An Integrated Framework of Two-Stream Deep Learning Models Optimal Information Fusion for Fruits Disease Recognition

Unber Zahra, Muhammad Attique Khan, *Member, IEEE*, Majed Alhaisoni, Areej Alasiry, Mehrez Marzougui, and Anum Masood

***Abstract*—Diseases impact the rates of production of many agricultural goods. These diseases require detection, which is difficult to do manually. Therefore, the creation of some automated illness detection systems is urgently required. Deep learning showed significant success in the area of precision agriculture for the recognition of plant disease. Compared with the traditional techniques, the deep learning architecture automatically extracts deep features from the deeper layer. In this work, we proposed a new automated method for classifying apple and grapefruit leaf disease recognition utilizing two-stream deep learning architecture. The proposed framework entails several steps. The first phase is picture contrast enhancement, which combines the information from DnCNN and top–bottom hat filtering to create a better image. Then, the augmentation process uses horizontal and vertical flips to increase the dataset's original size. The Inception-ResNet-V2 deep learning model is then adjusted and trained using deep transfer learning on the expanded dataset. After being extracted from the training model, the best features are chosen using two techniques—an entropy-based strategy and tree growth optimization. Finally, a new effective method combines the chosen features, and machine learning classifiers are used to complete the classification. On the augmented dataset, the proposed framework correctly classified apple and leaf diseases with the accuracy rates of 99.4% and 99.9%, respectively.**

***Index Terms*—Apple disease, contrast enhancement, deep learning (DL), denoising network, entropy, feature fusion, grape disease, tree growth algorithm.**

Manuscript received 18 July 2023; revised 4 September 2023; accepted 24 September 2023. Date of publication 5 December 2023; date of current version 16 January 2024. This work was supported by the Deanship of Scientific Research at King Khalid University through large group Research Project under Grant RGP2/249/44. *(Corresponding author: Anum Masood.)*

Unber Zahra is with the Department of Computer Science, HITEC University, Taxila 47080, Pakistan (e-mail: unber.zahra@hitecuni.edu.pk).

Muhammad Attique Khan is with the Department of Computer Science, HITEC University, Taxila 47080, Pakistan, and also with the Department of Computer Science and Mathematics, Lebanese American University, Beirut 47080, Lebanon (e-mail: attique.khan@ieee.org).

Majed Alhaisoni is with the Department of Computer Sciences, College of Computer and Information Sciences, Princess Nourah Bint Abdul Rahman University, Riyadh 11671, Saudi Arabia (e-mail: mmalhaisoni@pnu.edu.sa).

Areej Alasiry and Mehrez Marzougui are with the College of Computer Science, King Khalid University, Abha 61413, Saudi Arabia (e-mail: areej.alasiry@kku.edu.sa; mhrez@kku.edu.sa).

Anum Masood is with the Department of Physics, Norwegian University of Science and Technology, NO-7491 Trondheim, Norway (e-mail: anum.masood@ntnu.no).

Digital Object Identifier 10.1109/JSTARS.2023.3339297

I. Introduction

AGRICULTURE is a sector that is essential to meeting people's necessities and boosting a nation's economy [1]. All plants and fruits are subject to diseases that adversely affect production rates and quality [2]. Early care and fruit health monitoring are preferable to implement appropriate remedial action and avoid long-term financial damage. Early indicators of disease include some light spots or discolorations on the plant leaves [3]. Since it is now simple to survey large fields and automatic methods have been developed to detect fruit diseases in their early stages, the need for the manual involvement of human abilities is decreasing [4]. These automatic methods have substantially facilitated the procedure of diagnosing diseases. Using any sizable image dataset, various image processing methods and computer vision have been used to identify and categorize diseases in fruits [5]. Recently, diseases have increased in plants that affect them physically [6]. Therefore, identifying, treating, and managing diseases have become essential. The diseases are manually diagnosed by looking at the symptoms they cause. Any dataset, including damaged plant leaves, may be processed using automated methods, and after certain transformations and tweaks, the disease can be reliably diagnosed [7].

In the literature, classic automated methods do better with smaller dataset sizes. The classic approaches involve preprocessing images to improve contrast and remove noise, manually extracting features, such as color, shape, and texture, reducing features using principal component analysis, and utilizing machine learning classifiers [8], [9]. Deep learning (DL) models have recently significantly succeeded in computer vision, especially for agriculture and medical imaging. The DL models can capture the insight important information of each image that is later helpful in the accurate classification [10]. Several DL architectures have been introduced in the literature for the classification of plant leaf disease recognition, such as DFNeT [11], DLMC-Net [12], Custom CNN [9], and a few more [13]. Pretrained CNN models have been utilized to extract useful features in current agricultural research [14], [15], [16], [17]. However, feature fusion and selection approaches are used to achieve high precision in terms of time and accuracy [18]. Compared with an individual feature vector, feature fusion performs better since it combines the key information and improves the overall system performance in accuracy [19]. The disadvantage

of feature fusion is that it combines redundant and irrelevant data, which can be dealt with using feature selection or optimization processes. By using this method, unnecessary information can be removed while keeping the crucial elements [20].

Feature optimization is an important and hot research topic nowadays due to its direct influence on classification results [21], [22]. As previously stated, feature selection techniques eliminate unnecessary and irrelevant features significantly affecting the system's overall performance [23]. As a result, feature selection techniques reduce training and computing time, increase system prediction performance, remove duplicate information that takes up most prediction time, and considerably improve data classification [24], [25].

Zhu et al. [26] presented a method for grape disease recognition. A Wiener filtering method with a wavelet transform is used to denoise the photos. To increase the accuracy, morphological and segmentation methods have been applied. A method to identify plant diseases presented by Sladojevic et al. [27] comprises image preprocessing and augmentation. The next step is calculating features from the input images and distinguishing between the 13 distinct plant diseases through the Caffe DL model. This model obtains an accuracy of 96.3%, which was improved by the recent methods. Jhuria et al. [28] demonstrated a method for detecting fruit diseases. The imaging dataset is first preprocessed, which involves resizing each image to a size of 200×250 pixels. Next, morphological and texture features are employed independently for feature extraction, with morphological features providing finer results than texture features. Finally, classification is performed and obtained with 90% accuracy.

A. Major Problems

Other techniques were also presented, and they produced noteworthy accuracy. However, they struggle with challenges, such as the precise texture of various diseases, slight color changes among diseases, and redundant characteristics [29]. Other difficulties include the following.

1) Disease categorization for multiple fruits is difficult due to the similarity of fruit leaves texture, infected spots, and patches.
2) The majority of existing works have used handcrafted feature extractors to achieve classification, which limits performance and restricts results.
3) Irrelevant and redundant features mislead the classification problem and also lengthen computational time.

B. Major Contributions

This research developed a DL and optimal feature selection system to identify apple and grape leaf diseases. Rust, scab, black rot, and healthy class are the illnesses that affect apple leaves. Black rot, black measles, leaf blight, and a healthy class are all symptoms of grapefruit disease. The suggested framework's results are calculated utilizing a variety of classifiers and performance evaluation metrics. The following are the article's main contributions.

1) Developed a hybrid contrast enhancement technique to enhance an image's local information. Data augmentation is also carried out to expand the original dataset.
2) Inception-Resnet-V2 pretrained deep model is fine-tuned and trained on augmented contrast-enhanced images instead of original imaging through deep transfer learning. This process extracted useful features for the classification.
3) For the best feature selection, an entropy-controlled tree growth optimization approach was proposed. Second, time characteristics are chosen using entropy.
4) Proposed an intelligent fusion technique to combine the best-selected features. This technique improved the accuracy of the proposed framework.

The rest of this article is organized as follows. Section II discusses a few current investigations on the classification of plant and fruit diseases. The proposed methodology for detecting fruit leaf disease using DL is presented in Section III. Results and comparisons with currently used methods are reported in Section IV. Finally, Section V concludes this article.

II. Related Work

The diseases in plants are the reason for serious danger to plant development and agricultural productivity; it might result in food scarcity [30]. Therefore, researchers have conducted numerous investigations to identify plant diseases [15], [31], [32]. These methods are based on DL and conventional techniques [33]. For example, according to Chakraborty et al. [34], a system combining ML and DIP principles can identify diseases from contaminated apple leaves. The initial step in the disease recognition process is preprocessing using tools, such as the Otsu thresholding method and histogram equalization. Then, multiclass SVM distinguishes the original leaf image from the diseased kind with 96% accuracy via the image segmentation region of the affected section.

Rehman et al. [33] presented a parallel architecture for the real-time diagnosis and categorization of apple leaf disease. First, a hybrid contrast stretching technique to boost an image's visual impact is suggested, and then the MASK RCNN is set up to find the infected areas. For training the pretrained CNN model, the improved photos are used for feature extraction. A selection technique based on Kapur's entropy with the MSVM approach is developed to pick strong features for the final classification. The Plant Village dataset is used for the experimental procedure, and the ensemble subspace discriminant classifier achieves an improved accuracy of 96.6%. A DL-based automated detection system is proposed for examining and categorizing diseases affecting apple leaves by Alsayed et al. [35]. Various pretrained models, including InceptionV3, MobileNetV2, VGG16, and ResNet-V2, are considered using transfer learning. According to an analysis of several variables, including learning rate and optimizer, ResNet-V2 with the Adam optimizer is produced.

Di and Li [36] offered a new DL-based model for detecting four prevalent apple leaf diseases. This proposed work's major improvements include feature reuse in conjunction with

the DenseNet network, resizing and reorganizing parameters, using different convolution kernels for dimensionality reduction, and increasing network depth without increasing computational complexity. The mAP and average IoU values for this model are 99.99% and 90.88%, respectively. Yu et al. [37] developed the MSOResNet apple leaf disease identification model based on a residual network called ResNet50. The model's identification accuracy, speed, and parameter count are improved by decomposing the convolution kernel, updating the identity mapping technique, reducing the number of residual modules, and replacing the BN layer. This model achieved average accuracy, recall, and $F1$-score for identifying apple leaf diseases, which are 0.957, 0.958, and 0.957, respectively. Ji et al. [38] presented a combined CNN architecture based on an integrated strategy to separate healthy leaves from leaves with prevalent grape illnesses from the Plant Village dataset. The proposed UnitedModel can extract complementary discriminative features since it combines different CNNs—the UnitedModel averages 99.17% accuracy for validation and 98.57% accuracy for testing. Using the data acquired privately, Liu et al. [39] created a novel identification method for diagnosing grape leaf diseases based on enhanced CNN. In this network, the inception structure uses a dense connection technique to promote reuse and strengthen feature propagation. Based on CNN, a new model was built and trained. It demonstrated 97.22% of accuracy, outperforming GoogLeNet and ResNet-34. Atila et al. [40] developed the EfficientNet DL architecture to classify plant leaf diseases. This model's effectiveness was evaluated compared with other DL models trained on the Plant Village dataset. B4 and B5 models of this architecture had gained accuracy of 99.97% and 99.91%, the highest among all other models. Pixia and Xiangdong [41] presented a technique for identifying cucumber diseases using image processing. In the first stage, preprocessing is applied for smoothing. When a region of interest is extracted from an image dataset, the operation of the corrosion lesion is used (segmentation stage). Conclusively, the rate of disease detection is 96%. A hybrid classification approach was used in the study to identify apple diseases. The maximum accuracy of this method's SVM classifier was 95.94% [42]. This study [43] used a DCNN model with metaheuristic architecture to recognize and classify tomato diseases with 85.9% classification accuracy. Using the dataset, GLDD consisted of a total of 4449 original images. Lu et al. [14] presented a detailed review of DL techniques for plant disease classification, especially CNN. They discussed the importance of DL and compared performance with traditional machine learning techniques. Pal and Kumar [13] presented a CNN-based classification task. They employed the GrabCut approach for disease segmentation and the VGGN pretrained CNN model for the classification task. Faisal et al. [11] presented a dense fusion framework for plant leaf disease classification. They fine-tuned MobileNet and NasNet Mobile architectures for the feature extraction later employed for the classification task. Sharma et al. [12] presented a deeper lightweight model for the multiclass classification of plant leaf diseases. Abd Algani et al. [44] described a DL and optimization algorithm-based classification task framework.

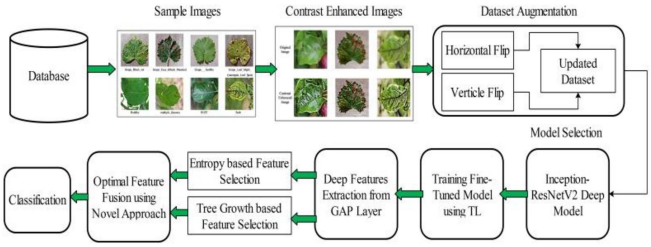

Fig. 1. Proposed flow diagram for fruit disease recognition.

Summary: The experiments previously mentioned concentrated on the traditional and DL characteristics before categorizing them with SVM and CNN classifiers. The preprocessing and augmentation stage, which is crucial when the training data are insufficient, was overlooked in these investigations. In a few research, hybrid strategies have also been offered to increase accuracy. In addition, the optimization process—which is crucial when redundant and irrelevant features are discovered—is not carried out. A novel approach is suggested that uses DL and feature selection to address these issues.

III. Proposed Methodology

The proposed framework is presented in this section. Fig. 1 shows the proposed framework that includes several steps, such as contrast enhancement, augmentation, feature extraction, optimization, and fusion. In the first step, contrast enhancement is applied, and then data augmentation is performed using the horizontal and vertical flip. After this, the Inception-ResNet-V2 deep neural network model is fine-tuned and trained on the augmented dataset using deep transfer learning. Next, entropy-based selection and tree growth optimization are used for best feature selection, which is finally fused using an efficient fusion approach. Finally, the final fused vector is passed to the neural network classifier for the final classification. A brief description of each step is given in the following text.

A. Data Acquisition and Preprocessing

In this section, the procedure for the acquisition and preprocessing of data is described. The collected data are subsequently put through various contrast enhancement and data augmentation methods to enhance image contrast and balance the classes. The following sections discuss the steps of data collection and preprocessing in depth.

1) Data Acquisition: The Plant Village dataset obtained from Kaggle is used in this work [45]. The dataset contains RGB images in jpg format with dimensions 256 × 256 ×3. This dataset consists of many classes of plant diseases from which we are working on two fruit plant diseases: apples and grapes. The selected dataset contains eight classes of six diseases, including apple (scab, rust, and multiple diseases) and grapes (leaf blight, black rot, and black measles). Sample images from the dataset are illustrated in Fig. 2.

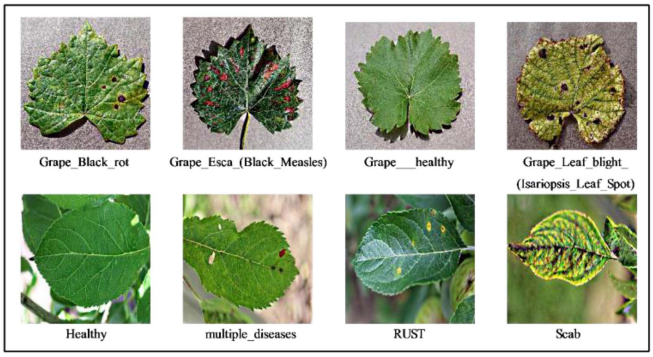

Fig. 2. Sample images of all classes from the selected dataset.

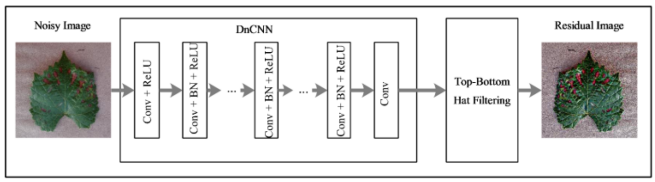

Fig. 3. Hybrid approach for contrast enhancement.

B. Contrast Enhancement

Contrast enhancement enhances the visual characteristics to eliminate noisy elements, such as contrast borders and edge details. It is altering an image to make it more suited for specific use. For example, it makes some image features more prominent or makes certain image positions less ambiguous. It enhances the image's clarity and visual effects and makes the original image easier for computers to process. Finally, it alters the image's pixel intensity to transform it into a better suitable human or machine analysis format. In this work, we proposed a hybrid approach consisting of DnCNN, a CNN method, and the top–bottom process, as illustrated in Fig. 3.

C. DnCNN Model

DnCNN: In the last ten years, the denoiser prior has been extensively used in the renowned plug-and-play framework for the restoration issues of images. The main concept behind this strategy is to add an implicit denoiser prior term to the variational model and then use the splitting technique to decouple this term into a denoising subproblem that commercial denoisers can tackle. More specifically, the following is the simple form of the restoration model using the denoiser prior:

$$\min_{u} \frac{1}{2} \parallel Di - j \parallel_2^2 + \gamma \delta (i) \tag{1}$$

where D is the linear degradation operator and $\delta(\cdot)$ is the implicit denoiser prior, which depicts a lesser value for a sharper and more natural image. By introducing a new variable, the splitting technique will first transform it into an equivalent two-variable problem by introducing a new variable k

$$\min_{i,j} \frac{1}{2} \parallel Di - j \parallel_2^2 + \gamma \delta (k) + \frac{\sigma}{2} \parallel k - i \parallel^2 \tag{2}$$

where σ will increase as the iteration goes up, then the solution can be approached by iterating the following two steps:

$$i^{x+1} = \arg_i \min \frac{1}{2} \parallel Di - j \parallel_2^2 + \frac{\sigma}{2} \parallel k^x - i \parallel^2 \tag{3}$$

$$k^{x+1} = \arg_k \min \frac{1}{2} + \gamma \delta (k)$$

$$\frac{\sigma}{2} \parallel k - i^{x+1} \parallel^2. \tag{4}$$

For k^{x+1}, where the $\gamma(\cdot)$ the implicit prior shows the image's good property, such as sharpness, the solution k can be explained as the denoised result of the given i^{x+1} [46].

Top–Bottom Hat Filtering: The top–bottom hat filtering approach is used to enhance the contrast concerning the lesion area's maximum and minimum intensity values. It is performed on the blue channel $\omega^B(a,b)$, which is extracted from the original image $\omega^B(a,b)$. The blue channel $\omega^B(a,b)$ is defined as follows:

$$\omega^B (a,b) = \frac{\omega^B}{\sum_{i=1}^{3} \omega^i} \tag{5}$$

where i is an index for red, green, and blue channels having values 1, 2, and 3, then the top–bottom hat filtering technique performs to enhance the lesion contrast. The top-hat filter works for foreground objects, while the bottom-hat filter works for background objects. The definition of top–bottom hat filters is given as follows:

$$\omega_{\text{top}} (a,b) = \omega^B (a,b) - \omega^B (a,b) \circ \text{SE} \tag{6}$$

$$\omega_{\text{bot}} (a,b) = \omega^B (a,b) \cdot \text{SE} - \omega^B (a,b) \tag{7}$$

where $\omega_{\text{top}}(a,b)$ defines as a top-hat filtering image, $\omega^B(a,b)$ is the selected blue channel, $\omega_{\text{bot}}(a,b)$ is a bottom-hat image, the opening operation is denoted by $\circ$, $\bullet$ operator denotes the closing operation, and SE represents the structuring element. The SE is selected as mean value of the given image instead of picking a random value. The structuring element is placed at the possible location and compared with neighboring pixels in the input image. The areas that fit adjacent pixels are checked with the structuring element. For top–bottom filtering, SE is initialized as nine, which indicates the window size across the input image is 9×9. Then, relate both images $\omega_{\text{top}}(a,b) = \omega^B$ and $\omega_{\text{bot}}(a,b)$ as follows:

$$\omega_{\text{Re}} (a,b) = \omega_{\text{top}} (a,b) - \omega^B (a,b) \tag{8}$$

$$\omega_F (a,b) = \omega_{\text{Re}} (a,b) - \omega^B (a,b) \tag{9}$$

where $\omega_{\text{Re}}(a,b)$ describes the relationship between the top-hat image and the original image and $\omega_F(a,b)$ defines the final top–bottom filtering image [47]. A few sample images are shown in Fig. 4 after applying this approach.

D. Image Augmentation

In the selected dataset, the number of images in each class is imbalanced, leading to biases and incorrect results for classification. That is why it is crucial to perform image augmentation [48]. We balanced those imbalanced classes using techniques,

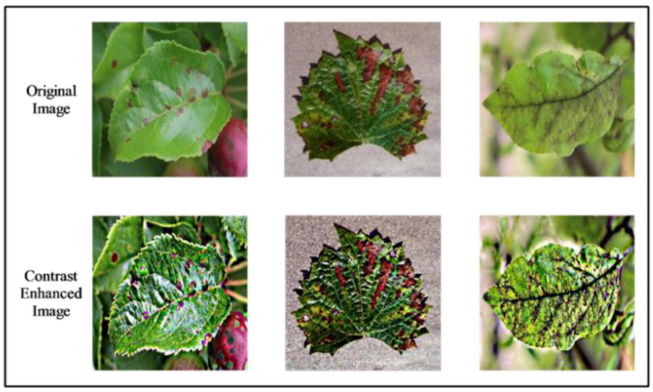

Fig. 4. Original and contrast-enhanced image examples.

TABLE I
NUMBER OF IMAGES BEFORE AND AFTER AUGMENTATION

Dataset	Class Label	Images before Augmentation	Images after Augmentation
grapes	Grape___Black__rot	2360	4720
	Grape___Esca_(Black_Measles)	2766	5532
	Grape___healthy	846	5076
	Grape_Leaf__blight_(Isariopsis_Leaf_Spot)	2152	4304
Seperated_dsApple	Healthy	516	4560
	multiple_diseases	91	4350
	RUST	622	4158
	Scab	592	5000
Total		**9945**	**37 700**

Bold denotes the best accuracy.

such as vertical and horizontal flips. The number of images before and after the augmentation step is shown in Table I. Also, the visual facts are shown in Fig. 5.

E. Convolutional Neural Network

A powerful machine learning technique, convolutional neural networks are assisted versions of neural networks. CNN employs several mathematical models, including gradient descent, regularization, and backpropagation. Convolution, pooling, and fully linked are the three main layers of a straightforward CNN model [49].

Convolutional Layer: Dorj et al. [50] demonstrated that the core part of a CNN is a convolutional layer. In this layer, filtering is performed in specific settings to produce some output from input. The neurons in convolutional layers are arranged in either a rectangular grid or a cubic block. It implies that the filters may

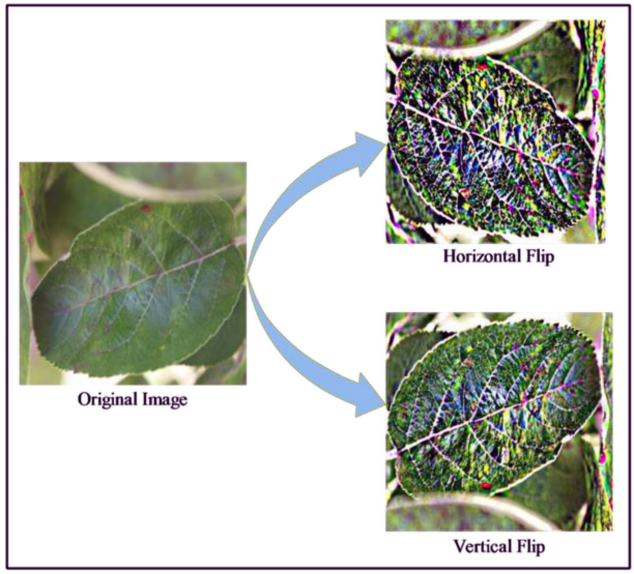

Fig. 5. Resultant images after the augmentation stage.

also consist of a cubic block of neurons or a rectangle grid. The filter is applied from the top left to the bottom right. Everywhere along the filter, a new neuron is acquired after calculating the weighted volume of pixels by $w^t x + b$. The volume of the output neurons is controlled by three types of hyperparameters in the convolutional layer: depth, zero padding, and stride.

The stride specifies executing filters by pixels in the input neuron. Stride specifies the distance or the number of pixels the filter skips or moves over in the input image pixel matrix. While the stride values of two or greater are rare, an enormous stride yields a smaller output when $\text{stride} = 1$, the filters move one pixel. When $\text{stride} = 2$, the filters skip two pixels. When $\text{stride} = 3$, the filters skip three pixels. Zero padding is used when the filter does not fit in the input image. It is the filling process in the input image with zeros for adjusting the size of the input neuron for research purposes. It is typically used when the size of the input neuron in the output neuron has to be maintained. The following equation is used to calculate output neuron size:

$$\text{Output} = \frac{w - k_s + 2p_s}{St} + 1 \tag{10}$$

where w denotes the size of the input neuron, k_s denotes the size of the filter, p_s is the padding size, and St represents the stride. Linear algebraic operations are also utilized in CNN. Let the matrix dimensions be X and Y, while X shows the rows, and Y shows the columns. A 2-D convolutional cube calculates the 2-D convolution with two input matrices. The dimensions of matrix M are (X_M, Y_M), while the dimensions of matrix N are (X_N, Y_N). The convolution equation is given as follows if the cube determines the complete output size

$$C\ (a,b) = \sum_{x=0} (X_M - 1) \sum_{y=0} (Y_M - 1)\ M\,(x,y) * N\,(a-x, b-y) \tag{11}$$

where

$$0 \leq a < X_M + X_N - 1 \text{ and } 0 \leq b < Y_M + Y_N - 1. \quad (12)$$

Max-Pooling Layer: A pooling layer performs the next operation after each convolutional layer. The input neuron size is reduced by using the pooling layers. This layer samples the convolutional layer's tiny rectangular blocks and acquires them to produce a single output from each block. The following equation presents a formulation for the max pooling layer, as defined in the following equation:

$$h_j^k = \max_{\bar{a}\in N(a),\ \bar{b}\in N(b)} h_j^{k-1}\left(\bar{a}, \bar{b}\right). \quad (13)$$

The pooling is produced by applying the filters and sampling the filters in each layer [50].

Fully Connected (FC) Layer: The last layer, the fully linked layer, is created by connecting all preceding neurons. Because it is completely connected from all input neurons to all output neurons, the FC layer typically encourages reducing spatial information. The BP technique is a gradient descent-based algorithm that bases its fitness function on decreasing cross-entropy loss to minimize neural network error [51]

$$P = \sum_{a=1}^{X}\sum_{b=1}^{Y} - s_a^{(b)} \log d_a^{(b)} \quad (14)$$

where X is the number of samples, $s_a = (0, \ldots, 0, 1, \ldots, 1, 0, \ldots, 0)$ is the wanted output vector, and d_a is the output vector of the Yth class that the following formula can attain:

$$d_a^{(b)} = \frac{e^{fa}}{\sum_{a=1}^{Y} e^{fa}}. \quad (15)$$

The weight penalty is adopted for developing the function L to include an η value to improve the values of the weights

$$P = \sum_{a=1}^{X}\sum_{b=1}^{Y} - s_a^{(b)} \log d_a^{(b)} + \frac{1}{2}\varphi \sum_{P}\sum_{Q} e_{p,q}^2 \quad (16)$$

where e_p denotes the weights of connection, the total number of layers is represented by P, and Q shows layer l connections [52].

F. Transfer Learning

Recently, several DL techniques have been introduced for fruit plant disease diagnosis. Transfer learning has been employed to diagnose diseases in fruit plants by transferring acquired information to enhance diagnostic performance [53], [54]. ImageNet dataset is utilized as a source domain represented by S_d and natural images are classified into 1000 classes as the source task S_t.

The augmented Plant Village is the target domain denoted by T_d. To classify images into eight classes is the target task represented by T_t. These four classes of grapes dataset are Grape_Black_rot, Grape_Esca_ (Black_Measles), Grape_healthy, and Grape_Leaf_blight_ (Isariopsis_Leaf_Spot) and the remaining four classes of apple dataset are Healthy, multiple_diseases, RUST, and Scab. The source domain is defined as follows:

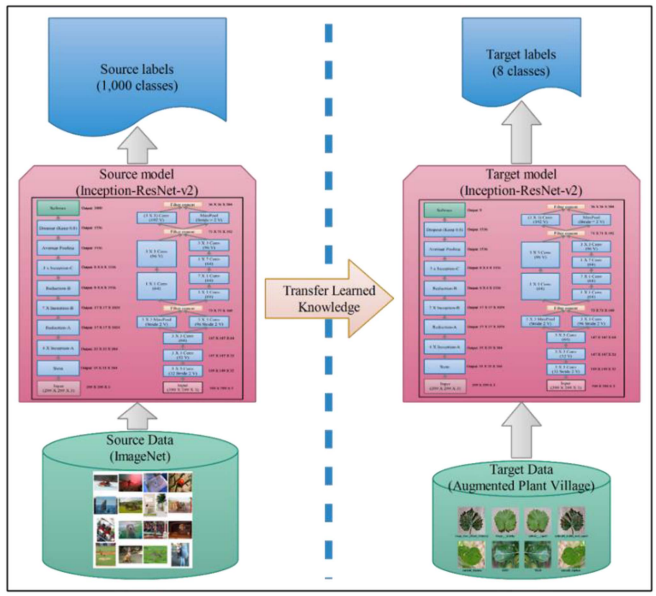

Fig. 6. Process of transfer learning for fruit leaf diseases model training.

$$S_d = \{F_i, M(F_i)\} \quad (17)$$

where F_i is the ImageNet feature space and $M(F_i)$ is the marginal distribution such that $F_i = \{f_1, f_2, \ldots, f_n\}$, $f_i \epsilon F_i$. The target domain is also made up of two components

$$T_d = \{F_c, M(F_c)\} \quad (18)$$

where F_c is the feature space of the augmented Plant Village dataset and $M(F_c)$ is the marginal distribution such that $F_c = \{f_1, f_2, \ldots, f_n\}$, $f_j \epsilon F_c$. Source task S_t, on the other hand, is a tuple comprising two components specified as follows:

$$T_s = \{\rho_i, F_i | F\} = \{\rho, \mu\} \quad (19)$$

where the label space of source data is denoted by ρ and μ is the objective function that learns from label space, such as $(f_i, g_i) f_i \epsilon F_i$, $g_i \epsilon \rho_i$ and feature vector pair.

Transfer learning aims to train a base network using a large source dataset and then apply the learned features to a smaller target dataset. Transfer learning aims to learn the conditional probability distribution of the target domain T_d, about the information gained from a source domain S_d and source task S_t for the given source domain S_d and target domain T_d with their associated source task S_t and target task T_t [55]. The process of transfer learning is illustrated in Fig. 6.

G. Inception-ResNet-V2 Deep Features

Inception-ResNet-v2 is the amalgamation of inception architecture [56] and residual network [57]. The reason for the popularity of inception models is their multibranch architectures. It comprises a collection of filters (1×1, 3×3, 5×5, etc.) combined in each branch using concatenation. The inception module split–transform–merge design is a potent symbolic capability in its dense layers [56]. The residual model is renowned for its ability to train very deep architectures. The hybrid Inception-ResNet-v2

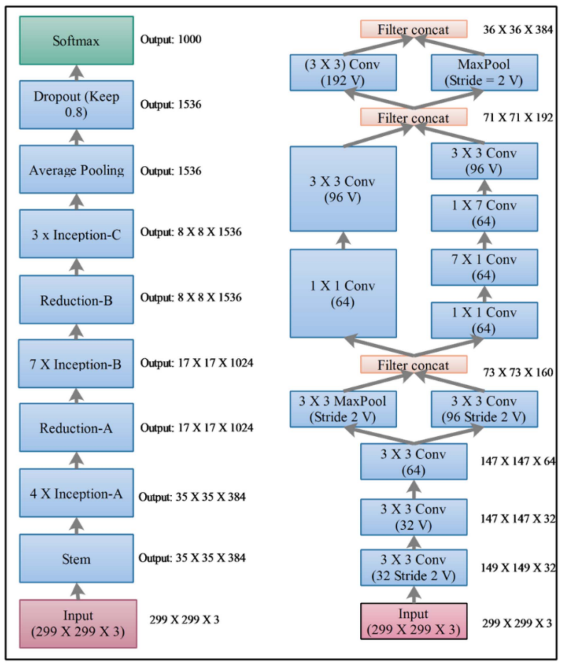

Fig. 7. Inception-ResNet-v2 model architecture [62].

network efficiently utilizes residual connections [58]. Initially, more than a million photos from the ImageNet collection were used to train the Inception-ResNet-v2 model. This 824-layer network can classify data into 1000 different categories. As a result, Inception-ResNet-v2 has picked up lively features that represent a wider variety of images. This network requires input images with dimensions of 299 $\times$ 299 $\times$ 3. The dropout layer's dropout rate is fixed at 0.8. Fig. 7 depicts the architecture of Inception-ResNet-v2. We fine-tuned this model based on this work's last FC layer. The last FC layer includes 1000 object classes, but in our work, the output classes are four for each dataset, as discussed in Section III-A. As a result, we built a new FC layer and removed the prior FC layer, which was then connected to the earlier ones. The categorization of fruit leaf diseases is then performed using a new model created by deep transfer-learning-based training. Section III-C of the document presents the DTL procedure. The newly trained model is later utilized for deep feature extraction. The global average pooling layer has opted for feature extraction. The activation is employed on this layer and 1536 deep features are extracted for each image. The extracted features are analyzed and found some redundant information. Therefore, we employed feature selection techniques to improve the accuracy and reduce the computational time.

H. Best Feature Selection

The selection of the best features from the original set of the feature vector is known as the best feature selection. The primary purpose of best feature selection is to reduce the size of the original feature vector in terms of redundant information and noisy features [59], [60]. This process helps fast execution of a framework and may improve accuracy. In this work, we proposed a two-way selection algorithm called tree growth–entropy selection.

The tree growth optimization algorithm is based on the natural phenomena of competition among trees to acquire food and light. He et al. [57] described that the algorithm's main phases consist of four groups. One among these groups is the best tree group, which is well-grown due to a favorable environment for growth. There is competition for food among this group of trees because the light requirement is met. The second group has competition over the light, so they maintain distance from other trees. In the third replace and delete group, some weak trees are deleted and replaced by new trees. The last group is referred to as reproductive or best trees. These are responsible for increasing their species and reproducing new trees [61]. Mathematically, this algorithm is defined as follows.

1) Calculate the fitness values after randomly generating the initial population of trees on the upper and lower bounds.
2) Locate the best among trees: The best tree serves as the minimal objective function and vice-versa when a minimum optimization problem exists. For example, the TGB j element at repetition j represents the finest version of it.
3) Use the below-mentioned formula to allow S_1 reasonable solutions for local search. (Search locally for numerous solutions. Replace it if the new solution's value is better than the initial response.)

$$X_a^{b+1} = \frac{X_a^b}{\beta} + lX_a^b. \tag{20}$$

The power reduction rate is denoted by β, which means trees are losing energy due to aging, growth, and decreased available food. And l is $U(0,1)$; it instructs the roots of trees to move to absorb food due to the light satisfaction of trees, causing growth at a rate of lX_a^b units.

4) Distancing between the closest optimal solutions using S_2 solutions at various angles α. First, use the following formula to determine the distance between the selected trees and others

$$D_a = \left(\sum_{a=1}^{S_1+S_2} \left(X_{S_2}^b - X_a^b\right)^2\right)^{1/2} \text{ and}$$
$$D_a = \begin{cases} D_a & if\ X_{S_2}^b \neq X_a^b \\ \infty & if\ X_{S_2}^b = X_a^b. \end{cases} \tag{21}$$

Then, select two solutions, $t1$ and $t2$, with minimal D_a and use the following formula for getting a linear combination among the trees ($\rho = U(0,1)$)

$$u = \rho t_1 + (1-\rho)\, t_2. \tag{22}$$

In last, navigating this tree among trees that are adjacent to $\gamma_a = U(0,1)$ angles, using the following formula:

$$X_{S_2}^b = X_{S_2}^b + \gamma_a u. \tag{23}$$

5) Use randomly generated solutions after eradicating S_3 worse solution.

6) Generate novel population (novel population $S = S_1\ S_2 + S_3$).
7) New solution S_4 is created, and a masked operator is used to change every new solution concerning the finest solution from the population S_1randomly. Now, it is added to the novel population (novel population = novel population $+\ S_4$).
8) After sorting the novel population, the total number of population S will initially be the initial population of the upcoming iteration.

Repeat step 2 if the stopping criteria do not meet [61].

This process continues until the number of iterations is completed. In the end, a feature vector is obtained called the best feature vector. However, based on this step's results, we analyzed that some important features were also reduced; hence, we considered another technique, entropy-based selection, for selecting the most numeric value features. The entropy-based selection is defined as follows.

The uncertainty of V, a random variable considering various possibilities among finite values' set, is measured using Shannon's entropy. Let P be the space of all potential probability distributions and let V be a random variable with a finite set of values, including n symbols represented by $\{v_1, v_2, \ldots, v_n\}$. The amount of information associated with v_i is defined as follows if the value of v_i happens with a probability distribution $P(v_i)$ such that $(P(v_i) \geq 0, i = 1, 2, \ldots, n$ and $\sum_{i=1}^{n} (v_i) = 1$ [63]

$$X\ (v_i) = -\log_2 P\,(v_i)\,. \tag{24}$$

Accordingly, the information produced while choosing the symbol v_i is $-\log_2 P(v_i)$ bits for a discrete source. The average quantity of information gathered from n output sources is as follows if the symbol v_i is picked $n \times P(v_i)$ times in n choices

$$-\,n \times P\,(v_1) \log_2 v_1 - n \times P\,(v_2) \log_2 v_2 - \ldots - n \times P\,(v_n) \log_2 v_n. \tag{25}$$

Entropy is defined as a function of the distribution of V, a random variable that depends on probabilities. Because of this, entropy $E(V)$ is defined as follows:

$$E\ (V) = -\sum_{i=1}^{n} P\,(v_i) \log_2 P\,(v_i)\,. \tag{26}$$

This process is continued for all features, and at the end, the features are sorted into descending order, and selects most 50% of them for further processing. In this work, we obtained the entropy-based selected feature vector of dimensions $N \times 750$ and $\mathrm{FV_{Tr}}$ is tree growth optimized vector of dimension $N \times 902$.

I. Feature Fusion

The above-selected features are finally fused using an intelligent fusion approach called serially entropy threshold. Consider, FV_E denotes the entropy-based selected feature vector of dimension $N \times 750$ and $\mathrm{FV_{Tr}}$ is the tree growth optimized vector of dimension $N \times 902$. Then, the serial fusion is defined as follows:

$$\text{Fused} = \begin{pmatrix} \mathrm{FV_E} \\ \mathrm{FV_{Tr}} \end{pmatrix}_{N \times 750 + N \times 902}. \tag{27}$$

The size of the fusion vector is 1650. After that, we computed the entropy value of this fused vector through (16). Based on the entropy value, a threshold function is defined for the final fusion

$$TA = \begin{cases} \widetilde{\text{Fused}}\,(i) & \text{for Fused} \geq E \\ \text{Ignore}, & \text{Elsewhere} \end{cases} \tag{28}$$

where $\widetilde{\text{Fused}}(i)$ is the final fused vector of dimensional $N \times 982$, and E denotes the entropy value of the original fused vector. Finally, the fused features are classified using machine learning classifiers, such as trilayered neural network, bilayered neural network, wide neural network, medium neural network, and many others.

IV. Results and Analysis

This section presents the results of our proposed work in terms of numerical values and sometimes plots. First, the Plant Village dataset was acquired, and two fruits—apples and grapes—were selected. Then, the detail of the dataset is given under Section III-A. After that, features are extracted from the trained model and performed optimization that is finally fused using a serial-based intelligent approach. Then, several experiments were performed, such as results computation on the original deep features, each optimization algorithm, and finally, fusion was applied. Hence, a total of four experiments have been performed for each dataset.

A. Experimental Setup

The experimental process of the proposed framework is presented in this section. For the training and testing, a 50:50 approach has been opted. Several hyperparameters have been initialized in the transfer-learning-based training process, such as a learning rate of 0.0001, momentum of 0.6, minibatch size of 16, epochs of 100, and stochastic gradient descent optimizer. All results are computed through tenfold cross validation. The entire experimental process is conducted on MATLAB2022a using a personal workstation having 16 GB of RAM and an 8 GB Nvidia graphics card.

B. Grapes Dataset Results

Experiment 1: For experiment 1, the augmented grapes dataset has been utilized for training fine-tuned model Inception-ResNet-v2. As a result, deep features are extracted from the average pooling layer. These extracted 1536 deep features are then passed to classifiers, and the results are presented in Table II. The results in this table present that the wide neural network is performing best with an accuracy of 99.9%. The recall and precision rates are 99.9%, which can be verified through the confusion matrix, as illustrated in Fig. 8. The computational time is also noted during the classification process, showing that the medium tree execution time is minimal.

TABLE II
CLASSIFICATION RESULTS OF EXPERIMENT 1 USING GRAPES DATASET

Classifier	Recall Rate (%)	Precision rate (%)	*F*1-Score (%)	FPR	Accuracy (%)	FNR (%)	Time (s)
Trilayered NN	99.8	99.9	99.8	0.0	99.8	0.2	130.5
Bilayered NN	99.8	99.8	99.8	0.0	99.8	0.2	79.1
Wide NN	**99.9**	**99.9**	**99.9**	**0.0**	**99.9**	**0.1**	103.3
Medium NN	99.9	99.9	99.9	0.0	99.9	0.1	73.9
Narrow NN	99.8	99.8	99.8	0.0	99.8	0.2	72.6
Bagged Tree	99.8	99.8	99.8	0.0	99.8	0.3	58.9
Cosine KNN	99.8	99.8	99.8	0.0	99.8	0.2	208.7
Medium KNN	99.8	99.8	99.8	0.0	99.8	0.2	200.4
Cubic SVM	99.9	99.9	99.9	0.0	99.9	0.1	60.4
Quadratic SVM	99.9	99.9	99.9	0.0	99.9	0.2	54
Linear SVM	99.9	99.9	99.9	0.0	99.8	0.2	68.2
Medium Tree	99.5	99.6	99.5	0.0	99.5	0.5	**27**

Bold denotes the best accuracy.

TABLE III
CLASSIFICATION RESULTS OF EXPERIMENT 2 USING GRAPES DATASET

Classifier	Recall Rate (%)	Precision rate (%)	*F*1-Score (%)	FPR	Accuracy (%)	FNR (%)	Time (s)
Trilayered NN	99.9	99.9	99.9	0.0	99.8	0.2	74.0
Bilayered NN	99.8	99.7	99.7	0.0	99.7	0.3	74.2
Wide NN	99.8	99.8	99.8	0.0	99.8	0.2	44.1
Medium NN	99.8	99.8	99.8	0.0	99.8	0.2	32.7
Narrow NN	99.8	99.8	99.8	0.0	99.8	0.2	39.3
Bagged Tree	99.8	99.8	99.8	0.0	99.8	0.2	27.0
Cosine KNN	99.8	99.7	99.7	0.0	99.7	0.3	82.5
Medium KNN	99.7	99.7	99.7	0.0	99.7	0.3	81.5
Cubic SVM	**99.9**	**99.9**	**99.9**	**0.0**	**99.9**	**0.1**	**21.7**
Quadratic SVM	99.9	99.9	99.9	0.0	99.9	0.1	24.1
Linear SVM	99.9	99.9	99.9	0.0	99.8	0.2	23.6
Medium Tree	99.5	99.5	99.5	0.0	99.5	0.6	10.0

Bold denotes the best accuracy.

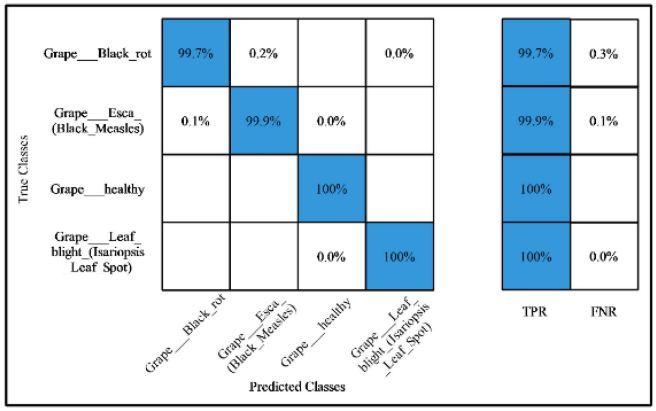

Fig. 8. Confusion matrix of TPR experiment 1 on grapes dataset.

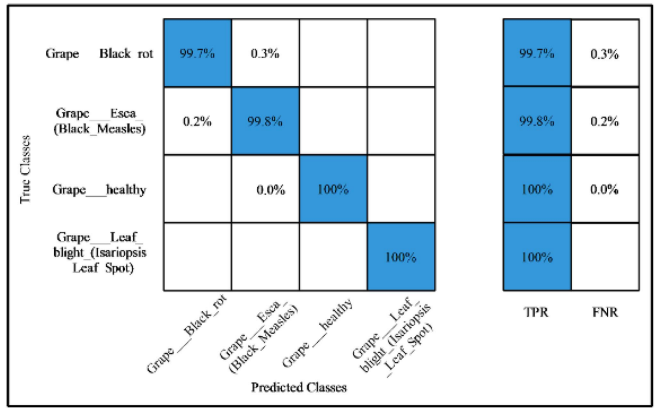

Fig. 9. Confusion matrix of TPR experiment 2 on grapes dataset.

Experiment 2: For experiment 2, entropy-based feature selection is used to select features from initially extracted features. As a result, 750 deep features are selected and then passed to classifiers, and the results are given in Table III. The results in this table show that the cubic SVM is performing best with an accuracy of 99.9%.

The recall and precision rates are 99.9%, which can be verified through the confusion matrix, as illustrated in Fig. 9. The accuracy achieved by other classifiers is also given in this table. Finally, the computational time for each classifier is computed, and the medium tree took a minimum time of 10 s, which was minimized in experiment 1.

Experiment 3: The tree growth optimization technique is used in this experiment to choose deep features from the initially extracted features. Tree growth optimization chose 902 characteristics. The results of using several classifiers with the data generated using this technique are shown in Table IV. The table demonstrates that, among other things, the cubic and quadratic SVM both have the best accuracy of 99.9%. Additionally, both classifiers have 99.9% recall and precision rates, supported by the confusion matrix in Fig. 10. The effectiveness of each additional classifier is also shown in the table below. In addition, the computational time is noted, and it is seen that the time is significantly decreased after the optimization process.

Experiment 4: This experiment is the fusion of tree growth optimization and entropy-based selection algorithms. The features obtained from both algorithms are fused, and the resulting features are 982. These features are then passed to multiple classifiers, and Table V gives the results of all these classifiers. It is evident from the given table that quadratic SVM is performing with a maximum accuracy of 99.9%. The recall and precision rates are 99.9% each. These are verifiable through the confusion matrix, as shown in Fig. 11. The performance of all other classifiers is also given in this table, which shows the improvement. The medium tree computational time is minimal, but compared with experiments 2 and 3, there is a little increase in the time, but

TABLE IV
CLASSIFICATION RESULTS OF EXPERIMENT 3 USING GRAPES DATASET

Classifier	Recall Rate (%)	Precision rate (%)	*F1*-Score (%)	FPR	Accuracy (%)	FNR (%)	Time (s)
Trilayered NN	99.8	99.8	99.8	0.0	99.8	0.2	89.2
Bilayered NN	99.78	99.8	99.8	0.0	99.8	0.2	54.3
Wide NN	99.9	99.9	99.9	0.0	99.8	0.1	60.2
Medium NN	99.9	99.9	99.9	0.0	99.9	0.1	44.5
Narrow NN	75.1	99.9	85.7	0.0	99.8	24.9	44.5
Bagged Tree	99.8	99.8	99.8	0.0	99.8	0.3	38.5
Cosine KNN	99.8	99.8	99.8	0.0	99.7	0.3	118.0
Medium KNN	99.8	99.8	99.8	0.0	99.8	0.2	107.4
Cubic SVM	**99.9**	**99.9**	**99.9**	**0.0**	**99.9**	**0.1**	41.8
Quadratic SVM	**99.9**	**99.9**	**99.9**	**0.0**	**99.9**	**0.1**	39
Linear SVM	99.8	99.8	99.8	0.0	99.8	0.2	39.1
Medium Tree	99.5	99.5	99.5	0.0	99.5	0.5	**13.9**

Bold denotes the best accuracy.

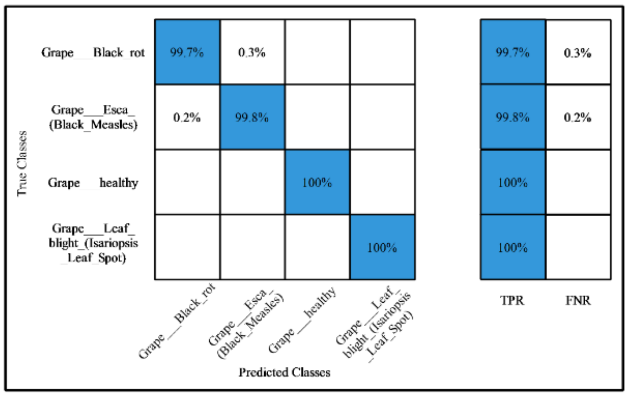

Fig. 10. Confusion matrix of TPR experiment 3 on grapes dataset.

TABLE V
CLASSIFICATION RESULTS OF EXPERIMENT 4 USING GRAPES DATASET

Classifier	Recall Rate (%)	Precision rate (%)	*F1*-Score (%)	FPR	Accuracy (%)	FNR (%)	Time (s)
Trilayered NN	99.8	99.8	99.8	0.0	99.8	0.2	118.9
Bilayered NN	99.8	99.8	99.8	0.0	99.8	0.2	95.3
Wide NN	99.9	99.9	99.9	0.0	99.9	0.1	112.8
Medium NN	99.9	99.9	99.9	0.0	99.8	0.2	70.6
Narrow NN	99.9	99.9	99.9	0.0	99.8	0.2	66.8
Bagged Tree	99.9	99.8	99.8	0.0	99.8	0.2	55.5
Cosine KNN	99.8	99.8	99.8	0.0	99.8	0.2	224.5
Medium KNN	99.8	99.8	99.8	0.0	99.8	0.2	210.6
Cubic SVM	99.9	99.9	99.9	0.0	99.9	0.1	67.9
Quadratic SVM	**99.9**	**99.9**	**99.9**	**0.0**	**99.9**	**0.1**	69.9
Linear SVM	99.9	99.9	99.9	0.0	99.8	0.2	72.5
Medium Tree	99.6	99.6	99.6	0.0	99.5	0.4	**26.2**

Bold denotes the best accuracy.

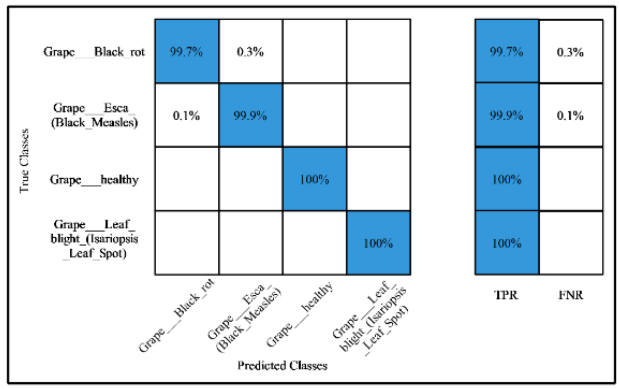

Fig. 11. Confusion matrix of TPR experiment 4 on grapes dataset.

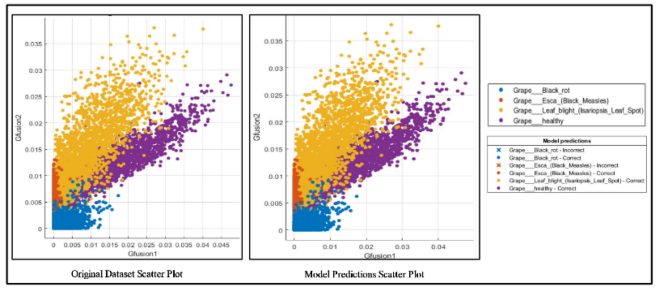

Fig. 12. Scatter plot of original dataset and model predictions for grapes dataset after the fusion.

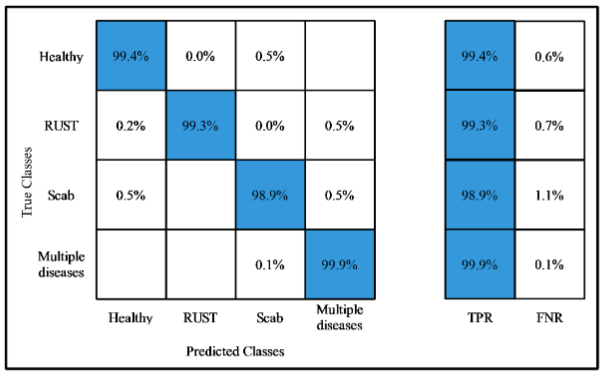

Fig. 13. Confusion matrix of experiment 1 on the apple dataset.

accuracy is improved. In addition, the scatter plot is illustrated in Fig. 12, which shows the original and predicted images after the fusion process.

C. Apple Dataset Results

Experiment 1: The augmented apple dataset has been utilized for training the fine-tuned Inception-ResNet-v2 model. A total of 1536 deep features are extracted from the average pooling layer of the newly fine-tuned model. These extracted deep features are passed to 12 of the same classifiers as in the grape's dataset case. The results are given in Table VI. As in the given table, the cubic SVM performs best with an accuracy of 99.4%. The recall and precision rates are 99.4% each, which can be verified through the confusion matrix, as illustrated in Fig. 13. The computational time for each classifier is also noted during the

TABLE VI
CLASSIFICATION RESULTS OF EXPERIMENT 1 USING AUGMENTED APPLE DATASET

Classifier	Recall Rate (%)	Precision rate (%)	*F*1-Score (%)	FPR	Accuracy (%)	FNR (%)	Time (s)
Trilayered NN	99.3	99.3	99.3	0.00	99.3	0.7	984.5
Bilayered NN	99.3	99.3	99.3	0.00	99.3	0.7	869.4
Wide NN	99.4	99.3	99.3	0.00	99.3	0.7	1037
Medium NN	99.4	99.4	99.4	0.00	99.4	0.7	722.8
Narrow NN	99.3	99.3	99.3	0.00	99.3	0.7	719.1
Bagged Tree	99.4	99.3	99.3	0.00	99.3	0.6	48.2
Cosine KNN	97.5	97.4	97.4	0.01	97.4	2.5	223.0
Medium KNN	97.7	97.7	97.7	0.01	97.7	2.3	138.4
Cubic SVM	**99.4**	**99.4**	**99.4**	**0.00**	**99.4**	**0.6**	52.3
Quadratic SVM	99.3	99.2	99.2	0.00	99.2	0.7	54.0
Linear SVM	98.9	98.7	98.8	0.00	98.9	1.1	55.3
Medium Tree	98.4	98.4	196.7	0.01	98.3	1.6	16.7

Bold denotes the best accuracy.

TABLE VII
CLASSIFICATION RESULTS OF EXPERIMENT 2 USING AUGMENTED APPLE DATASET

Classifier	Recall Rate (%)	Precision rate (%)	*F*1-Score (%)	FPR	Accuracy (%)	FNR (%)	Time (s)
Trilayered NN	99.2	99.2	99.2	0.0	99.2	0.8	473.9
Bilayered NN	99.3	99.3	99.3	0.0	99.3	0.7	369.0
Wide NN	99.3	99.3	99.3	0.0	99.3	0.7	645.5
Medium NN	**99.4**	**99.4**	**99.4**	**0.0**	**99.4**	**0.6**	628.2
Narrow NN	99.3	99.3	99.3	0.0	99.3	0.7	322.5
Bagged Tree	99.3	99.3	99.3	0.0	99.2	0.8	24.4
Cosine KNN	97.8	97.8	97.8	0.0	97.8	2.2	67.7
Medium KNN	98.1	98.1	98.1	0.0	98.1	1.9	64.5
Cubic SVM	**99.4**	**99.4**	**99.4**	**0.0**	**99.4**	**0.6**	25.9
Quadratic SVM	99.3	99.3	99.3	0.0	99.3	0.7	28.3
Linear SVM	99.0	99.0	99.0	0.0	99.0	1.0	30.7
Medium Tree	98.5	98.5	98.5	0.0	98.5	1.5	**9.1**

Bold denotes the best accuracy.

classification process, showing that the medium tree classifier has the minimum time.

Experiment 2: Entropy-based feature selection is used to select features from initially extracted deep features. The selected features are then passed to classifiers, and the results are given in Table VII. According to this table, medium neural networks and cubic SVM outperform others with an accuracy of 99.4%. Furthermore, both classifiers' recall and precision rates are 99.9% verifiable through the confusion matrix, as plotted in Fig. 14. The performance of other classifiers is also given in this table. Moreover, the computational time of each classifier is also noted, and it is observed that the medium tree execution is quicker than the other classifiers. Moreover, this experiment required less time for execution than experiment 1.

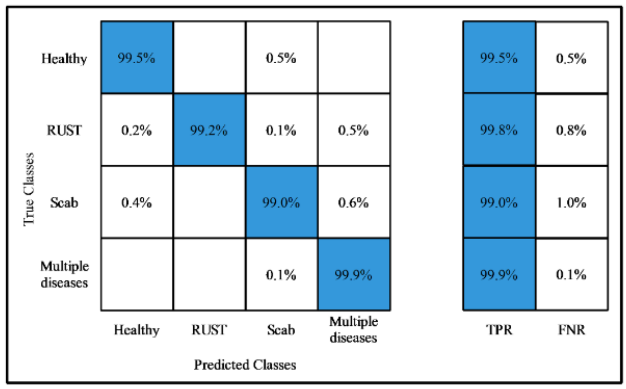

Fig. 14. Confusion matrix of experiment 2 on the apple dataset.

TABLE VIII
CLASSIFICATION RESULTS OF EXPERIMENT 3 USING AUGMENTED APPLE DATASET

Classifier	Recall Rate (%)	Precision rate (%)	*F*1-Score (%)	FPR	Accuracy (%)	FNR (%)	Time (s)
Trilayered NN	99.2	99.2	99.2	0.0	99.2	0.8	768.1
Bilayered NN	99.2	99.2	99.2	0.0	99.2	0.8	735.1
Wide NN	99.3	99.3	99.3	0.0	99.3	0.7	738.1
Medium NN	99.3	99.3	99.3	0.0	99.3	0.7	558.7
Narrow NN	99.2	99.2	99.2	0.0	99.2	0.8	540.8
Bagged Tree	99.2	99.3	99.2	0.0	99.2	0.8	34.6
Cosine KNN	97.2	97.2	97.2	0.0	97.2	2.8	101.0
Medium KNN	97.7	97.7	97.7	0.0	97.7	2.3	98.7
Cubic SVM	**99.4**	**99.4**	**99.4**	**0.0**	**99.4**	**0.6**	340.8
Quadratic SVM	97.2	99.2	98.2	0.0	99.2	2.8	51.2
Linear SVM	99.0	99.0	99.0	0.0	99.0	1.0	51.7
Medium Tree	98.4	98.4	98.4	0.0	98.4	1.6	**10.9**

Bold denotes the best accuracy.

Experiment 3: The improved tree growth optimization algorithm is applied to select the best deep features in this experiment. A total of 945 deep features are selected through this algorithm. Then, these features are passed to various classifiers. The performance of each classifier is presented in Table VIII. This table presents that the cubic SVM outperforms others with an accuracy of 99.4%. The recall rate and precision rate are 99.4%, which can be verified with the help of the confusion matrix, as shown in Fig. 15. The computational time for each classifier is also noted, and the medium tree classifier consumes less time than other classifiers. Moreover, it is also noted that the accuracy of this experiment is consistent, but time is reduced, which is the main strength of this work.

Experiment 4: The suggested intelligent fusion method is used in this experiment to perform a fusion. This method allows for enhanced tree growth optimization, and entropy-based selected characteristics that return 980 are merged. Following that, classification is conducted on these fused features, and Table IX provides the results of each classifier. This table explains how well medium MM performs, with a maximum accuracy of 99.4%. The confusion matrix in Fig. 16 provides evidence that the recall and precision rates are each 99.4%. The computing time for the medium tree classifier is less than that of experiments

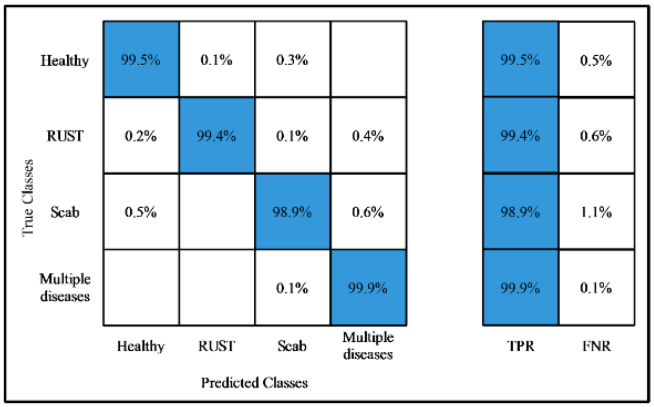

Fig. 15. Confusion matrix of experiment 3 on the apple dataset.

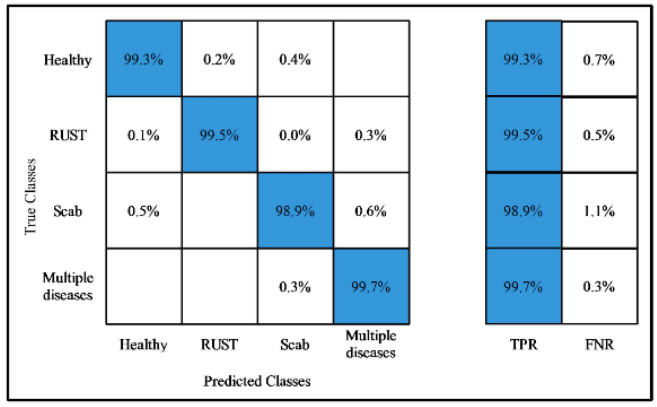

Fig. 16. Confusion matrix of experiment 4 on the apple dataset.

TABLE IX
CLASSIFICATION RESULTS OF EXPERIMENT 4 USING AUGMENTED APPLE DATASET

Classifier	Recall Rate (%)	Precision rate (%)	*F*1-Score (%)	FPR	Accuracy (%)	FNR (%)	Time (s)
Trilayered NN	99.3	99.3	99.3	0.0	99.3	0.7	879.2
Bilayered NN	99.3	99.3	99.3	0.0	99.3	0.7	811.2
Wide NN	99.3	99.3	99.3	0.0	99.3	0.7	1002
Medium NN	**99.4**	**99.4**	**99.4**	**0.0**	**99.4**	**0.6**	685.1
Narrow NN	99.2	99.2	99.2	0.0	99.2	0.8	681.8
Bagged Tree	99.3	99.3	99.3	0.0	99.3	0.7	46.3
Cosine KNN	97.7	97.6	97.6	0.0	97.6	2.3	159.9
Medium KNN	97.8	97.9	97.8	0.0	97.8	2.2	155.7
Cubic SVM	99.4	99.3	99.3	0.0	99.3	0.7	288.1
Quadratic SVM	99.2	99.3	99.2	0.0	99.2	0.8	60.3
Linear SVM	99.1	99.1	99.1	0.0	99.0	1.0	62.1
Medium Tree	98.4	98.4	98.4	0.0	98.4	1.6	**18.8**

Bold denotes the best accuracy.

TABLE X
COMPARISON OF THE PROPOSED FRAMEWORK ACCURACY ON SELECTED DATASETS WITH OTHER STATE-OF-THE-ART DL TECHNIQUES

Method	Dataset	Accuracy (%)	Time (s)
VGG16+ Proposed Optimization	Augmented Apple	96.5	17.5
VGG19+ Proposed Optimization		96.9	21.8
Resnet50+ Proposed Optimization		98.1	26.33
ResNet18+		97.6	20.3
Proposed Optimization			
Proposed		**99.4**	**9.1**
VGG16+Proposed Optimization	Augmented Grapes Dataset	97.2	21.4
VGG19+ Proposed Optimization		98.1	24.8
Resnet50+ Proposed Optimization		98.8	31.5
ResNet18+ Proposed Optimization		98.5	27.2
Proposed		**99.9**	**10.0**

Bold denotes the best accuracy.

2 and 3, but it is more. Overall, this experiment's precision is higher than others using the apple dataset.

D. Discussion

A detailed discussion of the proposed results has been conducted in this section. As shown in Fig. 1, the proposed framework contains a few important middle steps, such as entropy-based feature selection, improved tree growth-based selection, and fusion of both optimal feature vectors. Tables II–V presented the results of the grapes dataset with the best accuracy confusion matrix. Tables VI–IX present the results of the proposed framework on the apple dataset. Moreover, a confusion matrix is illustrated for each experiment. These tables present the results of the augmented datasets. Results presented in these tables show that the accuracy is improved after the fusion process. In addition, the entropy-based selection technique's computational time is less than the improved tree growth optimization algorithm. In addition, the fusion process improved the precision rate, FPR, and accuracy, but the computational time is jumped, which is a drawback of this step.

To analyze the impact of data augmentation, we computed the results on the original dataset and then compared them with an augmented dataset, as shown in Fig. 16. This figure shows that the difference between the accuracy of original (nonaugmented) and augmented datasets is almost 3%. The accuracy is improved after the augmentation process, as seen in Table X.

Fig. 17 illustrates the obtained accuracy for both datasets on those features, which are ignored during the fusion process. In this figure, it is clearly shown a difference in the accuracy of best fused features (see Fig. 16). There is a 5%–6% difference is noted among Figs. 16 and 17, whereas the augmented dataset has been employed for the evaluation process, see Fig. 18.

In the end, a brief comparison of the proposed framework is conducted with several DL techniques, such as VGG16 [64], VGG19 [65], ResNet50 [66], and ResNet18 [67]. A comparison is given in Table IX, showing that the proposed framework accuracy is improved than the recent techniques. Also, the computational time of the proposed framework is less than the other listed methods.

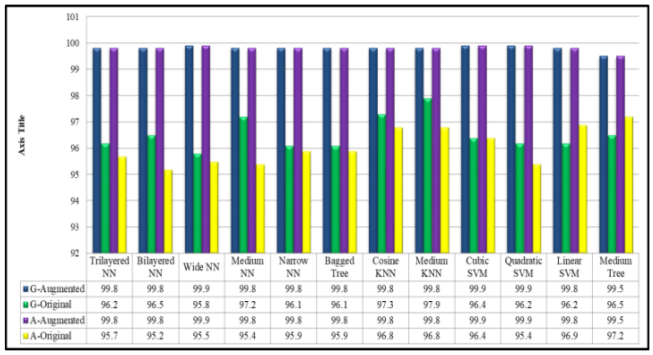

Fig. 17. Comparison in terms of accuracy among original and augmented datasets.

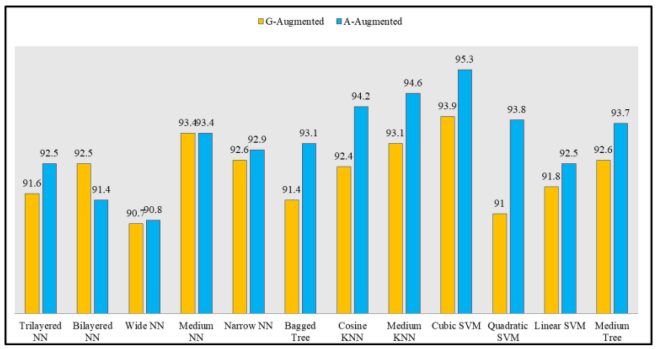

Fig. 18. Comparison among final selected features and ignored features during the final fusion step.

This table presents that ResNet50 gives better results than VGG16, VGG19, and ResNet18; however, the proposed framework shows improved accuracy. In addition, the proposed framework is also tested on the original dataset (before augmentation), and the results are shown in Table XI.

This table would present the degradation in accuracy if we did not perform the augmentation step. Hence, the augmentation process improved the accuracy of the proposed framework.

V. Conclusion

This work presents an automated DL and best feature selection-based approach for fruit leaf disease classification. The proposed method included several steps: contrast enhancement, data augmentation, DL-based feature extraction, feature selection, feature fusion, and classification. The original dataset images are not in enough size to train a DL model; therefore, we performed data augmentation. A pretrained DL model, Inception-ResNet-v2, extracts deep features. The best features are then chosen using tree growth optimization and entropy-based selection, which are later fused using an intelligent fusion approach. These selected features are then passed to 12 different classifiers and the classification accuracy. In comparison with some other DL methods, it is noted that the proposed method shows an improved accuracy of 99.9%. We concluded from the results that using appropriate feature selection techniques increases classification accuracy while decreasing computational time.

TABLE XI
Comparison of Proposed Framework Accuracy on Selected Datasets (Before Data Augmentation) With Other State-of-the-Art DL Techniques

Method	Dataset	Accuracy (%)	Time (s)
VGG16+Proposed Optimization	Before Augmentation Apple Dataset	92.2	13.2
VGG19+ Proposed Optimization		93.7	17.4
Resnet50+ Proposed Optimization		94.7	20.13
ResNet18+ Proposed Optimization		94.4	17.0
Proposed		**97.2**	**6.8**
VGG16+Proposed Optimization	Before Augmentation Grapes Dataset	93.5	16.3
VGG19+ Proposed Optimization		94.8	11.4
Resnet50+ Proposed Optimization		95.2	22.3
ResNet18+ Proposed Optimization		95.1	21.6
Proposed		**97.9**	**7.4**

Bold denotes the best accuracy.

In addition, the augmentation method improves accuracy in comparison with the accuracy of the original dataset. A drawback of this approach is that the fusion procedure lengthens the computing time. Future proposals for intelligent-based fusion techniques will use more effective optimization algorithms to improve the feature data. In addition, the network based on encoder–decoders will be considered for feature extraction.

References

[1] M. A. Khan, T. Akram, M. Sharif, and T. Saba, "Fruits diseases classification: Exploiting a hierarchical framework for deep features fusion and selection," *Multimedia Tools Appl.*, vol. 79, pp. 25763–25783, 2020.

[2] F. D. S. Juraev, "Problems of informatization of management of agricultural industry and modeling of agriconomic system in a market economy," *Amer. J. Appl. Sci.*, vol. 3, pp. 49–54, 2022.

[3] A. Adeel et al., "Diagnosis and recognition of grape leaf diseases: An automated system based on a novel saliency approach and canonical correlation analysis based multiple features fusion," *Sustain. Comput., Inform. Syst.*, vol. 24, 2019, Art. no. 100349.

[4] V. Singh and A. K. Misra, "Detection of plant leaf diseases using image segmentation and soft computing techniques," *Inf. Process. Agriculture*, vol. 4, pp. 41–49, 2017.

[5] V. Pooja, R. Das, and V. Kanchana, "Identification of plant leaf diseases using image processing techniques," in *Proc. IEEE Technol. Innov. ICT Agriculture Rural Develop.*, 2017, pp. 130–133.

[6] M. A. Khan et al., "CCDF: Automatic system for segmentation and recognition of fruit crops diseases based on correlation coefficient and deep CNN features," *Comput. Electron. Agriculture*, vol. 155, pp. 220–236, 2018.

[7] D. I. Patrício and R. Rieder, "Computer vision and artificial intelligence in precision agriculture for grain crops: A systematic review," *Comput. Electron. Agriculture*, vol. 153, pp. 69–81, 2018.

[8] L. C. Ngugi, M. Abelwahab, and M. Abo-Zahhad, "Recent advances in image processing techniques for automated leaf pest and disease recognition–A review," *Inf. Process. Agriculture*, vol. 8, pp. 27–51, 2021.

[9] K. M. Hosny, W. M. El-Hady, F. M. Samy, E. Vrochidou, and G. A. Papakostas, "Multi-class classification of plant leaf diseases using feature fusion of deep convolutional neural network and local binary pattern," *IEEE Access*, vol. 11, pp. 62307–62317, 2023.

[10] A. Ahmad, D. Saraswat, and A. El Gamal, "A survey on using deep learning techniques for plant disease diagnosis and recommendations for development of appropriate tools," *Smart Agricultural Technol.*, vol. 3, 2023, Art. no. 100083.

[11] M. Faisal, J.-S. Leu, C. Avian, S. W. Prakosa, and M. Köppen, "DFNet: Dense fusion convolution neural network for plant leaf disease classification," *Agronomy J.*, to be published, doi: 10.1002/agj2.21341.

[12] V. Sharma, A. K. Tripathi, and H. Mittal, "DLMC-Net: Deeper lightweight multi-class classification model for plant leaf disease detection," *Ecol. Inform.*, vol. 75, 2023, Art. no. 102025.

[13] A. Pal and V. Kumar, "AgriDet: Plant leaf disease severity classification using agriculture detection framework," *Eng. Appl. Artif. Intell.*, vol. 119, 2023, Art. no. 105754.

[14] J. Lu, L. Tan, and H. Jiang, "Review on convolutional neural network (CNN) applied to plant leaf disease classification," *Agriculture*, vol. 11, 2021, Art. no. 707.

[15] A. Almadhor, H. T. Rauf, M. I. U. Lali, R. Damaševičius, B. Alouffi, and A. Alharbi, "AI-driven framework for recognition of guava plant diseases through machine learning from DSLR camera sensor based high resolution imagery," *Sensors*, vol. 21, 2021, Art. no. 3830.

[16] P. Tiwari, A. Lakhan, R. H. Jhaveri, and T.-M. Gronli, "Consumer-centric internet of medical things for cyborg applications based on federated reinforcement learning," *IEEE Trans. Consum. Electron.*, to be published, doi: 10.1109/TCE.2023.3242375.

[17] A. Khamparia, D. Gupta, V. H. C. de Albuquerque, A. K. Sangaiah, and R. H. Jhaveri, "Internet of health things-driven deep learning system for detection and classification of cervical cells using transfer learning," *J. Supercomput.*, vol. 76, pp. 8590–8608, 2020.

[18] F. Saeed, M. A. Khan, M. Sharif, M. Mittal, L. M. Goyal, and S. Roy, "Deep neural network features fusion and selection based on PLS regression with an application for crops diseases classification," *Appl. Soft Comput.*, vol. 103, 2021, Art. no. 107164.

[19] Y. Li, J. Nie, and X. Chao, "Do we really need deep CNN for plant diseases identification?," *Comput. Electron. Agriculture*, vol. 178, 2020, Art. no. 105803.

[20] M. Sharif, M. A. Khan, Z. Iqbal, M. F. Azam, M. I. U. Lali, and M. Y. Javed, "Detection and classification of citrus diseases in agriculture based on optimized weighted segmentation and feature selection," *Comput. Electron. Agriculture*, vol. 150, pp. 220–234, 2018.

[21] J. S. H. Al-bayati and B. B. Üstündağ, "Evolutionary feature optimization for plant leaf disease detection by deep neural networks," *Int. J. Comput. Intell. Syst.*, vol. 13, 2020, Art. no. 12.

[22] S. Rehman et al., "Fruit leaf diseases classification: A hierarchical deep learning framework," *Comput., Mater. Continua*, vol. 75, pp. 1179–1194, 2023.

[23] H. Wang, S. Shang, D. Wang, X. He, K. Feng, and H. Zhu, "Plant disease detection and classification method based on the optimized lightweight YOLOv5 model," *Agriculture*, vol. 12, 2022, Art. no. 931.

[24] G. Chandrashekar and F. Sahin, "A survey on feature selection methods," *Comput. Elect. Eng.*, vol. 40, pp. 16–28, 2014.

[25] S. Rehman et al., "A framework of deep optimal features selection for apple leaf diseases recognition," *Comput., Mater. Continua*, vol. 75, pp. 697–714, 2023.

[26] J. Zhu, A. Wu, X. Wang, and H. Zhang, "Identification of grape diseases using image analysis and BP neural networks," *Multimedia Tools Appl.*, vol. 79, pp. 14539–14551, 2020.

[27] S. Sladojevic, M. Arsenovic, A. Anderla, D. Culibrk, and D. Stefanovic, "Deep neural networks based recognition of plant diseases by leaf image classification," *Comput. Intell. Neurosci.*, vol. 2016, 2016, Art. no. 3289801.

[28] M. Jhuria, A. Kumar, and R. Borse, "Image processing for smart farming: Detection of disease and fruit grading," in *Proc. IEEE 2nd Int. Conf. Image Inf. Process.*, 2013, pp. 521–526.

[29] L. S. P. Annabel, T. Annapoorani, and P. Deepalakshmi, "Machine learning for plant leaf disease detection and classification—A review," in *Proc. Int. Conf. Commun. Signal Process.*, 2019, pp. 538–542.

[30] M. Prabu and B. J. Chelliah, "Mango leaf disease identification and classification using a CNN architecture optimized by crossover-based Levy flight distribution algorithm," *Neural Comput. Appl.*, vol. 34, pp. 7311–7324, 2022.

[31] Z. Chen et al., "Plant disease recognition model based on improved YOLOv5," *Agronomy*, vol. 12, 2022, Art. no. 365.

[32] P. Kaur et al., "Recognition of leaf disease using hybrid convolutional neural network by applying feature reduction," *Sensors*, vol. 22, 2022, Art. no. 575.

[33] Z. U. Rehman et al., "Recognizing apple leaf diseases using a novel parallel real-time processing framework based on MASK RCNN and transfer learning: An application for smart agriculture," *IET Image Process.*, vol. 15, pp. 2157–2168, 2021.

[34] S. Chakraborty, S. Paul, and M. Rahat-uz-Zaman, "Prediction of apple leaf diseases using multiclass support vector machine," in *Proc. 2nd Int. Conf. Robot., Elect. Signal Process. Techn.*, 2021, pp. 147–151.

[35] A. Alsayed, A. Alsabei, and M. Arif, "Classification of apple tree leaves diseases using deep learning methods," *Int. J. Comput. Sci. Netw. Secur.*, vol. 21, pp. 324–330, 2021.

[36] J. Di and Q. Li, "A method of detecting apple leaf diseases based on improved convolutional neural network," *PLoS One*, vol. 17, 2022, Art. no. e0262629.

[37] H. Yu et al., "Apple leaf disease recognition method with improved residual network," *Multimedia Tools Appl.*, vol. 81, pp. 7759–7782, 2022.

[38] M. Ji, L. Zhang, and Q. Wu, "Automatic grape leaf diseases identification via UnitedModel based on multiple convolutional neural networks," *Inf. Process. Agriculture*, vol. 7, pp. 418–426, 2020.

[39] B. Liu, Z. Ding, L. Tian, D. He, S. Li, and H. Wang, "Grape leaf disease identification using improved deep convolutional neural networks," *Front. Plant Sci.*, vol. 11, 2020, Art. no. 1082.

[40] Ü. Atila, M. Uçar, K. Akyol, and E. Uçar, "Plant leaf disease classification using EfficientNet deep learning model," *Ecol. Inform.*, vol. 61, 2021, Art. no. 101182.

[41] D. Pixia and W. Xiangdong, "Recognition of greenhouse cucumber disease based on image processing technology," *Open J. Appl. Sci.*, vol. 3, 2013, Art. no. 27.

[42] S. R. Dubey and A. S. Jalal, "Apple disease classification using color, texture and shape features from images," *Signal, Image Video Process.*, vol. 10, pp. 819–826, 2016.

[43] A. Fuentes, S. Yoon, S. C. Kim, and D. S. Park, "A robust deep-learning-based detector for real-time tomato plant diseases and pests recognition," *Sensors*, vol. 17, 2017, Art. no. 2022.

[44] Y. M. Abd Algani, O. J. M. Caro, L. M. R. Bravo, C. Kaur, M. S. Al Ansari, and B. K. Bala, "Leaf disease identification and classification using optimized deep learning," *Meas., Sensors*, vol. 25, 2023, Art. no. 100643.

[45] S. P. Mohanty, D. P. Hughes, and M. Salathé, "Using deep learning for image-based plant disease detection," *Front. Plant Sci.*, vol. 7, 2016, Art. no. 1419.

[46] Y. Fang, H. Zhang, H. S. Wong, and T. Zeng, "A robust non-blind deblurring method using deep denoiser prior," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 734–743.

[47] M. A. Khan, M. Rashid, M. Sharif, K. Javed, and T. Akram, "Classification of gastrointestinal diseases of stomach from WCE using improved saliency-based method and discriminant features selection," *Multimedia Tools Appl.*, vol. 78, pp. 27743–27770, 2019.

[48] O. O. Abayomi-Alli, R. Damaševičius, A. Qazi, M. Adedoyin-Olowe, and S. Misra, "Data augmentation and deep learning methods in sound classification: A systematic review," *Electronics*, vol. 11, 2022, Art. no. 3795.

[49] S. K. Addagarla, G. K. Chakravarthi, and P. Anitha, "Real time multi-scale facial mask detection and classification using deep transfer learning techniques," *Int. J. Adv. Trends Comput. Sci. Eng.*, vol. 9, pp. 4402–4408, 2020.

[50] U.-O. Dorj, K.-K. Lee, J.-Y. Choi, and M. Lee, "The skin cancer classification using deep convolutional neural network," *Multimedia Tools Appl.*, vol. 77, pp. 9909–9924, 2018.

[51] B. Van Merriënboer et al., "Blocks and fuel: Frameworks for deep learning," 2015, *arXiv:1506.00619*.

[52] N. Zhang, Y.-X. Cai, Y.-Y. Wang, Y.-T. Tian, X.-L. Wang, and B. Badami, "Skin cancer diagnosis based on optimized convolutional neural network," *Artif. Intell. Med.*, vol. 102, 2020, Art. no. 101756.

[53] A. Saeed, A. Abdel-Aziz, A. Mossad, M. A. Abdelhamid, A. Y. Alkhaled, and M. Mayhoub, "Smart detection of tomato leaf diseases using transfer learning-based convolutional neural networks," *Agriculture*, vol. 13, 2023, Art. no. 139.

[54] M. Hassam et al., "A single stream modified MobileNet V2 and whale controlled entropy based optimization framework for citrus fruit diseases recognition," *IEEE Access*, vol. 10, pp. 91828–91839, 2022.

[55] A. Rehman, S. Naz, A. Khan, A. Zaib, and I. Razzak, "Improving coronavirus (COVID-19) diagnosis using deep transfer learning," in *Proc. Int. Conf. Inf. Technol. Appl.*, 2022, pp. 23–37.
[56] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 2818–2826.
[57] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.
[58] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-Resnet and the impact of residual connections on learning," in *Proc. 31st AAAI Conf. Artif. Intell.*, 2017, pp. 4278–4284.
[59] S. Abbas et al., "Crops leaf diseases recognition: A framework of optimum deep learning features," *Comput., Mater. Continua*, vol. 74, pp. 1139–1159, 2023.
[60] M. A. Khan et al., "Cucumber leaf diseases recognition using multi level deep entropy-ELM feature selection," *Appl. Sci.*, vol. 12, 2022, Art. no. 593.
[61] A. Cheraghalipour, M. Hajiaghaei-Keshteli, and M. M. Paydar, "Tree growth algorithm (TGA): A novel approach for solving optimization problems," *Eng. Appl. Artif. Intell.*, vol. 72, pp. 393–414, 2018.
[62] F. Baldassarre, D. G. Morín, and L. Rodés-Guirao, "Deep koalarization: Image colorization using CNNs and inception-resnet-v2," 2017, *arXiv:1712.03400*.
[63] T.-W. Chang, Y.-P. Huang, and F. E. Sandnes, "Efficient entropy-based features selection for image retrieval," in *Proc. IEEE Int. Conf. Syst., Man Cybern.*, 2009, pp. 2941–2946.
[64] H. Qassim, A. Verma, and D. Feinzimer, "Compressed residual-VGG16 CNN model for big data places image recognition," in *Proc. IEEE 8th Annu. Comput. Commun. Workshop Conf.*, 2018, pp. 169–175.
[65] V. Rajinikanth, A. N. Joseph Raj, K. P. Thanaraj, and G. R. Naik, "A customized VGG19 network with concatenation of deep and handcrafted features for brain tumor detection," *Appl. Sci.*, vol. 10, 2020, Art. no. 3429.
[66] I. Z. Mukti and D. Biswas, "Transfer learning based plant diseases detection using ResNet50," in *Proc. 4th Int. Conf. Elect. Inf. Commun. Technol.*, 2019, pp. 1–6.
[67] Z. Chen et al., "ResNet18DNN: Prediction approach of drug-induced liver injury by deep neural network with ResNet18," *Brief. Bioinform.*, vol. 23, 2022, Art. no. bbab503.

Unber Zahra received the bachelor's and master's degrees from the Department of Computer Science, HITEC University, Taxila, Pakistan, in 2018 and 2022, respectively.

Her research interests include artificial intelligence and optimization for agriculture data analysis.

Muhammad Attique Khan (Member, IEEE) received the master's and Ph.D. degrees in human activity recognition for the application of video surveillance and skin lesion classification using deep learning from COMSATS University Islamabad, Lahore, Pakistan, in 2018 and 2022, respectively.

He is currently an Assistant Professor with the Department of Computer Science, HITEC University, Taxila, Pakistan. He has more than 190 publications that have more than 6500 citations and impact factor more than 600 with H-index 50. His research interests include medical imaging, COVID19, MRI analysis, video surveillance, human gait recognition, and agriculture plants. He is a reviewer of several reputed journals, such as IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, IEEE TRANSACTIONS ON NEURAL NETWORKS, *Pattern Recognition Letters*, *Multimedia Tools and Application*, *Computers and Electronics in Agriculture*, *IET Image Processing*, *Biomedical Signal Processing Control*, *IET Computer Vision*, *EURASIP Journal of Image and Video Processing*, *IEEE Access*, *Sensors* (MDPI), *Electronics* (MDPI), *Applied Sciences* (MDPI), *Diagnostics* (MDPI), and *Cancers* (MDPI).

Majed Alhaisoni received the Ph.D. degrees from University of Hail, Ha'il, Saudi Arabia, in 2017. He is currently a Professor of computer science with the University of Ha'il Kingdom, Ha'il, Saudi Arabia. His research interests include artificial intelligence and optimization. He authored or coauthored more than 50 high impact factor papers from last 3 years. He is also a reviewer of many journals, such as *Multimedia Systems*, *Multimedia Tools and Applications*, IEEE TRANSACTION OF PATTEN ANALYSIS AND MACHINE INTELLIGENCE, and few others.

Areej Alasiry received the B.Sc. degree in information systems from King Khalid University, Abha, Saudi Arabia, in 2016, the M.Sc.(Hons.) degree in advanced information systems and the Ph.D. degree in computer science and information systems from Birkbeck College, University of London, London, U.K., in 2010 and 2015, respectively.

She is currently an Assistant Professor with the College of Computer Science, King Khalid University. She also holds the position of the College Vice Dean for Graduate Studies and Scientific Research. Her main research interests include machine learning and data science.

Mehrez Marzougui received the bachelor's and master's degrees in 2014 and 2016, respectively and the Ph.D. degree in 2021. He is an Assistant Professor with Information Systems Department, College of Computer Science, King Khalid University, Abha, Saudi Arabia. His research interests include medical data analysis using artificial intelligence and deep learning.

Anum Masood received the B.Sc. and M.Sc. degrees in computer science from COMSATS University Islamabad, Islamabad, Pakistan, in 2012 and 2014, respectively, and the Ph.D. degree in computer science and engineering from Shanghai Jiao Tong University, Shanghai, China, in 2019.

She was a Lecturer with the Department of Computer Science, COMSATS University Islamabad, from 2014 to 2020. She is currently a Postdoctoral Researcher with the Norwegian University of Science and Technology, Trondheim, Norway, and affiliated with PET Centre, St. Olav's Hospital, Trondheim, Norway. She also worked as a Visiting Researcher with the Institute of Neuroscience and Medicine, Forschungszentrum Jülich, Institute for Cardiogenetics, University of Luebeck, Germany and Liverpool John Moores University, U.K. Her research interests include medical image analysis, automated cancer detection, machine learning, and image processing.