Published by :
https://www.ijert.org/
An International Peer-Reviewed Journal

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 07, NCIRTAI - 2026

A Comprehensive Deep Learning System for Classifying Fruit Diseases

Ms. K. Sushmeena [1]
[1] Assistant Professor, Department of CSE, Sri Bharathi Engineering College For Women, Pudukkottai
Email Id:sushmeena.nk@gmail.com

Abstract

For the past five years, agriculture has been a significant area of research in image processing. Diseases have an impact on the number and quality of fruits, which can cause economic disruptions. Numerous automated methods for identifying and detecting fruit illnesses have been developed. Nevertheless, there are still certain problems that need to be resolved, like superfluous features and the dimensionality of feature vectors, which lengthen the system's processing time. Here, we suggest an integrated deep learning architecture for fruit disease classification. We take into account seven different fruit varieties: apples, cherries, blueberries, grapes, peaches, citrus, and strawberries. Several crucial phases are included in the suggested strategy. Two distinct feature kinds are recovered once the data is first increased. Texture and color features, or classical features, are extracted in the first feature type.A pretrained model is used in the second type to obtain deep learning characteristics.Through transfer learning, the pretrained model is reused.The maximal mean value of the serial technique is then used to combine the two types of information. A harmonic threshold-based genetic algorithm is then employed to optimize the resulting fused vector.Ultimately, several classifiers are used to classify the chosen characteristics. The PlantVillage dataset is evaluated, and a 99% accuracy rate is attained. The suggested approach is preferable when compared to more modern methods.

Keywords: Deep Learning, Classical Features, Feature Fusion, Data Augmentation, Fruit Illnesses, And Feature Selection

1.Introduction

A significant area of study in image processing and computer vision is agricultural imaging [1]. Any nation's economic progress is greatly aided by fruit trees [2, 3]. They support local jobs in addition to supplying food and raw materials. The Creative Commons Attribution 4.0 International License that governs this work allows for free use, distribution, and reproduction in any medium as long as the original work is appropriately populace [4]. Citrus fruits, apples, grapes, and peaches are among the fruit plants that significantly contribute to production. Because diseases have an impact on fruit output, and a nation's general economy is obviously impacted when fruit productivity declines. In order to prevent significant losses, it is crucial to identify these illnesses early on. Downy, greening, canker, and black spots are the most common citrus fruit diseases. Frog eye spots, cedar rust, mosaics, gray spots, and scabs are the main leaf diseases that impact apple output. Fruit yield and quality can be enhanced by early detection and identification of fruit illnesses [6]. Computerized methods must be used because the human detection process requires a significant amount of time and effort.

The merged vector was then supplied to the ensemble subspace discriminant (ESD) classifier for disease identification after being optimized using a modified genetic algorithm (GA).

The rest of this document is structured as follows: Section 2 discusses the current studies (related research). The suggested framework is explained in Section 3 using various visualizations and mathematical modeling, and the findings are shown in Section 4. Lastly, Section 5 presents the conclusions.

2.Related Work

To identify and diagnose illnesses in fruit plants and leaves, researchers have developed a number of automated techniques [20,21]. These systems make use of deep CNN features that are handmade. A two-phase technique for identifying illnesses in citrus fruits was created by Sharif et al. [5]. In the first phase, the lesion area was detected in the citrus fruits and leaves. They used an optimal weight-based segmentation technique to find the lesion. The color, texture, and geometric elements were then merged. Skewness, entropy, and PCA-based techniques were used to pick features. The chosen feature vector was then fed into a support vector machine (SVM), which produced a recognition accuracy of 90.4% on a private dataset and 97% for citrus illnesses. A back-propagation neural network-based technique for detecting grape leaf disease was presented in [22]. The diseased area was first divided using the Otsu segmentation approach after the images were denoised using a Wiener filtering technique based on the wavelet transform. The perimeter, circularity, area, shape complexity, and rectangularity were then used to determine the characteristics.

A CNN-based approach for identifying apple leaf diseases

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 07, NCIRTAI - 2026

citrus fruits are high in vitamin C, they are good for human health [5]. Fruit

When classifying fruit diseases, feature extraction is essential. In order to identify diseases, both handmade and deep CNN features are retrieved throughout the feature extraction process. Color and texture qualities are crucial handmade characteristics for identifying fruit plant and leaf diseases [12]. Researchers used color features to identify diseases in [13]. Researchers used color texture characteristics [15] and the local binary pattern (LBP) [14] to extract texture information. Deep learning-based features have also attracted a lot of interest for the categorization of various fruit illnesses [16,17]. Accurate recognition can be enhanced by deep CNN features. Plant leaf diseases are identified using some deep feature-based algorithms [18,19]. To choose the best characteristics, academics have also suggested feature selection methods. The best feature selection methods can reduce the computing time.

A system for categorizing fruit plant diseases is presented here. 16 classes of the Plant Village database, which includes various fruit plants like apples, blueberries, cherries, oranges, peaches, grapes, and strawberries, were used to test our method. Using the maximum mean value serial technique, we extracted the LBP, color, and deep ResNet50 features from the suggested framework and then concatenated them into a single vector.

A strategy for categorizing various fruit illnesses was given by Khan et al. [6]. They made use of the pretrained VGG-16 and Caffe AlexNet networks' characteristics. Researchers created a technique for the segmentation and identification of grape leaf disease in a different study [24].

A method for reducing and improving haze was initially implemented in this system. The optimal channel was then chosen using LAB color transformation. The geometric, color, and texture features were used to calculate the features during feature extraction. Canonical correlation analysis was used to integrate the retrieved features, and neighborhood component analysis was used to choose the optimal feature set. This approach produced segmentation and classification accuracies of 90% and 92%, respectively.

A method for diagnosing apple leaf diseases was proposed by Khan et al. [25]. First, a hybrid approach was used to improve the pictures. This hybrid approach combines three-dimensional (3D) Gaussian, 3D median, and 3D box filtering with de-correlation. The LBP, color, and color histogram-based features were retrieved, merged, and optimized using a GA. A deep CNN model-based approach for detecting apple leaf diseases was presented by Chao et al. [26]. They used global average pooling layers to integrate the DenseNet and Xception models.

An SVM was used to classify the characteristics that was introduced by Liu et al. [23]. They obtained a 97.62% accuracy rate by using 13689 photos of apple leaves to train the AlexNet model.

Each of the aforementioned methods concentrated on utilizing deep learning to classify fruit illnesses. Deep learning training challenges have been highlighted, and several

leaf generative adversarial network (GAN) for grape disease recognition was shown in [28], where images of grape leaves in four distinct diseases were produced.

studies have used data augmentation strategies to address these challenges. Additionally, the problem of irrelevant features—which can be fixed by feature selection techniques—has not received much attention from researchers. However, problems with the categorization stage continue.

3.Proposed Methodology

The suggested framework for classifying fruit plant diseases from leaf photos is presented in this part along with technical and visual details. LBPs [29], strong color features, and deep ResNet50 [30] features are among the main processes of the suggested framework, along with data augmentation to increase the number of photos in each class. A modified GA is then used to optimize these extracted features after they have been concatenated using a maximum mean value serial technique. In order to recognize images, the optimized feature vector is then input into several classifiers. The primary flow diagram for this procedure is shown in Fig. 1. Each procedure's specifics are given below.

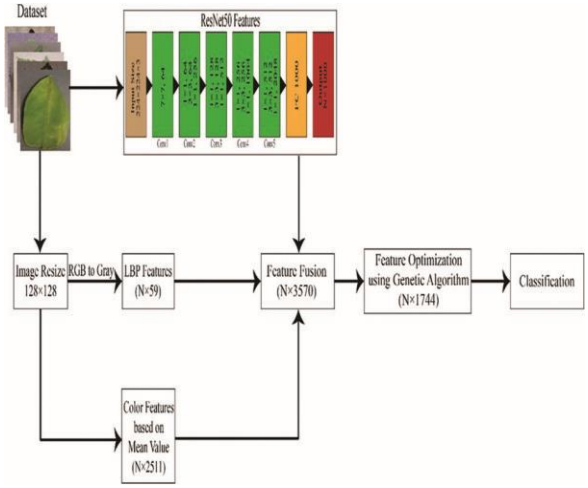


Figure 1: Flow of the proposed fruit diseases classification framework

There are now 912 pictures of healthy strawberry classes, up from 456. The following is a mathematical definition of the horizontal and vertical flip operations:

$FH = Ii(n+1-j)$ (1)

$FV = I(n+1-j)j$ (2),

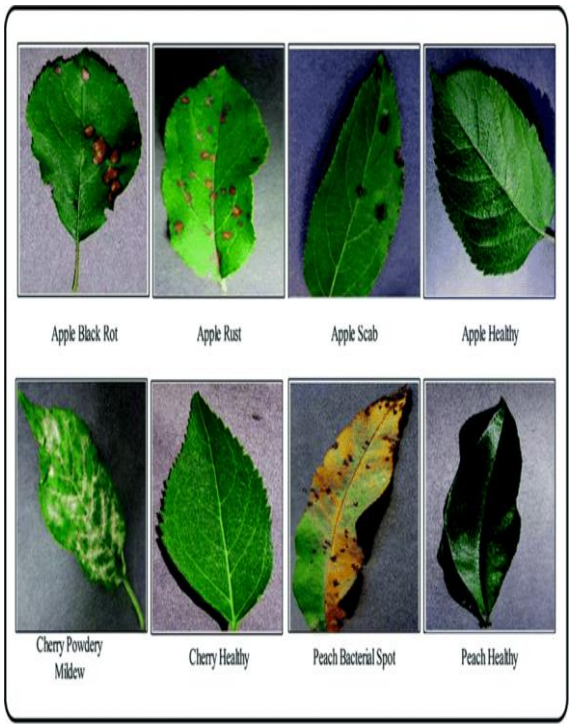
where I is the original database picture with dimensions N ×M ×k, FH is the horizontal flip operation, and FV is the

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 07, NCIRTAI - 2026

were taken out of the CNN models. Furthermore, researchers [27] used a global average pooling layer to collect features from the VGG-16 network in order to build a transfer learning strategy for the diagnosis of apple illnesses. A deep CNN-based approach for grape leaf disease identification was presented by Adeel et al. [4]. Using pretrained networks like AlexNet and ResNet101, they applied the transfer learning technique and used kurtosis and Yager entropy to choose the best features. A

3.1 Dataset Collection

The Plant Village database [31] was used in this study to provide a dataset for the assessment of the suggested method. There are 38 classifications and 54303 leaf photos in the Plant Village dataset. We used 16 kinds of both healthy and sick fruit trees in this investigation. Apple, blueberry, cherry, grape, orange, peach, and strawberry leaves were used to take the pictures. This dataset's photos were all shrunk to 256 x 256 pixels. Fig. 2 displays sample photos from this dataset.



3.2 Dataset Augmentation

In order to balance the dataset, we used data augmentation in this study to enhance the quantity of data for classes with a small number of photos. Image flips were used for the augmentation in order to change the original image's perspective. The apple scab class had 630 images at first, however after augmentation, there were 1260 images. After augmentation, the number of photos in the original apple cedar rust class increased from 276 to 550. After augmentation, the number of photos in the grape healthy and peach healthy classes climbed to 846 and 720, respectively, from 423 and 360.

vertical flip operation.

3.3 Feature Extraction

An essential component of image processing and computer vision is feature extraction. To represent the information in the image, features are extracted. Correct classification of images is made possible by the extraction of robust features. Both classical and deep learning characteristics—that is, LBP, color, and deep features extracted using the ResNet50 CNN pretrained model—were the main focus of this investigation. Below is a mathematical explanation of each technique.

3.3.1 Local Binary Patterns (LBP) Features

Texture analysis on image datasets is a common application of LBP features. They use nearby pixels to infer an image's texture information. Let Xh(a,b) be an image of size M × N, where (a,b) represents the pixel positions. The symbols qc and qh stand for the central pixel and its neighboring pixels, respectively. The LBP characteristics can be computed using these parameters:

$$LBP^{(q_c)}{}_{h,r} = \sum_{h=0}^{h-1} x(q_h - q_c)2 \qquad (3)$$

$$x(y) = \begin{cases} 1, & y \geq 0 \\ 0, & y < 0 \end{cases} \qquad (4)$$

where h denotes the neighboring pixel, and r is the neighborhood radius. The extracted feature set size of the LBP features was N ×59. Here, N represents the total number of images, and for each image, 59 features were extracted.

feature maps were 128 and 512. The following six blocks had feature sizes of 256 and 1024. Feature maps of sizes 512 and 2048 were found in the last three blocks.

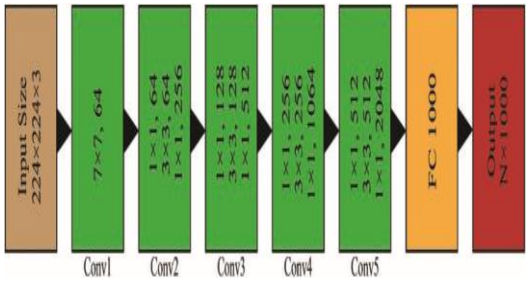


Figure 3: An architecture of ResNet50 deep learning model

3.4 Features Fusion

Following feature extraction, we used a novel technique called the "maximum mean value serial approach" to integrate all of the features into a single vector. To improve classification accuracy, a new feature set of size N× 3570 was created by combining three feature vectors—LBP, color, and ResNet50 features—into a single vector. The following is a mathematical expression for the combination process:

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 07, NCIRTAI - 2026

3.3.2 Color Features

When using RGB photographs to identify diseases, color features [32] are essential. To extract the color information from the database, we used three color spaces: RGB, HSV, and LAB. The color space was first divided into individual channels before being transformed into a histogram. The mean, standard deviation, variance, kurtosis, and skewness were then computed for every channel. This computation was carried out for each of the three color spaces' nine channels. A vector c_fv(i) of size N ×6500 was obtained by serially combining the calculated parameters. By creating a threshold function that chooses the features based on the mean value and removes roughly 60% to 70% of irrelevant characteristics, robust color features were chosen. This can be explained mathematically.

By creating a threshold function that chooses the features based on the mean value and removes roughly 60% to 70% of irrelevant characteristics, robust color features were chosen. This can be explained mathematically as follows: where ψcf (i) is the robust color feature set chosen from c_fv(i) of size N ×2511.

$$c\_fv(i) = \{RGB; LAB; HSV\}_{i=1}^{3} \quad (5)$$

$$_{CF} = \psi^{cf}(i),\ c\_fv(i) \geq \mu \quad (6) \quad \textit{Eliminate otherwise}$$

where $\psi_{cf}(i)$ is the robust color features set of size $N \times 2511$, selected from $c\_fv(i)$.

3.3.3 Deep Learning Features

The ResNet50 [30] model was used in this investigation to extract deep features. 50 layers and 16 bottleneck residual blocks made up this model, which was developed utilizing the residual learning technique. Every residual block underwent three convolutional processes of sizes 1 × 1, 3 × 3, and 1 × 1.

The model produced 2048 features from an input image that was 224 x 224. The first three leftover blocks have feature map sizes of 64 and 256. The following four blocks'

In this investigation, the combined feature vector was used to pick the best features using the GA. The input was the composite feature vector. The population, or group of solutions, was chosen based on the best features. The remedy, referred to as a chromosome, is made up of genes that illustrate a potential fix for the given issue. After every iteration, the GA assessed the produced solutions using the fitness function. The GA chose parents at random from the population. The following generation is produced by these parents. A threshold function was then applied to the optimal solution that the GA ultimately produced. The ideal solution's harmonic mean served as the foundation for the threshold function.

Initialization: The GA performs an initialization using a set of individuals, known as a population. The population was set to 20, which is the possible number of solutions. The number of generations was set to 500, signifying that this algorithm performed 500 iterations to evaluate the fitness function. The mutation rate and crossover rate were set to

Let us consider three feature vectors with dimensions of N × 59, N × 2511, and N × 1000, respectively: ψ(LBP), ψ(cf), and ψ(df). Assuming that ψfd is a fused vector with dimensions N × K, the mean value of each vector was calculated as follows:

$$\left(\sum(\psi(LBP))\right),\quad \mu(2) = \frac{1}{-}\left(\sum(\psi(cf))\right),\quad \mu(3) = \frac{1}{-}\left(\sum(\psi(df))\right)$$

$$\mu(1) = \frac{1}{-}\check{\mu} = \frac{\sum(\mu(1), \mu(2), \mu(3))}{3}$$

$$\psi_{fd}(i) = \begin{pmatrix} \psi(LBP)_{N\times 59} \\ \psi(cf)_{N\times 2511} \\ \psi(df)_{N\times 1000} \end{pmatrix}$$

where $\psi(LBP)_N \times 59$, $\psi(cf)_N \times 2511$, and $\psi(df)_N \times 1000$ represent the LBP, color, and deep ResNet50 features, respectively. $\psi_{fd}(i)$ is the serial fused vector, and $\widetilde{\psi_{fd}(i)}$ is the final maximum mean value of the serial approach-based feature vector. This vector is further optimized using a modified GA, and this process is known as threshold-function-based GA feature selection.

3.5 Features Selection

A GA is a feature optimization method that draws inspiration from the notion of biological evolution [33]. The GA is a member of the evolutionary class of algorithms.

$$\gamma_{cr} = CrossOver(X_1, X_2) \quad (13)$$

$$X_1 = vZ_1 + (1-v) \times Z2 \quad (14)$$

$$X_2 = vZ_2 + (1-v) \times Z_1 \quad (15)$$

Mutation: Mutation maintains genetic diversity and avoids the premature convergence of the GA. In this process, one or more genes are flipped based on the defined mutation rate. In our method, a uniform mutation rate of 0.01 was utilized.

Fitness Function: The fitness function is a key parameter for selecting the best features. The fitness function verifies the quality of the solution; hence, a good fitness function yields more optimized results. In this method, the "fitcknn" function is used as the fitness function. This function returns the K-nearest neighbor (KNN) classification model based on the input features. The Euclidean distance is used in this fitness function for the KNN classification model. The Euclidean distance is formulated in mathematical form as follows:

$$(p, q) = \sqrt{(p_1 - q_1)^2 + \ldots + (p_n - q_n)^2} \quad (16)$$

To calculate the error rate, we utilized the "kfoldLoss" function. This function returns the loss of a cross-validated classification model. The classification error for the loss function of the KNN model is expressed as:

$$Kloss = \sum_{h=1}^{n} m_h I\{\hat{r}_h \neq r_h\} \quad (17)$$

After the completion of all iterations, a new optimized feature

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 07, NCIRTAI - 2026

0.01 and 0.8, respectively.

Selection: The most important step in the GA is the selection of the best features. In this study, we applied the roulette-wheel method for parent selection. The probability-based roulette wheel selection is mathematically defined as:

$$W \quad \frac{w_i}{(w_i)} \quad =(11)$$

$$w_i = \exp\left(-b_1 \times \frac{S_\rho}{W_l}\right) \quad (12)$$

where, b_1 represents the selected parent pressure, S_ρ denotes the sorted population, and W_l is the last selected population.

Crossover: This step generates a better individual by swapping the genes of two parents. In this study, we utilized a single-point crossover rate of 0.8. A single-point crossover randomly selects a point from both parents. A high crossover rate may cause a premature convergence of the GA. In mathematical form, it can be defined as:

4. Experimental Setup and Results

The proposed framework was evaluated on 16 classes of the publicly available Plant Village dataset. A brief description is provided In the preprocessing step, we performed data augmentation to increase the number of images per class. Different features, including handcrafted and deep features, were extracted and combined; subsequently, the fused vector was optimized using the GA. The handcrafted features included LBP and color features, and for deep feature

Algorithm: For GA based Features Optimization
Output: $Vec(i) \leftarrow SelectedVector$

Input: $\psi_{fd}(i) \leftarrow FusedVector$

Step 1: Parameters Initialization

- $Population \leftarrow N = 20$
- $Iterations \leftarrow T = 500$
- $\gamma_{cr} \leftarrow 0.8$
- $\gamma_{mr} \leftarrow 0.01$
- $\beta \leftarrow 5$

Step 2: Fitness Function

- $Type \leftarrow FKNN$
- $Neighbors \leftarrow 5$ - $K-fold \leftarrow 2$
- $Distance \leftarrow Euclidean\ n$
- $Kloss \leftarrow \sum_{h=1} m_h I\{\hat{r}_h \neq r_h\}$

Step 3: Selection

- $W = \frac{w_i}{(w_i)}$

$w_i \leftarrow \exp \quad -b_1 \times \frac{S_\rho}{W_l}$

Step 4: Crossover

- $\gamma_{cr} \leftarrow CrossOver(X_1, X_2)$

vector is obtained; subsequently, it is passes into a harmonic-mean-based threshold function. Mathematically, the function is

$$- = \frac{}{n \quad 1}$$ expressed as follows: *n(18)*

The final selected features represented by $Vec(i)$ were validated using multiple classifiers. The highest accuracy was achieved using the ESD.

Step 8: Apply Threshold function by Eqs. (18) and (19).

Step 9: Selected best features are returned as output.

extraction, we utilized the ResNet50 model. In the training phase of the model, a 70:30 approach was used. Extensive experiments were performed on different cross-validations, including 5-, 10-, 15-, and 20-fold cross-validations. Multiple classifiers were selected for a fair comparison, including the linear SVM, quadratic SVM, cubic SVM, medium Gaussian SVM, coarse Gaussian SVM, medium KNN, cosine KNN, weighted KNN, ensemble bagged trees (EBT), and ESD. Each classifier was evaluated using different performance measures such as accuracy, false-negative rate (FNR), precision, sensitivity, F1 score, and time.

Table 1: Technical details of the selected database

Database details		
Total classes	16	
Total images	15801	
Training ratio	70%	
Testing ratio	30%	
Image size	256	256

4.1 Results for 5-Fold Cross-Validation

In the first experiment, the optimized feature set was fed to the classifiers using five-fold cross validation for evaluation. The best accuracy was 99%, achieved using the ESD classifier, as shown in Tab. 2. Other measures such as precision, sensitivity, F1 score, and FNR calculated using the ESD were 95.5%, 95.5%, 95.5%, and 1%, respectively. The accuracy was verified, as shown in Fig. 4. The computational time for the ESD was 746.3 s. The best computational time was 68.2 s, which was achieved on the EBT classifier. However, the accuracy achieved using the EBT was 95%. The worst performance observed in the five-fold cross-validation was an accuracy of 89.7%, which was calculated using the C-KNN classifier.

4.3 Results for 15-Fold Cross-Validation

For the 15-fold cross-validation, the best results were obtained using the ESD classifier. The best accuracy, FNR, precision, sensitivity, and F1 score obtained using the ESD were 99%, 1%, 99.7%, 99.5%, and 99.6%, respectively, as shown in Tab. 4. This accuracy was further verified, as shown in Fig. 6. However, the computational time of the ESD was 1709.5 s. The Q-SVM and C-SVM yielded good accuracies of 98.9% and 98.8%, respectively. However, the Q-SVM and CSVM incurred 1339.9 and 1678.9 s for recognition, respectively. The C-KNN recorded the worst accuracy of 90.1%. The best computational time afforded by the EBT was 218.2 s.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 07, NCIRTAI - 2026

Step 5: Mutation

- *Type*← *Uniform*

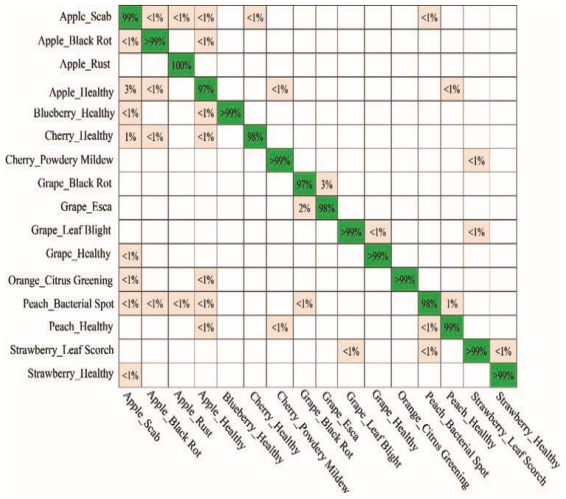
Step 6: Repeat Step 2
Step 7: *Sel*(*i*)← *BestFeatures*

Table 2: Results for 5-Fold cross-validation

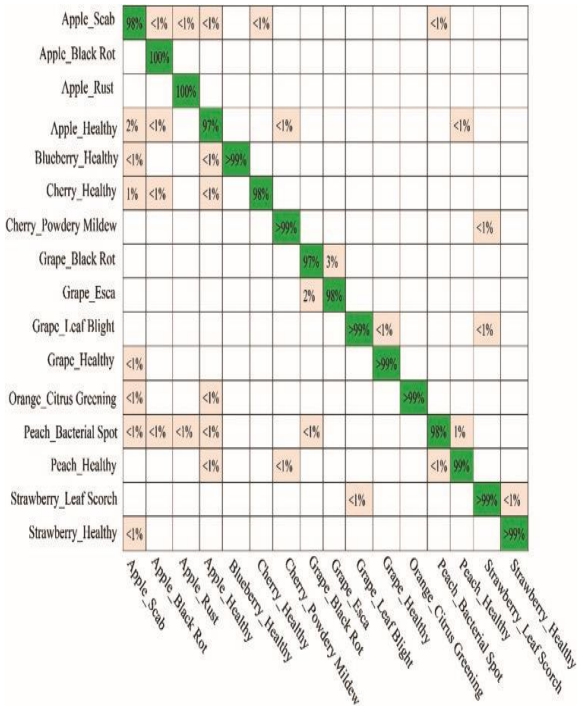
Classifier	Accuracy (%)	FNR (%)	Precision (%)	Sensitivity (%)	F1 Score (%)	Time (s)
L-SVM	98.4	1.6	99.3	98.8	99.0	337.8
Q-SVM	98.8	1.2	99.2	99.2	99.2	564.4
C-SVM	98.7	1.3	98.4	98.1	98.3	684.9
MGSVM	96.5	3.5	97.7	97.4	97.5	1071.3
CGSVM	95.9	4.1	98.4	97.1	97.7	1058.2
M-KNN	90.0	10.0	93.9	94.8	94.3	281.9
C-KNN	89.7	11.3	94.0	94.9	94.4	342.5
W-KNN	90.5	9.5	93.5	95.1	94.3	270.7
EBT	95.0	5.0	97.7	96.9	97.3	**68.2**
ESD	**99.0**	**1.0**	**99.5**	**99.5**	**99.5**	746.3

4.2 Results for 10-Fold Cross-Validation

Next, we used 10-fold cross-validation to evaluate the proposed framework. The maximum accuracy achieved on the Q-SVM and ESD was 99%, as shown in Tab. 3. The FNR for both classifiers was 1%, and the highest precision was 99.7%, which was achieved using the Q-SVM. The accuracy was verified, as shown in Fig. 5. The computational times for the Q-SVM and ESD were 729.6 and 993.3 s, respectively. The sensitivity and F1 score were 99.4% and 99.5%, respectively, achieved using the ESD classifier. The best computational time was 82.9 s, which was achieved using the EBT classifier with a 95.1% accuracy. The C-KNN recorded the worst accuracy of 90.1%.



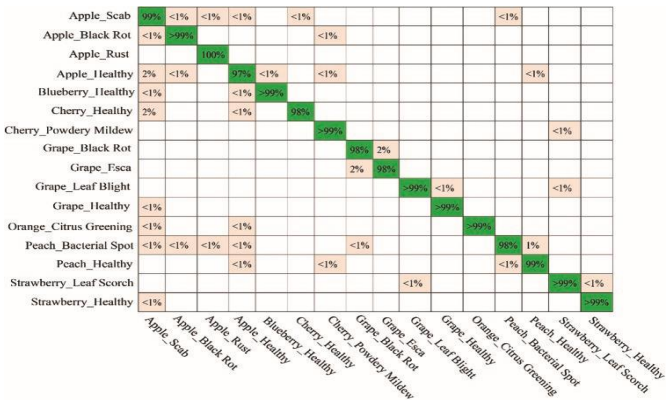
However, those classifiers required 1870.8 and 2151.9 s, respectively. The best computational time was achieved by



4.4 Results for 20-Fold Cross-Validation

The final experiment was performed using a 20-fold cross-validation. The maximum accuracy achieved in this experiment was 99% for the ESD classifier, whereas the FNR, precision, sensitivity, F1 score, and computational time were 1%, 99.5%, 99.4%, 99.4%, and 2130.9 s, respectively, as shown in Tab. 5. In addition, this accuracy was verified, as shown in Fig. 7. The accuracies of the Q-SVM and C-SVM were 98.9% and 98.9%, respectively, indicating good performances.

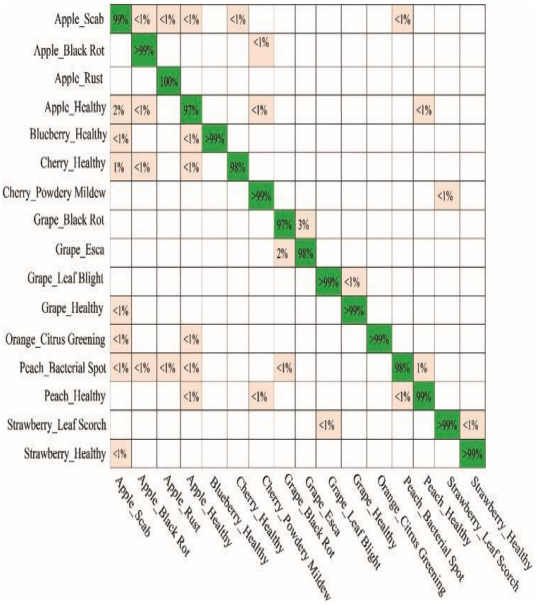
Author	Year	No. of classes	Accuracy (%)
Khan et al. [6]	2018	6	98.6
Barbedo [7]	2019	8	82.0
Thapa et al. [20]	2020	4	97.0
Akram et al. [21]	2020	5	97.8
Proposed	**2021**	**16**	**99.0**



International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 07, NCIRTAI - 2026

the EBT, with a 95.3% accuracy. Meanwhile, the C-KNN classifier recorded the worst accuracy of 90.3%.

Additionally, we compared the proposed framework with previous fruit plant disease detection techniques. Specifically, we compared our technique with methods evaluated on only four to eight classes, as presented in Tab. 6. Khan et al. [6] performed experiments on six classes of diseases and achieved a 98.6% accuracy. The authors of [7] evaluated their model on eight classes and obtained an accuracy of 82%. Meanwhile, the authors of [20,21] used four and five classes, respectively, to verify their methodology and achieved accuracies of 97% and 97.8%, respectively. In this study, we evaluated our framework on 16 classes comprising different diseases and healthy images and achieved an accuracy of 99



5.Conclusion

Here, a novel framework for categorizing fruit plant illnesses from leaf photos was introduced. Dataset acquisition, data expansion, LBP extraction, color based on mean value, ResNet50 features, feature fusion, feature optimization using enhanced GA, and classification were the main stages of the suggested system. We conducted comprehensive trials to evaluate our approach, and the results were promising. The ESD and Q-SVM classifiers yielded a maximum accuracy of 99%, while the precision, sensitivity, and F1 score were determined to be 99.7%, 99.5%, and 99.6%, respectively. After analyzing every result, we came to the conclusion that our suggested framework outperforms the other approaches under comparison. Additionally, we came to the conclusion that using the threshold function to choose features substantially reduced computation time without sacrificing classification accuracy. Future research will take into account other fruit types and employ fresh optimization strategies to reduce calculation time. The Ministry of Science, ICT (No. 2019H1D8A1105622) and the Soon yang University Research Fund provided funding for this study through the National Research Foundation of Korea's (NRF) X-mind Corps program. Regarding the current study, the authors affirm that they have no conflicts of interest to disclose.

6.Future Enhancement

Future research can focus on extending the proposed system in several directions. First, additional fruit categories and disease types can be included to improve system generalization. Second, real-time disease detection can be implemented using mobile or edge devices for practical agricultural applications. Third, advanced deep learning architectures such as Vision Transformers or lightweight CNN models can be explored to reduce computational cost. Furthermore, integrating environmental data such as temperature and humidity may improve prediction accuracy. Finally, deploying the system as a cloud-based or mobile application can support farmers in making timely disease management decisions.

REFERENCES

[1] J. Li, X. Rao and Y. Ying, “Detection of common defects on oranges using hyperspectral reflectance imaging,” *Computers and Electronics in Agriculture,* vol. 78, pp. 38–48, 2011.

[2] S. Zhang, H. Wang, W. Huang and Z. You, “Plant diseased leaf segmentation and recognition by fusion of superpixel, K-means and PHOG,” *Optik,* vol. 157, pp. 866–872, 2018.

[3] B. J. Samajpati and S. D. Degadwala, “Hybrid approach for apple fruit diseases detection and classification using random forest classifier,” in *2016Int. Conf. on Communicationand Signal Processing (ICCSP)*, Melmaruvathur, India, pp. 1015–1019, 2016.

[4] M. B. Tahir, M. A. Khan, K. Javed, S. Kadry, Y. D. Zhang *et al.,* “Recognition of apple leaf diseases using deep learning and variances-controlled features reduction,” *MicroprocessorsandMicrosystems,* vol. 7, pp. 104027, 2021.

[5] I. M. Nasir, A. Bibi, J. H. Shah, M. A. Khan, M. Sharif *et al.,* “Deep learning-based classification of fruit diseases: An application for precision agriculture,” *Computers, Materials & Continua,* vol. 66, pp. 1949–1962, 2021.

[6] H. T. Rauf, B. A. Saleem, M. I. U. Lali, M. A. Khan, M. Sharif *et al.,* “A citrus fruits and leaves dataset for detection and classification of citrus diseases through machine learning,” *Data in Brief,* vol. 26, no. 5, pp. 104340, 2019.

[7] G. Manogaran, M. Alazab, K. Muhammad and V. H. C. De Albuquerque, “Smart sensing based functional control for reducing uncertainties in agricultural farm data analysis,” *IEEE Sensors Journal,* vol. 5, pp. 1–9, 2021.

[8] N. Muhammad, S. Rubab, N. Bibi, O. Y. Song, M. A. Khan et al., “Severity recognition of aloe vera diseases using AI in tensor flow domain,” Computers, Materials & Continua, vol. 66, pp. 2199–2216, 2021.