infrastructure has been presented. The lack of huge datasets needs the usage of a pre-trained module. During the fine-tuning process, NASNet is provided with input images sized $224x224x3$. Then, the input images pass over various reduction and normal layers that extract better features from these images. At last, the attained features are provided as two Dense layers sized at $128x1$ and $3x1$ for the purpose of classification.

3.2.3. Xception model

Xception refers to 'extreme inception' and it comprises of 36 deep layers apart from the FC layer. Xception contains depthwise convolution layers including the MobileNet model and it comprises of 'shortcuts', in which the outcome of a certain layer is added to the output from the preceding layers. Unlike the InceptionV3, the Xception model combines the input records with compacted lumps, thus mapping the spatial connection for every channel autonomously; later, $1x1$ depthwise convolutional layer is implemented to capture the cross-channel connections. The Xception model overtakes the

InceptionV3 model with ImageNet dataset classification. In this case, a pre-trained Xception module (trained on ImageNet data) is developed. The pre-trained method is utilized due to the lack of a huge database for detection. The classification models, viz., head of the architecture, are later exchanged with dense layers sized at $128x1$ and $2x1$ for binary classification and $128x1$ and $2x1$ for ternary classification, correspondingly.

3.2.4. BO-based hyperparameter tuning

The BO technique is exploited for optimum hyperparameter adjustment process. BO functions by creating a posterior distribution of the function viz., Gaussian procedure that better describes the operation to be optimized [24]. When the observation counts increase, the posterior distribution also increases. Then, it becomes more apparent, especially at the region in parameter space that is worth computing and exploring. BO technique comprises of two key elements: A statistical method for modeling the objective function and an acquisition function for determining the next set of samples. Alternatively, the acquisition function offers sample points in the searching region as well. The function drives the tradeoff exploration and exploitation processes. Exploitation is a sample whereas the statistical models predict the highest objective scores while exploration refers to the return of a solution: The point estimate is either with a large posterior mean or large (x). The sampling is done at the location where the forecast uncertainty is higher. Then, the objective is estimated based on the initial space-filling analysis model. This model frequently contains only those points that are selected randomly and iteratively used for allocating the remainder of the N function's budget evaluation, demonstrated in Algorithm 1. For the BO algorithm, the time complexity is (n^3) in which n represents the observation counts. The time complexity for DL is $(w \cdot m \cdot e)$, where w represents the weight counts, m indicates the learning instance counts, and e indicates the running epochs. In this study, hyperparameter tuning was conducted by the researchers based on the attention module. The filter counts exploited in every attention layer can be enhanced. The optimization technique defines a better integration of the filter numbers for every attention layer.

Algorithm 1. Bayesian optimization algorithm.
Place the Gaussian process before f.
Observe f at n_0 point based on the space-filling experimental model.
Set $n = n_0$.
While $n \leq N$ do
Upgrade the posterior probability distribution on f utilizing each accessible dataset;
Consider x_n as a maximizer of the acquisition function over x, from which the acquisition function was calculated through the existing posterior distribution;
Observe $y_n = f(x_n)$;
Increment n;
End

3.3. Optimal fruit classification process

In order to identify the fruit diseases, the SSAE classifier is used. AE is a NN with multiple Hidden Layers (HLs) of unsupervised feature learning [25]. The basic design is to apply more than

one layer of NN for mapping the input dataset to attain the output vectors. Furthermore, the AE is applied to reduce the data feature sizes that could characterize both linear and non-linear conversions. The unlabeled input vector is exposed to weight mapping so as to attain the values of HL resultant vector as formulated below.

$$y_i = f_\theta(x_j) = S\left(\sum_{j=1}^{N} W_{ij}\, x_j + b_i\right). \tag{7}$$

Let y_i be the activation value of the HL, W_{ij} indicates the weight coefficient, b_i represents the offset vector of HL and $S(x)$ represents the activation function and the mathematical expression is:

$$L(x_{i,} y_i) = \frac{1}{2}\|x_i - y_i\|^2. \tag{8}$$

The weight parameter from the input layer to the HL is $\theta = \{W, b\}$, and the weight parameter in the HL to the resultant layer is represented as $\theta' = \{W', b'\}$. Figure 2 shows the architecture of the SSAE model.

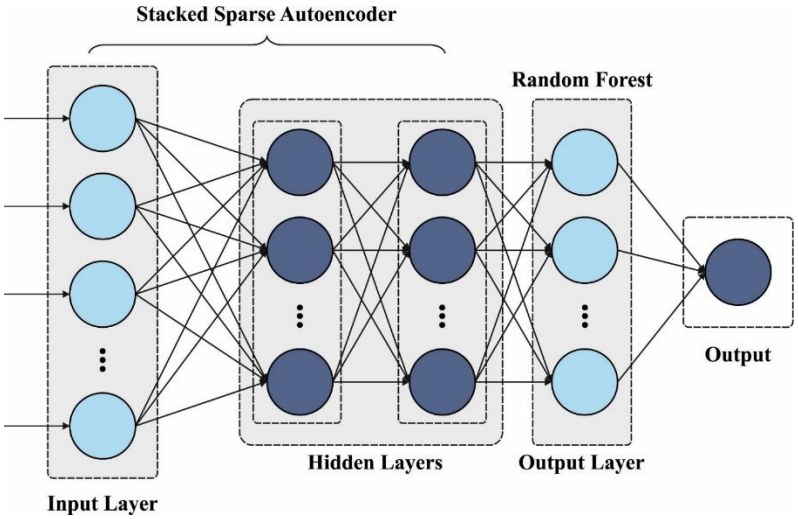

Figure 2. Architecture of the SSAE model.

Once a certain number of neurons is acquired from the HL, then they are considered as features so as to reduce the data size. Due to the existence of various neurons from the HL, a sparsity constraint is added for training the network and extracting valuable features.

$$J_S = J + \beta \sum_{j=1}^{S_2} K_L\left(\rho \| \rho_j\right) c, \tag{9}$$

here, ρ indicates the sparsity parameter; S_2 denotes the number of neurons from the HL; β characterizes the penalty factor to control sparsity; and $K_L(\rho \| \rho_j)$ indicates the difference between ρ and $\hat{\rho}_j$.

Typically, a simple SAE is not suitable for training purposes. So, the current study adopts a stacking technique in which every HL is separately trained with unsupervised learning of SAE. Then,

these layers are linked to produce a stacked network. Every sensor extracts 31 feature parameters in the vibration signal which are signified as $31\text{x}N$ input feature vector as SSAE.

To raise the classification effectiveness of the SSAE system, the BRO technique is utilized in the current study. Rahkar Farshi (2020) introduced the BRO algorithm based on battle royal video game. In this game, the player needs to make use of and explore the surroundings to survive [26]. All the players outside the safer region get affected in two ways while the former is exposed to the risk of being removed from the battle.

The early population of the BRO technique is made up of n randomly-produced individuals in D-dimensional region. $x_i^t(i \in [1,n])$ and $x_{i,dam}^t$ represent the location data and damage level of the i^{th} player in t^{th} iteration amongst the population, correspondingly. The player from the best location who stayed back in the original location is termed as the winner. However, the loss level is fixed as 0. The player in a worse position gets damaged during a mutual attack and so, they are named as loser. The loss levels of the loser can be upgraded, as demonstrated in Eq (10). Furthermore, due to the global best place and the present place, the location of the loser is upgraded, as demonstrated in Eq (11) and it aims to defend itself and attack the enemy from another side.

$$x_{i,dam}^{t+1} = x_i^t, dam + 1, \tag{10}$$

$$x_i^{t+1} = x_i^t + r(x_{best} - x_i^t). \tag{11}$$

In this expression, r indicates the coefficient $(r \in [0,1])$. Once a player becomes the winner, then the damage levels return to 0. In order to emphasize the exploration process, once the player's cumulative damage levels exceed the predetermined value $(thre = 3)$, the player dies and respawns from the present safer region. In this scenario, the damage level returns to 0. The average player respawns from the present safer region, as given below.

$$x_i^{t+1} = r(ub_d - lb_d) + lb_d. \tag{12}$$

In Eq (12), ub_d and lb_d denote the lower and upper boundaries of the safer region. Furthermore, during the iteration procedure, if the iteration count is better than the area update threshold (Δ), then the safer region gets reduced by the global optimum location while the center, the upper and lower limits are demonstrated as follows:

$$lb_d = x_{best} - SD(\overline{x_d}), \tag{13}$$

$$ub_d = x_{best} + SD(\overline{x_d}), \tag{14}$$

here, $SD(\overline{x_d})$ denotes the standard deviation, x_{best} indicates the existing optimum solution. lb_d and ub_d exceed the lower and upper boundary of the solution space as given below.

$$\Delta = \Delta + round\left(\frac{\Delta}{2}\right). \tag{15}$$

The initial value Δ is $\log_{10}(T_{\max}),\ and\ T_{\max}$ represents the maximal iteration counts. The pseudocode of the BRO algorithm is shown in Algorithm 2.

The BRO algorithm grows an FF to realize a better effectiveness of the classifier and finds a positive integer to represent the optimal outcomes of the candidate. Here, FF corresponds to a reduced classifier error rate.

$$fitness(x_i) = ClassifierErrorRate(x_i)$$
$$= \frac{No.\ of\ misclassified\ instances}{Total\ no.of\ instances} \times 100. \quad (16)$$

Algorithm 2. Pseudocode of BRO algorithm.

Start

Randomly initializing a population (x_n), and each parameter;

$Shrink = ceil(Iog_{10}(T_{\max}))$;

$\Delta = round\left(\frac{T_{\max}}{Shrink}\right)$;

For $i =$ I: $T_{\max}$

Find the nearby player (x_j^t) by computing the Euclidean distance;

$d = i; v = j$

If $f(x_i^t) < f(x_j^t)$

$d = j; v = i$;

End if

If $X_{d,dam}^t < thre$

Upgrade the damage level and the location of the loser;

Else

The loser respawns in the present safer region;

$x_{\mathrm{d},dam}^t = 0$

End if

Re-evaluate the fitness value of x_i^t;

$x_{v,dam}^t = 0$;

If $t \geq \Delta$

Upgrade ub_d and Ib_d;

Upgrade the threshold (Δ) ;

End if

If Ib_d/ub_d exceeds the Lower or upper limits of the solution space, then set to the original ub_d and Ib_d;

Record the better individual and the fitness value.

End

4. Performance validation

The performance of the proposed BROFF-FDGC system, in terms of FDD, was validated using the the CASC IFW Database [27]. Table 1 provides the details about the dataset. Figure 3 shows some of the sample images. The images in the dataset were collected from the Internal Feeding Worm (IFW)

dataset of Comprehensive Automation for Specialty Crops (CASC) research work after obtaining previous consent from Purdue University. The dataset contains images for four cultivars of apples: Golden Delicious, Fuji, York, and Red Delicious. Every cultivar has images in different phases of ripening of defective and healthy apples. The images of the damaged apples show noticeable dark spots on the outside skin as a result of damage from internal feeding worms. Individual apple images were clipped to a size range of 120 × 120 pixels for the purpose of image processing.

Table 1. Details of the database.

Class	No. of Instances
NON-HEALTHY	3800
HEALTHY	2058
Total No. of Instances	5858

Figure 3. Sample images.

The confusion matrices generated by the BROFF-FDGC methodology on the FDD process are shown in Figure 4. The simulation values specify that the BROFF-FDGC approach correctly classified the healthy and damaged apples.

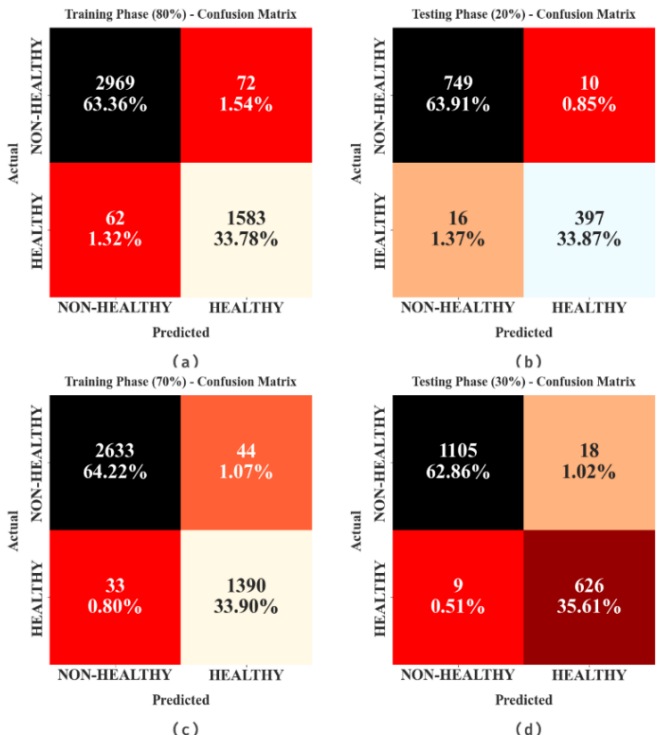

Figure 4. Confusion matrices of the BROFF-FDGC algorithm (a and b) 80% of TR set/20% of TS set and (c and d) 70% of TR set/30% of TS set.

Table 2 offers a brief overview on the FDD results accomplished by the BROFF-FDGC methodology with 80:20 of TR phase and TS phase. Figure 5 exhibits the overall FDD performance of the BROFF-FDGC system for 80% of the TR phase. The simulation values indicate that the BROFF-FDGC technique classified non-healthy and healthy apple images properly. In addition, it is also observed that the BROFF-FDGC technique attained an average $accu_{bal}$ of 96.93%, $sens_y$ of 96.93%, $spec_y$ of 96.93%, F_{score} of 96.87%, and an MCC of 93.73%.

Table 2. FDD analysis outcomes of the BROFF-FDGC methodology on 80:20 of TR set/TS set.

Class	$Accu_{bal}$	$Sens_y$	$Spec_y$	F_{Score}	MCC
Training Phase (80%)					
NON-HEALTHY	97.63	97.63	96.23	97.79	93.73
HEALTHY	96.23	96.23	97.63	95.94	93.73
Average	96.93	96.93	96.93	96.87	93.73
Testing Phase (20%)					
IFW DAMAGED	98.68	98.68	96.13	98.29	95.13
HEALTHY	96.13	96.13	98.68	96.83	95.13
Average	97.40	97.40	97.40	97.56	95.13

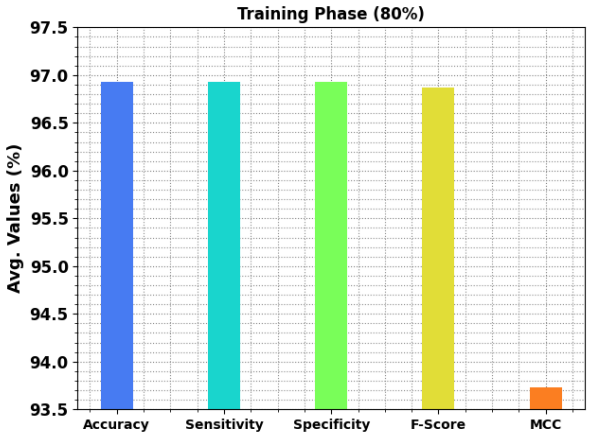

Figure 5. Average outcomes of the BROFF-FDGC methodology on 80% of TR set.

Figure 6 demonstrates the overall FDD examination outcomes achieved by the BROFF-FDGC technique on 20% of the TS set. The results point out that the BROFF-FDGC model can recognize non-healthy and healthy apple images accurately. Furthermore, it is also observed that the BROFF-FDGC technique accomplished an average $accu_{bal}$ of 97.40%, $sens_y$ of 97.40%, $spec_y$ of 97.40%, F_{score} of 97.56%, and an MCC of 95.13%.

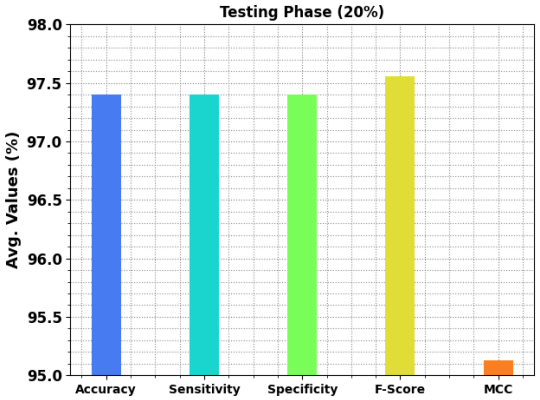

Figure 6. Average outcomes of the BROFF-FDGC methodology on 20% of TS set.

Table 3 shows the comprehensive FDD analytical outcomes attained by the BROFF-FDGC technique with 70:30 of the TR set and TS set. Figure 7 shows the overall FDD analysis outcomes of the BROFF-FDGC system at 70% of the TR set. The obtained outcomes infer that the BROFF-FDGC technique identified non-healthy and healthy apple images accurately. Furthermore, it is also noted that the BROFF-FDGC approach accomplished an average $accu_{bal}$ of 98.02%, $sens_y$ of 98.02%, $spec_y$ of 98.02%, F_{score} of 97.93%, and an MCC of 95.87%.

Table 3. FDD analysis outcomes of the BROFF-FDGC system on 70:30 of TR set/TS set.

Class	$Accu_{bal}$	$Sens_y$	$Spec_y$	F_{Score}	MCC
Training Phase (70%)					
NON-HEALTHY	98.36	98.36	97.68	98.56	95.87
HEALTHY	97.68	97.68	98.36	97.30	95.87
Average	98.02	98.02	98.02	97.93	95.87
Testing Phase (30%)					
NON-HEALTHY	98.40	98.40	98.58	98.79	96.69
HEALTHY	98.58	98.58	98.40	97.89	96.69
Average	98.49	98.49	98.49	98.34	96.69

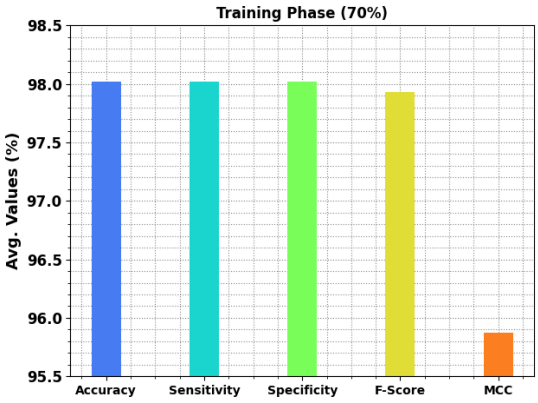

Figure 7. Average outcomes of the BROFF-FDGC algorithm on 70% of TR set.

Figure 8 shows the overall FDD examination outcomes produced by the BROFF-FDGC method on 30% of the TS set. The outcomes indicate that the BROFF-FDGC method detected non-healthy as well as healthy apple images appropriately. Besides, it is observed that the BROFF-FDGC methodology accomplished an average $accu_{bal}$ of 98.49%, $sens_y$ of 98.49%, $spec_y$ of 98.49%, F_{score} of 98.34%, and an MCC of 96.69%.

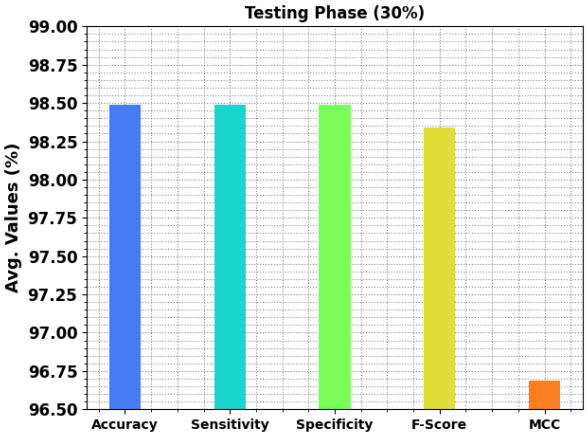

Figure 8. Average outcomes of the BROFF-FDGC methodology on 30% of TS set.

The TRAC and VDAC values achieved by the BROFF-FDGC system for the FDD solution are shown in Figure 9. The outcomes exhibit that the BROFF-FDGC method produced excellent outcomes with high TRAC and VDAC values. The BROFF-FDGC system attained the highest TRAC performance.

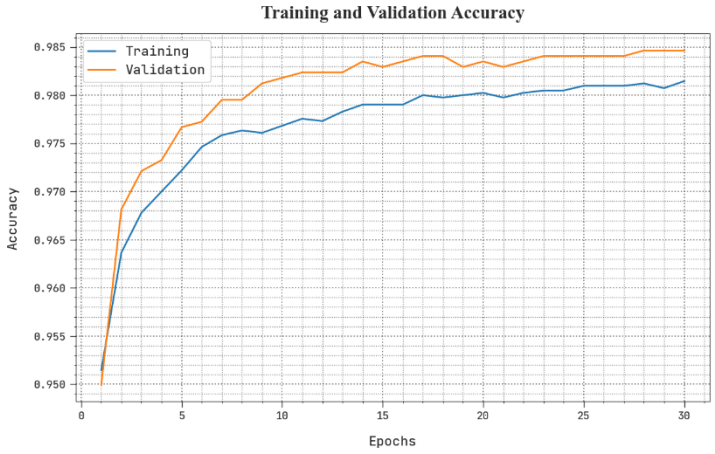

Figure 9. TRAC and VDAC outcomes of the BROFF-FDGC approach.

The TRLS and VDLS values, accomplished by the BROFF-FDGC technique in terms of FDD solution, are shown in Figure 10. The outcomes infer that the BROFF-FDGC system achieved better outcomes with lesser TRLS and VDLS values. This is obvious due to the fact that the BROFF-FDGC technique achieved the least VDLS performance.

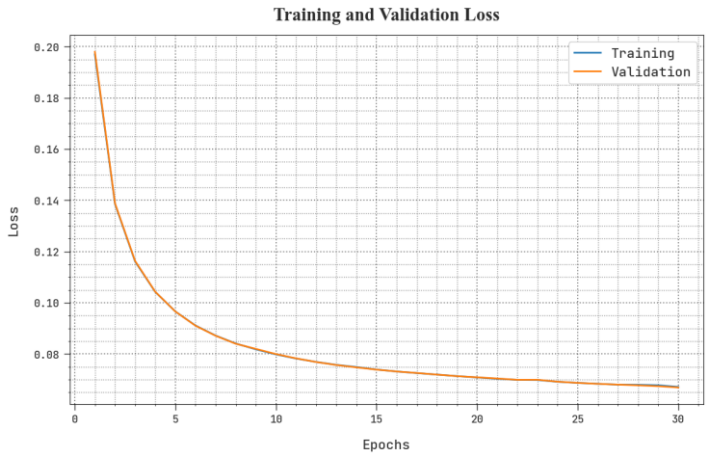

Figure 10. TRLS and VDLS outcomes of the BROFF-FDGC approach.

Figure 11 shows an apparent PR curve plotted with the values produced by the BROFF-FDGC system on the test database. The outcomes denote that the BROFF-FDGC method improved the PR outcomes on both the classes.

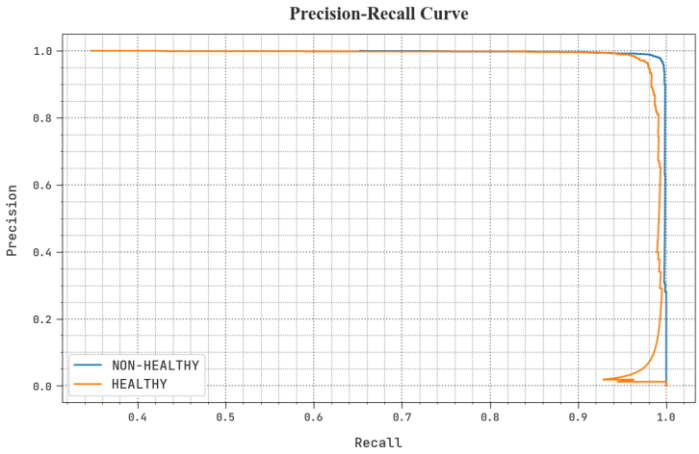

Figure 11. Precision-recall outcomes of the BROFF-FDGC methodology.

A comprehensive ROC inspection was conducted upon the BROFF-FDGC technique using the test database and the results are shown in Figure 12. The simulation values imply that the BROFF-FDGC methodology established its ability on both the classes.

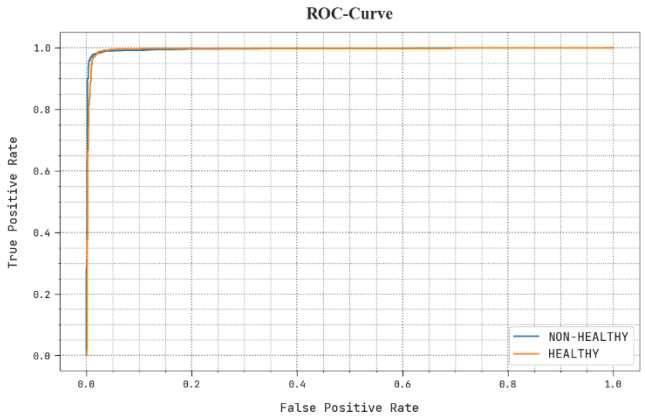

Figure 12. ROC outcomes of the BROFF-FDGC methodology.

Table 4 shows the comprehensive analysis outcomes of the BROFF-FDGC system with recent models [28]. Figure 13 exhibits the comparative analysis results achieved by the BROFF-FDGC approach in terms of $accu_y$. The simulation values indicate that the BROFF-FDGC approach gained an optimal solution with the maximal values of $accu_y$. in terms of $accu_y$, the BROFF-FDGC technique yielded a maximum $accu_y$ of 98.49% while the ResNet-50, DenseNet-121, NASNetA, EfficientNetB0, EfficientNetB1, and the EfficientNetB2 models accomplished the least $accu_y$ values such as 91.94%, 96.96%, 96.79%, 96.53%, 98.06%, and 98.15% correspondingly.

Table 4. Comparative analysis outcomes of the BROFF-FDGC method and other recent methodologies.

Methods	$Sens_y$	$Spec_y$	$Accu_y$
BROFF-FDGC	98.49	98.49	98.49
ResNet-50	86.65	86.53	91.94
DenseNet-121	99.01	97.54	96.96
NASNetA	97.39	98.26	96.79
EfficientNetB0	97.71	98.59	96.53
EfficientNetB1	97.69	99.91	98.06
EfficientNetB2	97.88	98.34	98.15

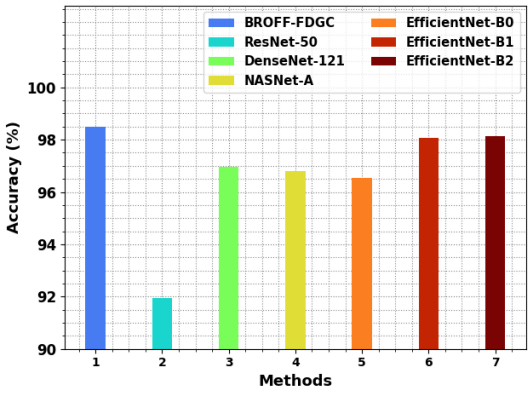

Figure 13. $Accu_y$ outcomes of the BROFF-FDGC algorithm and other recent methods.

Figure 14 exhibits the comparative analytical outcomes of the BROFF-FDGC methodology in terms of $sens_y$ and $spec_y$. The outcomes demonstrate that the BROFF-FDGC system attained better performance with the maximum values of $sens_y$ and $spec_y$. With regards to $sens_y$, the BROFF-FDGC technique produced a high $sens_y$ of 98.49% whereas the ResNet-50, DenseNet-121, NASNetA, EfficientNetB0, EfficientNetB1, and the EfficientNetB2 techniques achieved the least $sens_y$ values such as 86.65%, 99.01%, 97.39%, 97.71%, 97.69%, and 97.88% correspondingly.

Also, in terms of $spec_y$, the BROFF-FDGC method accomplished a maximum $spec_y$ of 98.49% while the ResNet-50, DenseNet-121, NASNetA, EfficientNetB0, EfficientNetB1, and the EfficientNetB2 techniques achieved the least $spec_y$ values such as 86.53%, 97.54%, 98.26%, 98.59%, 99.91%, and 98.34% correspondingly. Therefore, it can be inferred that the BROFF-FDGC technique shows maximum performance on FDD over other DL models.

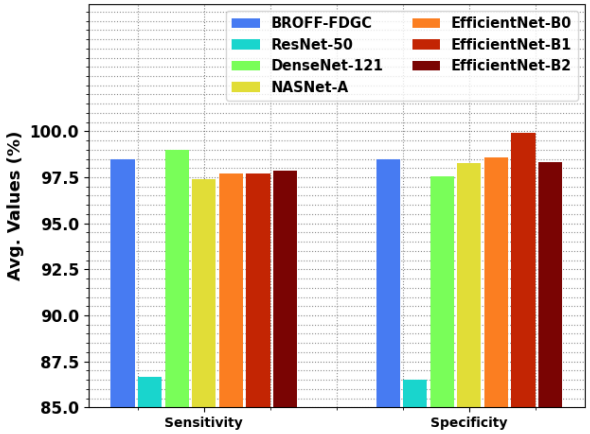

Figure 14. $Sens_y$ and $spec_y$ outcome of the BROFF-FDGC algorithm with other recent systems.

5. Conclusions

In the current study, a novel BROFF-FDGC methodology has been presented for detection and classification of the fruit diseases. The proposed BROFF-FDGC method integrates the BF-based noise removal, fusion-based feature extraction, BO-based hyperparameter tuning, SSAE classification, and BRO-based parameter optimization processes. A fusion of DL models namely Inception v3, NASNet, and Xception models is executed for the purpose of feature extraction. Besides, the BRO algorithm is also employed for optimum hyperparameter tuning of the SSAE technique. The BROFF-FDGC algorithm was extensively validated through simulation and the outcomes exhibit the greater performance. The obtained outcomes display the superior performance of the BROFF-FDGC approach compared to the rest of the methods. In the future, hybrid metaheuristic optimization algorithms can be designed to improve the FDD performance.

Use of AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Conflict of interest

The authors declare that they have no conflicts of interest. The manuscript was written through the contributions of all authors. All authors have approved the final version of the manuscript.

References

1. C. C. Ukwuoma, Q. Zhiguang, M. B. B. Heyat, L. Ali, Z. Almaspoor, H. N. Monday, Recent advancements in fruit detection and classification using deep learning techniques, *Math. Probl. Eng.*, **2022** (2022), 9210947. https://doi.org/10.1155/2022/9210947

2. A. Khattak, M. U. Asghar, U. Batool, M. Z. Asghar, H. Ullah, M. Al-Rakhami, et al., Automatic detection of citrus fruit and leaves diseases using deep neural network model, *IEEE Access*, 9 (2021), 112942–112954. https://doi.org/10.1109/ACCESS.2021.3096895
3. Y. Gulzar, Fruit image classification model based on MobileNetV2 with deep transfer learning technique, *Sustainability*, **15** (2023), 1906. https://doi.org/10.3390/su15031906
4. X. Liu, L. Wei, C. Miao, Q. Zhang, J. Yan, S. Li, et al., Application of exogenous phenolic compounds in improving postharvest fruits quality: Classification, potential biochemical mechanisms and synergistic treatment, *Food Rev. Int.*, 2023. https://doi.org/10.1080/87559129.2023.2233599
5. B. Güven, İ. Baz, B. Kocaoğlu, E. Toprak, D. E. Barkana, B. S. Özdemir, Smart farming technologies for sustainable agriculture: From food to energy. In: *A sustainable green future: Perspectives on energy, economy, industry, cities and environment*, Springer, Cham, 2023, 481–506. https://doi.org/10.1007/978-3-031-24942-6_22
6. F. Li, Y. Zheng, S. Liu, F. Sun, H. Bai, A multi-objective apple leaf disease detection algorithm based on improved TPH-YOLOV5, *Appl. Fruit Sci.*, 2024. 1–17. https://doi.org/10.1007/s10341-024-01042-7
7. H. Li, Y. Jin, J. Zhong, R. Zhao, A fruit tree disease diagnosis model based on stacking ensemble learning, *Complexity*, **2021** (2021), 6868592. https://doi.org/10.1155/2021/6868592
8. P. Dhiman, V. Kukreja, P. Manoharan, A. Kaur, M. M. Kamruzzaman, I. B. Dhaou, et al., A novel deep learning model for detection of severity level of the disease in citrus fruits, *Electronics*, **11** (2022), 495. https://doi.org/10.3390/electronics11030495
9. V. Ashok, D. S. Vinod, A novel fusion of deep learning and android application for real-time mango fruits disease detection, In: *Intelligent system design*, Singapore: Springer, 2020, 781–791. https://doi.org/10.1007/978-981-15-5400-1_74
10. M. S. Hossen, I. Haque, M. S. Islam, M. T. Ahmed, M. J. Nime, M. A. Islam, Deep learning based classification of papaya disease recognition. In: *2020 3rd International Conference on Intelligent Sustainable Systems (ICISS)*, 2020, 945–951. https://doi.org/10.1109/ICISS49785.2020.9316106
11. S. Thite, K. Patil, R. Jadhav, Y. Suryawanshi, P. Chumchu, Empowering agricultural research: A comprehensive custard apple (*Annona squamosa*) disease dataset for precise detection, *Data Brief*, **53** (2024), 110078. https://doi.org/10.1016/j.dib.2024.110078
12. A. Majid, M. A. Khan, M. Alhaisoni, M. Asfand E. yar, U. Tariq, N. Hussain, et al., An integrated deep learning framework for fruits diseases classification, *CMC Comput. Mater. Con.*, **71** (2022), 1387–1402. https://doi.org/10.32604/cmc.2022.017701
13. F. A. Shah, M. A. Khan, M. Sharif, U. Tariq, A. Khan, S. Kadry, et al., A cascaded design of best features selection for fruit diseases recognition, *CMC-Comput. Mater. Con.*, **70** (2022), 1491–1507. https://doi.org/10.32604/cmc.2022.019490
14. A. M. Mostafa, S. A. Kumar, T. Meraj, H. T. Rauf, A. A. Alnuaim, M. A. Alkhayyal, Guava disease detection using deep convolutional neural networks: A case study of guava plants, *Appl. Sci.*, **12** (2022), 239. https://doi.org/10.3390/app12010239
15. S. F. Syed-Ab-Rahman, M. H. Hesamian, M. Prasad, Citrus disease detection and classification using end-to-end anchor-based deep learning model, *Appl. Intell.*, **52** (2022), 927–938. https://doi.org/10.1007/s10489-021-02452-w

16. C. Senthilkumar, M. Kamarasan, An effective classification of citrus fruits diseases using adaptive gamma correction with deep learning model, *Int. J. Eng. Adv. Technol.*, **9** (2020), 2618–2629. https://doi.org/10.35940/ijeat.B4066.129219
17. M. Nikhitha, S. R. Sri, B. U. Maheswari, Fruit recognition and grade of disease detection using inception v3 model. In: *2019 3rd International conference on Electronics, Communication and Aerospace Technology (ICECA)*, 2019, 1040–1043. https://doi.org/10.1109/ICECA.2019.8822095
18. A. Adeel, M. A. Khan, T. Akram, A. Sharif, M. Yasmin, T. Saba, et al., Entropy-controlled deep features selection framework for grape leaf diseases recognition, *Expert Syst.*, **39** (2022), e12569. https://doi.org/10.1111/exsy.12569
19. S. Kejriwal, D. Patadia, V. Sawant, Apple leaves diseases detection using deep convolutional neural networks and transfer learning, In: *Computer vision and machine learning in agriculture*, Singapore: Springer, 2022, 207–227. https://doi.org/10.1007/978-981-16-9991-7_13
20. T. Bahraini, T. Hamedani, S. M. Hosseini, H. S. Yazdi, Edge preserving range image smoothing using hybrid locally kernel-based weighted least square, *Appl. Soft Comput.*, **125** (2022), 109234. https://doi.org/10.1016/j.asoc.2022.109234
21. T. Saba, A. S. Mohamed, M. El-Affendi, J. Amin, M. Sharif, Brain tumor detection using fusion of hand crafted and deep learning features, *Cogn. Syst. Res.*, 59 (2020), 221–230. https://doi.org/10.1016/j.cogsys.2019.09.007
22. A. Gupta, Anjum, S. Gupta, R. Katarya, InstaCovNet-19: A deep learning classification model for the detection of COVID-19 patients using Chest X-ray, *Appl. Soft Comput.*, **99** (2021), 106859. https://doi.org/10.1016/j.asoc.2020.106859
23. A. Altamimi, F. Alrowais, H. Karamti, M. Umer, L. Cascone, I. Ashraf, An improved skin lesion detection solution using multi-step preprocessing features and NASNet transfer learning model, *Image Vision Comput.*, **144** (2024), 104969. https://doi.org/10.1016/j.imavis.2024.104969
24. Q. Liang, A. E. Gongora, Z. Ren, A. Tiihonen, Z. Liu, S. Sun, et al., Benchmarking the performance of Bayesian optimization across multiple experimental materials science domains, *npj Comput. Mater.*, **7** (2021), 188. https://doi.org/10.1038/s41524-021-00656-9
25. H. Bai, X. Zhan, H. Yan, L. Wen, Y. Yan, X. Jia, Research on diesel engine fault diagnosis method based on stacked sparse autoencoder and support vector machine, *Electronics*, **11** (2022), 2249. https://doi.org/10.3390/electronics11142249
26. H. Wu, X. Zhang, L. Song, C. Su, L. Gu, A hybrid improved BRO algorithm and its application in inverse kinematics of 7R 6DOF robot, *Adv. Mech. Eng.*, 2022. https://doi.org/10.1177/16878132221085125
27. The CASC IFW Database, Available from: https://engineering.purdue.edu/RVL/Database/IFW/index.html.
28. N. Ismail, O. A. Malik, Real-time visual inspection system for grading fruits using computer vision and deep learning techniques, *Inform. Process. Agric.*, **9** (2022), 24–37. https://doi.org/10.1016/j.inpa.2021.01.005

AIMS Press