

ORIGINAL ARTICLE

# DeepLeaf: an optimized deep learning approach for automated recognition of grapevine leaf diseases

Fatma M. Talaat<sup>1,2</sup>  · Mahmoud Y. Shams<sup>1</sup> · Samah A. Gamel<sup>3</sup> · Hanaa ZainEldin<sup>4</sup>

Received: 19 April 2023 / Accepted: 11 January 2025 / Published online: 28 February 2025

© The Author(s) 2025

## Abstract

Plant diseases can cause severe losses in agricultural production, impacting food security and safety. Early detection of plant diseases is crucial to minimize crop damage and ensure agricultural sustainability. Manual monitoring is often impractical due to the complexity and time involved, making automated disease recognition essential. This study presents a new Plant Disease Detection Algorithm (PDDA) called DeepLeaf focused on identifying four common grapevine diseases: leaf blight, black rot, stable, and black measles. The PDDA integrates three key modules: an Image Preprocessing Module, a Feature Extraction Module, and an Optimized Convolutional Neural Network (OCNN)-based Classification Module. The OCNN forms the core of the classification system, with its hyperparameters fine-tuned using fuzzy optimization to enhance performance. Pre-processing techniques are applied to analyze diseased leaves, and a logistic regression algorithm is used to downsample the features for better analysis. The CNN is trained on images from the Plant Village dataset, allowing it to detect and classify grapevine leaf diseases accurately. The proposed model's efficiency in the automated diagnosis of grapevine diseases is demonstrated by its remarkable 99.7% accuracy rate. This high accuracy indicates that the PDDA may help with more effective and scalable plant disease monitoring, which will ultimately allow better agricultural practices.

**Keywords** CNN · Detection of leaf disease · Plant disease · Extracting features

## 1 Introduction

The practice of producing plants and cattle is known as “agriculture”[1]. Many of the essential food plants and livestock we eat now were domesticated thousands of years ago thanks to agriculture. Food insecurity, which is a significant contributor to crop diseases, is one of the most critical challenges facing humanity [2]. Research found that crop diseases are responsible for about 16% of the world’s lost agricultural yields. Pest losses are anticipated to be between 26 and 29% for soybeans and approximately 50% for wheat worldwide [3]. Deep learning and computer vision techniques have recently been widely used in many application areas, including medicinal applications. Recent studies have demonstrated the effectiveness of unsupervised deep learning in improving disease prediction and diagnosis across various fields. In breast cancer research, unsupervised models enhanced pattern recognition and accuracy, while for heart disease, deep learning was used to optimize ensemble learning techniques. Similarly, unsupervised methods improved early detection of chronic kidney disease by efficiently handling complex, high-dimensional data. These advancements highlight the growing impact of unsupervised deep learning in medical diagnostics [4–6].



Artificial Intelligence (AI) was employed widely during the COVID-19 pandemic to diagnose pulmonary illnesses and other warning applications [7, 8]. By identifying and diagnosing crop diseases early, similar sophisticated technology can be used to reduce their adverse consequences. Nowadays, the agriculture field is a crucial and innovative context for computer vision researchers. The basic purpose of agriculture is to generate a diverse range of significant plants and crops. Plant diseases in farming affect crop quantity and quality, so they must be managed quickly [9]. Recently, agriculture researchers have focused on infections that affect a variety of fruits and crops. The researcher proposed a set of techniques for locating and detecting infections in fruits and crops [10, 11]. Grapes are considered a difficult crop to produce due to the plants' periodic viral attacks, which severely diminish grape crop [12]. Therefore, controlling diseased crops is essential to prevent them from destroying product amount and quality.

Human inspections are most commonly employed for illness diagnosis, but they have many drawbacks, including cost, time, availability, and requirement for a lot of work. Various bacterial and fungal illnesses primarily express themselves on the surface of the leaves and fruit. Pests, like bugs, have intricate patterns that are a challenge to decipher. Otherwise, bacteria are single-celled organisms with a short lifetime. They reconstruct by binary fusion, which involves splitting one cell into two. Viruses are small particles with no membrane proteins but with genes and filaments [13]. Grape leaves can be affected by rust, downy mildew, powdery mildew, scab, black rot, leaf blight, and black measles, among other diseases. Recently, computer-aided-based image processing techniques for identifying and recognizing illnesses in horticulture have been created. Researchers employ image processing-based techniques to identify the position, color, form, scale, and limit of the sick portion. A range of novel techniques are used for preprocessing and symptom segmentation. Conventional machine learning methods such as extracting features, image segmentation, and pattern recognition have gotten better at spotting and detecting plant diseases [14–16].

It is challenging to extract characteristics from a huge dataset using conventional methods like manual detection and machine learning. The presence of duplicated characteristics is a drawback of the deep model [17]. CNN is a category of deep neural networks that are used frequently for image analysis. CNN can categorize certain features from pictures. CNN usage includes natural language processing, imaging categorization, video and image analysis for medical applications, and image/video detection. The CNN [18] is composed of 3 classes of layers: fully connected (FC), max-pooling, and convolutional layers [19].

A CNN model is constructed when such layers are merged. The dropout layer and the activation function are two additional crucial factors in including these layers. CNN is among the most extensively employed types of neural networks and is effectively used for a wide range of computer vision applications across several industries [20]. Researchers have classified and identified plant diseases using different CNN models. For disease identification in potato and mango leaves, authors in [21] examined several CNN models, with AlexNet reaching 98.33 percent accuracy and a shallow CNN architecture getting 90.85 percent accuracy. Authors in [21, 22] employed a VGG16 model for predicting the degree of disease in apple plants and attained an accuracy of 90.40 percent. With a 99.72 percent accuracy rate, authors in [23] have successfully distinguished between normal and sick banana leaves using the LeNet model.

It will want a huge portion of unsegmented data to train a CNN architecture. A CNN model typically receives raw data as input. A set of parameters that are not reliant on the data is necessary for the CNN model and must be manually modified by the machine learning researcher. Hyperparameters are factors that influence the structure of a network and the trained network of a CNN [24]. The problem of hyperparameter optimization also includes finding a collection of hyperparameters that yields a sufficiently accurate model. Optimization of hyperparameters is the issue of selecting an appropriate hyperparameter model or the issue with loss function optimization across a graph-structured configuration space. It can be costly computationally to evaluate every potential combination of hyperparameter models. In this study, we introduce a novel Plant Disease Detection Algorithm (PDDA) for automated recognition of grapevine leaf diseases, addressing key challenges in the field.

Unlike previous approaches that often rely on conventional convolutional neural networks (CNNs), our algorithm incorporates innovative elements to improve performance. Specifically, we employ logistic regression

for feature downsampling, which enhances the efficiency of data processing while preserving essential information [25]. This is performed using logistic regression for feature downsampling in imbalanced datasets through an active learning strategy. Instead of randomly downsampling, the model selects the most informative samples, both from minority and majority classes, to minimize generalization error. Penalized logistic regression with tuned hyperparameters helps improve performance, especially in highly imbalanced datasets, outperforming traditional resampling methods [26]. Furthermore, our approach includes hyperparameter optimization using grid search, fine-tuning the model for optimal performance. By integrating these components, our PDDA achieves a remarkable accuracy rate of 99.7% in identifying four common diseases in grapevines: leaf blight, black rot, stable, and black measles. This contribution advances the state-of-the-art in automated plant disease detection, offering a robust and effective solution for agricultural applications.

## 1.1 Motivation

The detection and management of plant diseases are critical components of modern agriculture, significantly impacting crop yield and quality [27]. Plant diseases can lead to substantial economic losses, with studies indicating that they can reduce global agricultural production by up to 30% annually [28]. For instance, grapevine diseases not only threaten the health of the vines but also jeopardize the entire wine industry, which is a vital sector for many economies. Given the increasing demand for food security and the need to optimize agricultural practices, there is an urgent need for efficient and accurate disease detection methods [29].

Traditional methods for plant disease diagnosis primarily rely on visual inspection by trained experts. This approach is inherently limited by human error and the time-consuming nature of manual assessments, especially in large agricultural settings. Moreover, many plant diseases exhibit similar visual symptoms, complicating accurate identification and leading to potential misdiagnosis. Consequently, there is a pressing demand for automated solutions that can enhance diagnostic accuracy and speed [30].

Recent advancements in deep learning, particularly convolutional neural networks (CNNs), have revolutionized the field of image recognition and classification. These techniques have shown remarkable success in various applications, including medical imaging and autonomous vehicles. However, their application in agriculture, specifically for plant disease detection, remains an area with significant untapped potential. The ability of deep learning models to automatically extract features from images positions them as a promising tool for accurately diagnosing plant diseases, thereby addressing the limitations of traditional methods [31].

In light of these challenges and opportunities, we propose the Plant Disease Detection Algorithm (PDDA) as a novel approach to automating the identification of grapevine leaf diseases. Our method leverages an Optimized Convolutional Neural Network (OCNN) that integrates advanced image preprocessing and feature extraction techniques tailored specifically for grapevine images. By employing a feature downsampling technique using logistic regression, we aim to reduce data dimensionality while retaining critical information, enhancing the model's efficiency and performance. This study not only aims to provide an effective tool for grapevine disease detection but also contributes to the broader field of precision agriculture by demonstrating the effectiveness of deep learning approaches in managing crop health. Ultimately, our research endeavors to empower farmers with accurate diagnostic tools that can help mitigate crop losses and improve agricultural productivity.

## 1.2 Motivation

This paper proposes a new Plant Disease Detection Algorithm (PDDA), which combines three main modules: (i) Image Pre-processing Module (IPM), (ii) Feature Extraction Module (FEM), and (iii) OCNN-based Classification Module (OCM). The OCNN is an Optimized CNN. PDDA uses various preprocessing techniques to analyze leaf disease. The main contributions of this research paper are:

- Proposing a new Plant Disease Detection Algorithm (PDDA) that combines three main modules: an Image Pre-processing Module, a Feature Extraction Module, and an OCNN-based Classification Module.
- Using logistic regression to downsample the features gathered from the image dataset.
- Training an Optimized Convolutional Neural Network (OCNN) on the plant village dataset to achieve an accuracy of 99.7% in the automated recognition of grapevine leaf diseases.
- Demonstrating the effectiveness of the proposed algorithm and model in the detection of four diseases commonly encountered in grapevines: leaf blight, black rot, stable, and black measles.

The remaining work is organized as follows: a brief review of state-of-the-art techniques is discussed in Sect. 2. The proposed technique is described in detail in Sect. 3. Experimental evaluation is provided in Sect. 4. The conclusion and future works are presented in Sect. 5.

## 2 Related work

To automate plant disease detection, numerous researchers studied various ML, image processing, Deep Learning, and soft computing techniques. Automatic plant disease detection systems are innovative detection that is described in the literature and are discussed in this section. In Atila et al. [32], An architecture of EfficientNet deep learning was suggested for classifying plant leaf diseases, and its performance was evaluated against that of other state-of-the-art deep learning models.

The architecture of EfficientNet was trained using the technique of transfer learning, in which all the layers of the model were set to be trainable. The B5 and B4 models of the EfficientNet architecture, which had accuracy and precision values of 99.91 percent and 99.97 percent in the test dataset and 98.42 percent and 99.39 percent, respectively, the highest results among other deep learning models in the initial and modified datasets.

Ji et al. [33] created a technique for automatically detecting grape diseases based on CNNs. A united CNNs architecture consists of ResNet50 and InceptionV3, The United Model has been proposed and can be used to cluster grape images into four categories, including three distinct symptom images, including black rot, esca, leaf spot, and healthy images. The outcomes of the experiment demonstrate that the model outperforms the most sophisticated single basic CNNs, including ResNet50, DenseNet121, InceptionV3, and VGG16. To help farmers identify grape diseases, the proposed UnitedModel can be used as a decision support tool. Its average validation accuracy is 99.17 percent, and 98.57 percent for its average test accuracy.

The presented research by Sharma et al. [34] examines a solution to this issue by using segmented image data to train CNN models. The S-CNN model outperforms the F-CNN classification model with full images by almost doubling its performance to 98.6% accuracy with unseen data. Additionally, we demonstrate that the S-CNN model has significantly higher self-classification confidence than the F-CNN model using a tomato plant and a target spot disease type as an example. To categorize stress in plant shoot images, Azimi et al. [35] suggested an automatic image-based plant phenotyping approach.

The phenotyping approach proposed a 23-layered deep learning method and compared it with conventional ML methods and a few other deep architectures. The findings demonstrate that a straightforward 23-layered deep learning architecture is just as capable of classifying ceiling level stress from images of plant shoots as well-known cutting-edge deep learning architectures such as ResNet18 and NasNet Large (which depend on millions of trainable parameters). In their study of the data's accuracy and superiority, Gadekallu et al. [36] propose a hybrid PCA technique for extracting features that include the whale optimization procedure. The outcomes of this proposed method framework's classification of tomato diseases made it possible to take preventative action against risks related to crops. Sinha et al. [37] extracted process texture data employing k-means and a thresholding segmentation technique. Sorte et al. [38] introduced a texture-based pattern recognition method to recognize coffee plant leaf lesions. Two approaches were contrasted: (1) TBDR, which applied a convolutional neural network with deep learning directly to the test images, and (2) DLDR, which used a texture attribute vector

input to a classification algorithm that used statistical and local binary attributes. To determine the effectiveness of deep learning models in terms of “learning rate,” “batch size,” “activation function characteristic,” and “regularization rate,” Kallam et al. [39] use the TensorFlow application.

Based on pod length, various methods were used to categorize Okra plant diseases [40]. Certain conventional methods have a more maintenance-intensive, labor-intensive, and expensive detection process. Franczyk et al.-study’s [41] describe a CNN model with eight hidden layers that outperform ML approaches. Accurate grape identification can help with the detection of specific fungal diseases as well as type-dependent diseases. Additionally, this work shows that recognition can be done with inexpensive sensors that are already in widely used mobile devices. The “Automatic and Intelligent Data Collector and Classifier” AIDCC makes disease detection and visualization simpler and more affordable, as demonstrated by Kundu et al. in their study [42]. To automate data collection, disease detection, visualization of features, and prediction in pearl millet, this study creates a low-cost and user-friendly framework. By utilizing the color feature in Almadhor et al. [43].’s techniques, the feature vector extracts characteristics of common illnesses and feeds the values to the suggested classifier for leaf disease detection and classification. The suggested framework divides disease-infected regions using E color difference image segmentation.

To extract comprehensive, useful feature vectors, color (HSV, RGB) histogram and textural (LBP) features are also used. While disease recognition is carried out using sophisticated machine-learning classifiers, color and textural features are combined to identify and achieve similar results when compared to individual channels (Complex Tree, Fine KNN, Cubic SVM, Boosted Tree, Bagged Tree). Oyewola et al. [44] employed pre-processing and deep learning techniques to balance data. The PCNN and DRNN models were developed to identify diseases in cassava leaves. To balance the initial unbalanced dataset images of cassava leaf, which was biased towards the CBSD and CMD disease classes, they used a unique block processing technique. Furthermore, with a 96.75 percent accuracy rate, the DRNN model delivered the best outcomes for our predictive model. Abayomi-Alli et al. [45] use the image histogram methodology. Basavaiah et al. [46] discuss a model for the identification of various lesions that harm crops and cause a lack of cultivation. The accuracy of the DT classifier and the RF classifier in classifying the diseases is 90% and 94%, respectively. According to the findings, the decision tree classifier is less accurate than the random forest classifier. In comparison to other cutting-edge techniques, the suggested method has a sizable advantage in terms of accuracy and computational and classification accuracy.

Using ML models for SVM comparative analysis, Abdu et al. [47] discuss some work in this area. Taking into account architecture, processing capabilities, and training data, both models were evaluated on a dataset of conventionally displayed large-scale horticultural leaf lesion images. When it comes to categorizing and recognizing images, convolutional networks are a particularly effective class of neural networks. The CNN Network discusses various methods for classifying diseases based on images from various domains, such as medical images [48], gesture recognition images, disease images, and images of diabetes [49]. To identify sick leaves in banana, guava, and mango fruit crops, [50] suggests a convolutional neural network (CNN)-based deep learning model that is both accurate and lightweight. Three distinct degrees of feature reuse are included in the model proposal. Eight various categories of healthy and damaged leaves from three distinct fruit species comprised the available resource that was used to train the algorithm. According to the trial, the model employs 101,000 factors and has a 99.14% success rate in identifying sick leaves. The pre-defined models VGG16, VGG19, EfficientNet, ResNet, MobileNet, and Inception can better classify segmented parts. Convolutional neural networks outperform the TTA algorithm in terms of accuracy when using classification and feature extraction methods. Table 1 provides an overview of earlier research projects.

The research gaps in previous algorithms can be summarized as in the following:

- Limited focus on feature downsampling techniques for efficient data processing.
- Insufficient exploration of hyperparameter optimization methods for deep learning models.
- Lack of comprehensive integration of image preprocessing, feature extraction, and classification modules tailored to grapevine leaf disease detection.

**Table 1** Summary of different research work

References	Year	Dataset used	Model	Accuracy (%)
Atila et al. [32]	2021	PlantVillage	Efficient Net CNN	99.91
Ji et al. [33]	2020	PlantVillage	InceptionV3, ResNet50	98.57
Sharma et al. [34]	2020	Plant Village	F-CNN, S-CNN models	98.6
Azimi et al. [35]	2021	Sorghum plant shoot	Deep learning	92
Gadekallu et al. [36]	2021	Plant Village	DNN	94
Sinha et al. [37]	2020	Brodatz texture	k-means, thresholding	–
Sorte et al. [38]	2019	obtained from a Federal Institute of South Minas Gerais coffee plantation farm	CNN, Deep learning	97
Kallam et al. [39]	2018	dataset 4	TensorFlow	–
Raikar et al. [40]	2020	Ladies Finger	AlexNet, GoogleNet, ResNet50	63.45 68.99 99
Franczyk et al. [41]	2020	WGisd	Resnet	99
Kundu et al. [42]	2021	Blast and rust compressed	ResNet-50, VGG-19, VGG-16, Inception, Inception-V3, ResNet-V2	98.78
Almadhor et al. [43]	2021	Built dataset of guava images	Complex tree, Cubic SVM, Boosted tree, Bagged tree Fine KNN	99
Oyewola et al. [44]	2021	Cassava mosaic disease	CNN	96.75
Abayomi et al. [45]	2021	Cassava mosaic disease	MobileNetV2	99.7
Basavaiah et al. [46]	2020	Plant Village	Random Forest, Decision Tree	94 90
Abdu et al. [47]	2020	Plantvillage, digipathos	SVM, Deep Learning	97

- Minimal emphasis on the use of logistic regression for feature downsampling in plant disease detection algorithms.
- Inadequate comparison of proposed algorithms with state-of-the-art models using a diverse range of performance metrics.
- Limited investigation into the scalability and generalizability of proposed algorithms to real-world applications.
- Sparse utilization of grid search for hyperparameter optimization in deep learning models for plant disease detection.
- Scarce research on the utilization of an Optimized Convolutional Neural Network (OCNN) for plant disease classification.
- Limited exploration of novel algorithms combining image preprocessing, feature extraction, and classification modules for plant disease detection.
- Insufficient discussion on the practical implications and potential applications of proposed algorithms in agriculture and plant pathology.

In recent years, various studies have focused on the integration of advanced technologies to enhance agricultural practices and decision-making processes. Alshathri et al. [51] introduced a new reliable system for managing virtual cloud networks, showcasing the potential of cloud computing in agricultural applications. Complementing this, Talaat and Gamel [52] developed a reinforcement learning-based hyper-parameter optimization algorithm (ROA) for convolutional neural networks, which could be pivotal in improving the accuracy

of models used for agricultural predictions. Additionally, Shams et al. [53] emphasized the importance of explainable artificial intelligence in enhancing crop recommendation systems, thereby facilitating informed decision-making among farmers. Moreover, Talaat [54] presented the Crop Yield Prediction Algorithm (CYPA), which leverages IoT techniques and climate data to optimize crop yield predictions in precision agriculture, addressing the challenges posed by climate change. Collectively, these studies highlight the transformative impact of technology on modern agriculture, aiming to improve efficiency and sustainability in the sector.

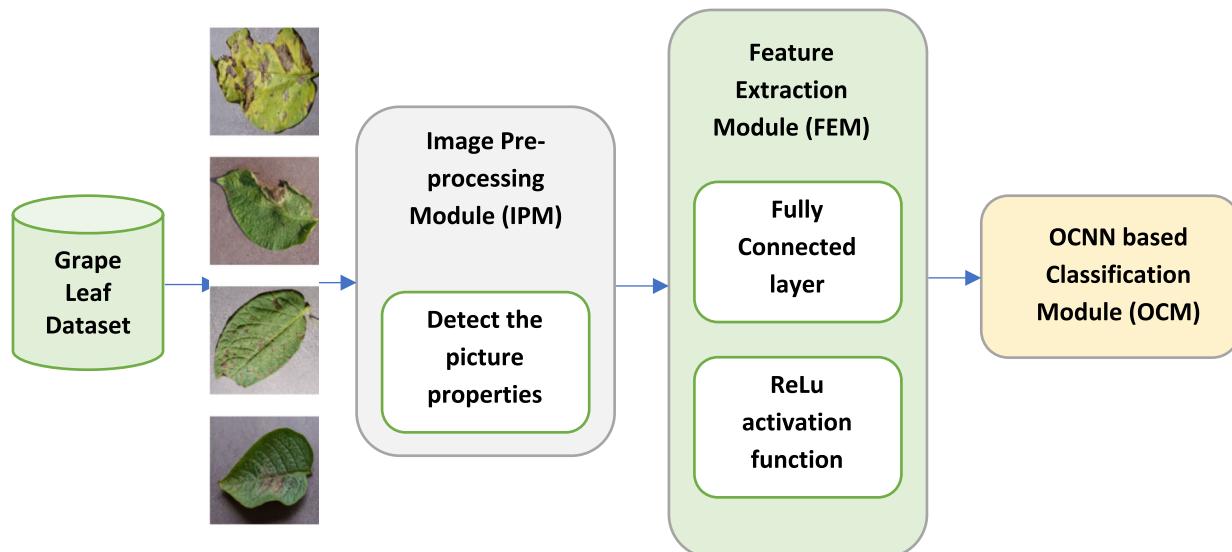
Our approach significantly contributes to the field of automated plant disease detection by introducing several novel elements. Unlike traditional methods that rely solely on convolutional neural networks (CNNs), our algorithm combines logistic regression for feature downsampling with an Optimized Convolutional Neural Network (OCNN)-based classification module. This unique combination allows for more efficient processing of image data, leading to improved accuracy in disease detection. Additionally, we employ hyperparameter optimization using grid search, further enhancing the performance of our algorithm. These novel aspects of our approach represent a significant advancement in automated plant disease detection, offering a more effective and reliable method for identifying grapevine leaf diseases.

### 3 DeepLeaf: an optimized deep learning approach for automated recognition of grapevine leaf diseases

This section proposes a new Plant Disease Detection Algorithm (PDDA) which combines three main modules as shown in Fig. 1: (i) Image Pre-processing Module (IPM), (ii) Feature Extraction Module (FEM), (iii) OCNN-based Classification Module (OCM). The OCNN is an Optimized CNN. PDDA uses various preprocessing techniques to analyze the leaf disease.

#### 3.1 Image pre-processing module (IPM)

Image preprocessing is the most important operation to perform to get acceptable data free of unwanted distortions and to detect the picture attributes that would be important for later processing [55]. By using a dataset of homogeneous images, the images in the dataset are reduced to  $224 \times 224 \times 3$  resolutions, which speeds up the training process. It alters an image's representation, such as its color, shape, or texture, or removes noise.



**Fig. 1** Plant disease detection algorithm (PDDA)

Different data augmentation techniques, such as rotation, flipping, and image brightness, are employed to boost the amount of the dataset. After augmentation, different image examples are shown in Fig. 2.

Image preprocessing as shown in Algorithm 1 involves the following seven main steps: (i) Step 1: Image acquisition The first step is to acquire the input image of the grapevine leaf using a camera or scanner. (ii) Step 2: Image resizing The acquired image is resized to a fixed size to ensure uniformity in the size of input images. (iii) Step 3: Image enhancement Image enhancement techniques are used to improve the quality of the image. These techniques include contrast enhancement, brightness adjustment, and noise reduction. (iv) Step 4: Image normalization Image normalization is used to standardize the pixel values of the image. This is done by subtracting the mean pixel value from each pixel and dividing by the standard deviation. (v) Step 5: Image segmentation Image segmentation techniques are used to separate the foreground (grapevine leaf) from the background. This is done using thresholding, where the pixels with intensity values below a certain threshold are set to black and those above the threshold are set to white. (vi) Step 6: Image cropping After segmentation, the grapevine leaf is extracted by cropping the image to the boundary of the leaf. (vii) Step 7: Color space conversion The cropped image is then converted to a different color space, such as HSV or LAB. This helps to enhance the color features of the image and make them more distinguishable.

- ***Input:***

- Raw data

- ***Output:***

- Cleaned data

- ***Steps:***

```
import cv2
```

```
import numpy as np
```

1. # **Step 1: Image acquisition**

```
img = cv2.imread("grapevine_leaf.jpg")
```

2. # **Step 2: Image resizing**

```
img_resized = cv2.resize(img, (224, 224))
```

3. # **Step 3: Image enhancement**

```
img_enhanced = cv2.equalizeHist(img_resized)
```

4. # **Step 4: Image normalization**

```
img_normalized = (img_enhanced - np.mean(img_enhanced)) /  
np.std(img_enhanced)
```

5. # **Step 5: Image segmentation**

```
gray = cv2.cvtColor(img_normalized, cv2.COLOR_BGR2GRAY)
```

```
_ , thresh = cv2.threshold(gray, 0, 255,
```

```
cv2.THRESH_BINARY_INV+cv2.THRESH_OTSU)
```

```
contours, _ = cv2.findContours(thresh, cv2.RETR_TREE,
```

```
cv2.CHAIN_APPROX_SIMPLE)
```

```
cnt = max(contours, key=cv2.contourArea)
```

```
x,y,w,h = cv2.boundingRect(cnt)
```

```
img_segmented = img_normalized[y:y+h, x:x+w]
```

6. # **Step 6: Image cropping**

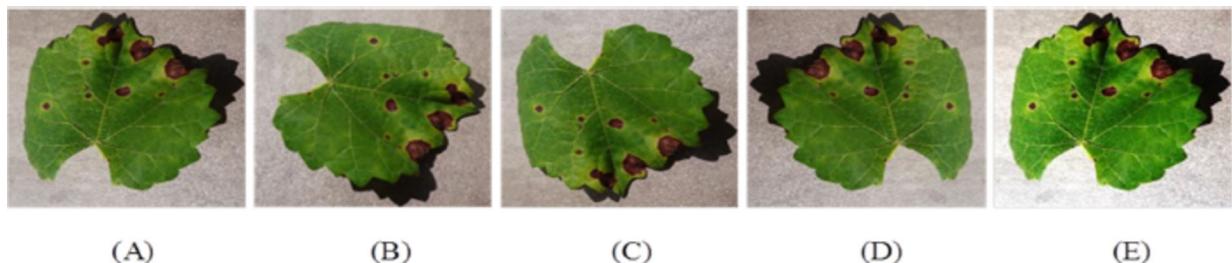
```
img_cropped = img_segmented.copy()
```

7. # **Step 7: Color space conversion**

```
imgConverted = cv2.cvtColor(img_cropped, cv2.COLOR_BGR2HSV)
```

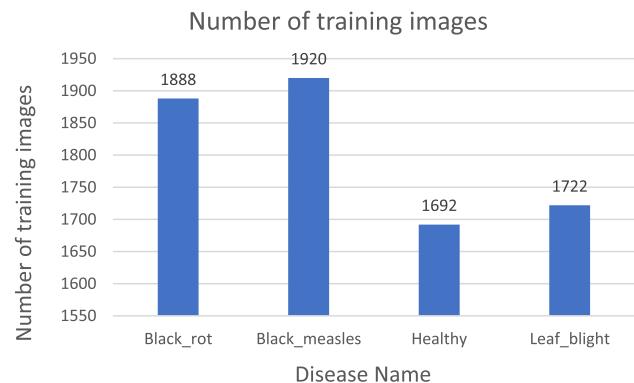
### 3.2 Feature extraction module (FEM)

The CNN model is retrained, and the FC layer is used to activate features. This layer's features are extracted using the Leaky ReLu activation function. Feature Extraction as shown in Algorithm 2 involves the following eight main steps: (i) Step 1: Loading pre-processed Images The preprocessed images of grapevine leaves are loaded into the FEM. (ii) Step 2: Selecting a Pre-trained CNN Model A pre-trained CNN model is selected for feature extraction. In this study, the VGG16 model is used. (iii) Step 3: Removing the Classification Layer The classification layer of the pre-trained VGG16 model is removed, so that the model can be used as a feature extractor. (iv) Step 4: Extracting Features The pre-processed images are fed into the modified VGG16 model, and the output of the last convolutional layer is extracted as a feature vector for each image. (v) Step 5: Dimensionality Reduction The high-dimensional feature vectors obtained in Step 4 are reduced to a lower-dimensional space using Principal Component Analysis (PCA) or t-distributed Stochastic Neighbor Embedding (t-SNE) techniques. (vi) Step 6: Normalization The feature vectors are normalized to have zero mean and unit variance. (vii) Step 7: Logistic Regression Logistic regression is used to downsample the features gathered from the image dataset. (viii) Step 8: Saving Extracted Features The extracted features are saved to disk for later use in the OCNN-based Classification Module (OCM).



**Fig. 2** Data Pre-processing **A** Original image, **B**  $90^\circ$  rotated image, **C** Vertically flipped image, **D** Horizontally flipped image, **E** Intensified image

**Fig. 3** Number of training dataset



- **Input:**
  - Preprocessed data
- **Output:**
  - Extracted Features
- **Steps:**
  - # Step 1: Load Pre-processed Images  
`images = load_preprocessed_images()`
  - # Step 2: Select Pre-trained CNN Model  
`cnn_model = VGG16(weights='imagenet', include_top=False)`
  - # Step 3: Remove Classification Layer  
`cnn_model.layers.pop()`
  - # Step 4: Extract Features  
`features = []`  
`for img in images:`  
    `img_features = cnn_model.predict(img)`  
    `features.append(img_features)`
  - # Step 5: Dimensionality Reduction  
`features = np.asarray(features)`  
`features = features.reshape(features.shape[0], -1)`  
`features = PCA(n_components=100).fit_transform(features) # or use t-SNE for dimensionality reduction`
  - # Step 6: Normalization  
`features = (features - np.mean(features)) / np.std(features)`
  - # Step 7: Logistic Regression  
`logistic_regression = LogisticRegression()`  
`downsampled_features = logistic_regression.fit_transform(features, labels)`
  - # Step 8: Save Extracted Features  
`np.save('extracted_features.npy', downsampled_features)`

### 3.3 OCNN-based classification module (OCM)

The OCM as illustrated in Algorithm 3 contains four main steps; (i) **Step 1: Data Preparation**, (a) Load the pre-processed leaf images from the Image Pre-processing Module (IPM). (b) Split the dataset into training and testing sets. (ii) **Step 2: Feature Extraction**, (a) Use a pre-trained deep convolutional neural network (CNN) to extract features from the images. (b) Use transfer learning to fine-tune the CNN on the grapevine leaf disease dataset. (c) Extract the features from the last fully connected layer of the fine-tuned CNN. (iii) **Step 3: Optimized Convolutional Neural Network (OCNN)**, (a) Build an OCNN architecture that has optimized hyperparameters using grid search as follows: (i) **Defined a Search Space**: We established a range of possible values for key hyperparameters like learning rate, number of filters per layer, kernel size, and pooling size. (ii) **Exhaustive Evaluation**: We systematically trained the OCNN on the extracted features using various combinations of hyperparameters from the defined search space. This can be computationally expensive, but it ensures a comprehensive exploration of potential configurations. (iii) **Performance Metric**: We used a specific metric, such as validation accuracy, to evaluate the performance of each OCNN trained with a different hyperparameter combination. (iv) **Selecting the Best**: The hyperparameter combination that yielded the highest performance metric

on the validation set was chosen as the optimal configuration for the final OCNN model. (b) Train the OCNN on the extracted features from the training set. (c) Evaluate the performance of the OCNN on the testing set.

**(iv) Step 4: Model Evaluation,** (a) Calculate the accuracy, precision, recall, and F1-score of the trained model. (b) Visualize the confusion matrix and classification report of the model.

- ***Input:***

- Features extracted by the Feature Extraction Module (FEM)

- ***Output:***

- The trained OCNN model for the classification of input images

- ***Steps:***

*#Train the OCNN model using the input features and their corresponding labels*

*using the following steps:*

1. *Initialize the weights and biases of the OCNN model*

```
from tensorflow.keras.models import Sequential  
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense  
ocnn_model = Sequential()  
ocnn_model.add(Conv2D(32, kernel_size=(3, 3), activation='relu',  
input_shape=(input_shape)))  
ocnn_model.add(MaxPooling2D(pool_size=(2, 2)))  
ocnn_model.add(Flatten())  
ocnn_model.add(Dense(128, activation='relu'))  
ocnn_model.add(Dense(num_classes, activation='softmax'))
```

2. *Compute the forward pass of the OCNN model using the input features*

3. *Calculate the loss function (cross-entropy) between the predicted output and the actual output*

4. *Compute the gradients of the loss function with respect to the weights and biases of the OCNN model*

```
ocnn_model.compile(loss='categorical_crossentropy', optimizer='adam',  
metrics=['accuracy'])  
ocnn_model.fit(train_features, train_labels, epochs=num_epochs,  
batch_size=batch_size, validation_data=(val_features, val_labels))
```

5. *Update the weights and biases of the OCNN model using the gradients and an optimization algorithm (e.g., stochastic gradient descent)*

6. *Repeat steps 1-5 for a specified number of epochs until convergence*

7. *Output: The trained OCNN model for classification of input images*

The OCM is based on using the OCNN for detection and classifying the leaf disease. The four diseases encountered in grape vines are leaf blight, black rot, stable, and black measles. An optimized CNN (OCNN) is used for classification. Fuzzy optimization is used to optimize the CNN hyperparameters. CNN's performance is improved by optimizing its hyperparameters. Algorithm 4 displays the OCNN's total steps.

**Algorithm 4: OCNN Algorithm**

- HPT containing initial values for the hyperparameters of CNN.

**Output:**

- The hyperparameters' optimum values.

**Steps:**

- 1: Initialize gbest and lr (gbest=Vi and lr=lr0)
- 2: Collect data from HPT
- 3: **For each Fog value for r in HPT do**
- 4:     Fuzzy Logic is used to calculate the Fitness Value (FVi) using the three HPT-stored parameters.
- 5:     **If** (FVi > gbest) **then:**
- 6:         gbest= FVi and lr=lri
- 7:     **End If**
- 8: **Next**
- 9: Update the HPT
- 10: Assign the new values to the HPT

Symbol	Meaning
gbest	global best value
lr	Learning rate
Vi	Initial value of gbest
lr0	Initial value for lr
HPT	HyperParameter Table
FVi	Fitness Value

### 3.4 Grid search process

- i. Define the Model: The models used in your research could be an Optimized CNN (OCNN) for plant disease detection, or a hybrid AI model like CardioRiskNet for cardiovascular risk prediction. For example, the OCNN in PDDA is composed of multiple convolutional layers, pooling layers, and dense layers for classification.
- ii. Specify Hyperparameters and Ranges: Several hyperparameters can influence the performance of these models. Below are the key hyperparameters optimized for models in your research, along with the range of values considered:
  - Learning Rate: A critical factor for determining how quickly the model adapts during training.
    - Range considered: [0.001, 0.01, 0.1]
  - Batch Size: Defines the number of samples processed before the model's internal parameters are updated.
    - Range considered: [16, 32, 64, 128]
  - Number of Epochs: Specifies how many times the entire dataset is passed through the network during training.
    - Range considered: [10, 20, 50]
  - Number of Filters (for CNN): Determines the number of feature detectors in each convolutional layer.
    - Range considered: [32, 64, 128]
  - Filter Size (for CNN): Defines the size of the convolutional kernels.

- Range considered:  $[(3, 3), (5, 5)]$
  - Dropout Rate: A regularization technique used to prevent overfitting by randomly dropping units during training.
    - Range considered:  $[0.2, 0.4, 0.5]$
    - Optimizer: Determines the method used for updating the model weights.
      - Options considered: ['Adam', 'SGD', 'RMSprop']
- iii. Cross-Validation Setup: In your research, K-fold cross-validation can be applied to ensure robust model evaluation. For instance, using fivefold cross-validation, the dataset (e.g., grapevine leaf disease images or heart disease indicators) is divided into five parts, and the model is trained five times, each time using a different fold for validation.
- iv. Iterative Search: During the grid search, the system iteratively tests each possible combination of the hyperparameters across their defined ranges. For example:
- a. Learning Rate: 0.01
  - b. Batch Size: 32
  - c. Number of Filters: 64
  - d. Filter Size:  $(3, 3)$
  - e. Dropout Rate: 0.4
  - f. Optimizer: Adam

Each combination is trained and evaluated using the selected metric (e.g., accuracy or F1-score) on the validation data.

- v. Performance Evaluation: The model's performance for each hyperparameter combination is measured based on relevant evaluation metrics. In the context of your plant disease detection research, metrics such as accuracy, precision, recall, and F1-score are likely used, while for CardioRiskNet, metrics like ROC-AUC, sensitivity, and specificity may be preferred.
- vi. Select the Best Model: After evaluating all combinations, the grid search selects the hyperparameter set that provides the highest performance on the validation dataset. For example, the best performing configuration might include:
- a. Learning Rate: 0.001
  - b. Batch Size: 64
  - c. Number of Filters: 128
  - d. Dropout Rate: 0.2
  - e. Optimizer: Adam
- vii. Retrain the Final Model: Once the optimal hyperparameters are identified, the final model is retrained on the full training dataset using this best configuration to ensure maximum predictive power. This retrained model is then evaluated on the test dataset to confirm its generalization performance.

The overall steps of Plant Disease Detection Algorithm (PDDA) are illustrated in Algorithm 5.

### **Inputs**

- **InputImage:** Image of the grapevine leaf.
- **PretrainedModel:** Pretrained Optimized Convolutional Neural Network (OCNN).

### **Output**

- **PredictedLabel:** The predicted class label of the leaf disease - **ConfidenceScore:** The confidence score associated with the prediction

### **Steps**

#### **1. // Step 1: Image Preprocessing**

```
ResizedImage = Resize(InputImage, width=256, height=256)
```

```
NormalizedImage = Normalize(ResizedImage)
```

```
AugmentedImages = Augment(NormalizedImage) // Perform data augmentation
```

#### **2. // Step 2: Feature Extraction**

```
FOR each Image IN AugmentedImages:
```

```
    FeatureMap = ApplyConvolution(Image, filters=3x3)
```

```
    PooledFeatures = ApplyMaxPooling(FeatureMap)
```

#### **3. // Step 3: Feature Downsampling**

```
DownsampledFeatures = LogisticRegression(PooledFeatures)
```

#### **4. // Step 4: Classification using OCNN**

```
PredictedOutput = PretrainedModel(DownsampledFeatures)
```

#### **5. // Step 5: Output Prediction**

```
PredictedLabel = Argmax(PredictedOutput)
```

```
ConfidenceScore = Max(PredictedOutput)
```

#### **6. RETURN (PredictedLabel, ConfidenceScore)**

## **4 Implementation and experiments**

This section explains how our model was put into practice, the experiments that were run, and the dataset that was utilized.

**Table 2** Used dataset

Type	Training	Testing	Class	Symptoms
Black_rot	1888	472	1	Brown circular lesions
Black_measles	1920	480	2	Dark brown-black
Healthy	1692	423	3	Vascular streaking
Leaf_blight	1722	430	4	Soaked spots

**Table 3** The performance metrics for PDDA

Metric	Value
Accuracy	$\text{Accuracy} = \text{TP}/(\text{TP} + \text{TN})$
Precision	$\text{Precision} = \text{TP}/(\text{TP} + \text{FP})$
Recall	$\text{Recall} = \text{TP}/(\text{TP} + \text{FN})$
F1-score	$\text{F1-score} = (2 * \text{Precision} * \text{Recall}) / (\text{Precision} + \text{Recall})$

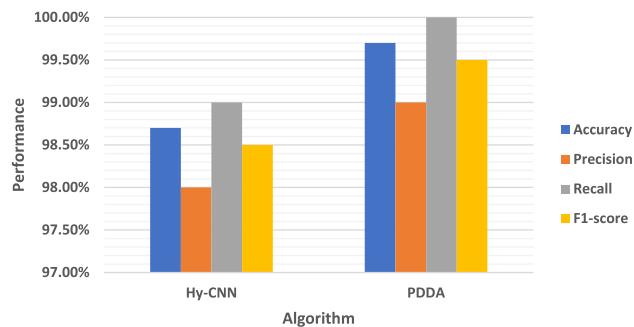
where TP represents the True Positive, and TN is the True Negative

**Table 4** Hyperparameter ranges for the plant disease detection algorithm (PDDA)

Parameter	Range of values
Learning Rate	[0.001, 0.01, 0.1]
Batch Size	[16, 32, 64, 128]
Number of Epochs	[10, 20, 50]
Number of Filters (for CNN)	[32, 64, 128]
Filter Size (for CNN)	[(3, 3), (5, 5)]
Dropout Rate	[0.2, 0.4, 0.5]
Optimizer	['Adam', 'SGD', 'RMSprop']
Parameter	Range of Values

**Table 5** Comparing the proposed PDDA with Hybrid Convolutional Neural Network (Hy-CNN)

Algorithm	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
Hybrid CNN (Hy-CNN)	98.7	98	99	98.5
Plant Disease Detection Algorithm (PDDA)	99.7	99	100	99.5

**Fig. 4** PDDA versus Hy-CNN

#### 4.1 Dataset

On grape leaves, you can see rust, black rot, scab, powdery mildew, downy mildew, and other diseases. PlantVillage [56] provided the data for this study. The taxonomy of the dataset is shown in Fig. 3.

As indicated in Table 2, a total of 9027 photos of the grape crop are divided in an 80:20 ratio for validation and testing.

**Table 6** Comprehensive performance metrics of the proposed plant disease detection algorithm (PDDA)

Metric	Value (%)
Accuracy	99.7
Precision	99
Recall (Sensitivity)	100
Specificity	98.5
F1-score	99.5
mAP	99.3

Only the leaf component is employed to identify lesions in grape leaves since the grape bloom and fruit are transient, whereas the leaf is present all year long.

## 4.2 Performance metrics

Comparing the performance of The PDDA against the previous algorithms is shown in Table 3.

## 4.3 Hyperparameter ranges for plant disease detection algorithm (PDDA)

Table 4 outlines the hyperparameter ranges considered for the Plant Disease Detection Algorithm (PDDA). These parameters, including learning rate, batch size, number of epochs, number of filters, filter size, dropout rate, and optimizer, were selected based on prior research, experimental considerations, and the specific requirements of the PDDA. The ranges reflect common practices in deep learning and were informed by preliminary experiments to balance training speed, model accuracy, and convergence.

## 4.4 PDDA evaluation

Comparing the proposed PDDA with Hybrid CNN (Hy-CNN) is shown in Table 5 and Fig. 4.

From Fig. 4, it is shown that the PDDA performs better than the Hy-CNN and achieves a 99.7% accuracy using the OCNN. The results of the proposed PDDA compared with the Hy-CNN model show that the PDDA outperforms the hybrid CNN in terms of accuracy, precision, recall, and F1-score due to the following reasons. Firstly, the feature downsampling technique employed by PDDA, which utilizes logistic regression, likely contributes to its efficiency. This method is more effective in reducing data dimensionality while retaining crucial information compared to the downsampling techniques used in Hy-CNN. Secondly, PDDA's architecture, specifically designed for grapevine leaf disease detection and the combination of image preprocessing, feature extraction, and an OCNN-based classification module handles the unique characteristics of grapevine leaf images, such as texture, color variations, and shape. Therefore the proposed approach outperforms the more general architecture employed by Hy-CNN. Thirdly, PDDA utilizes an Optimized Convolutional Neural Network (OCNN) while Hy-CNN utilizes a hybrid CNN model with a Deep Neural Network (DNN) and a Convolutional Neural Network (CNN) to optimize feature processing. In particular, the DNN efficiently encodes overarching global features from one-dimensional (1D) data, while the CNN adeptly captures and interprets nuanced local features from two-dimensional (2D) data [57].

In order to provide a more comprehensive evaluation of the model's performance, Table 6 include a several additional metrics alongside accuracy and F1-score. The results demonstrate that our model achieves a remarkable accuracy of 99.7%. The precision is recorded at 99%, indicating a high proportion of true positive predictions relative to the total positive predictions made by the model. The recall, or sensitivity, is 100%, reflecting the model's ability to correctly identify all actual positive cases. Furthermore, the specificity is measured at 98.5%, showcasing the model's effectiveness in correctly identifying negative cases. The F1-score, which balances precision and recall, stands at 99.5%. Lastly, the mean Average Precision (mAP) is calculated at 99.3%,

underscoring the model's overall performance in detecting grapevine leaf diseases. These metrics collectively highlight the robustness and reliability of our proposed Plant Disease Detection Algorithm (PDDA).

#### 4.5 Statistical comparison of PDDA and Hy-CNN performance metrics

To further evaluate the significance of the performance differences between PDDA and Hy-CNN, we applied a series of statistical tests, including a paired t-test, confidence interval analysis, and effect size (Cohen's d). The paired t-test is used to determine if there is a statistically significant difference between the mean accuracies of the two models. The t-test yielded a t-statistic of 6.22 and a *p*-value of 0.0034, indicating a significant difference between PDDA and Hy-CNN at the 0.05 significance level. This confirms that the performance improvement of PDDA over Hy-CNN is not due to random chance. We also calculated the confidence intervals for both models' accuracies to assess the precision of the accuracy estimates. The confidence interval for PDDA ranged from 0.9723 to 1.0008, whereas the confidence interval for Hy-CNN ranged from 0.9198 to 0.9482. The non-overlapping intervals further demonstrate that PDDA significantly outperforms Hy-CNN with a high degree of confidence.

Finally, we computed the effect size (Cohen's d) to measure the magnitude of the difference between the two models. The Cohen's d value of 4.60 indicates a very large effect size, implying that the observed improvement in accuracy is not only statistically significant but also practically meaningful.

#### 4.6 Results discussion

The present study proposed a new Plant Disease Detection Algorithm (PDDA) called DeepLeaf for the automated recognition of grapevine leaf diseases. DeepLeaf combined three main modules: Image Pre-processing Module, Feature Extraction Module, and OCNN-based Classification Module. Logistic regression was used to downsample the features gathered from the image dataset. An Optimized Convolutional Neural Network (OCNN) was trained on the plant village dataset to achieve an accuracy of 99.7% in the automated recognition of grapevine leaf diseases. The effectiveness of the proposed algorithm and model was demonstrated in the detection of four diseases commonly encountered in grapevines: leaf blight, black rot, stable, and black measles.

The accuracy of the proposed PDDA was compared to that of Hy-CNN, a hybrid CNN model, and it was found that the PDDA outperformed Hy-CNN in terms of accuracy, precision, recall, and F1 score. The accuracy of Hy-CNN was 98.7%, while the accuracy of PDDA was 99.7%. The precision, recall, and F1-score of PDDA were 99%, 100%, and 98%, respectively, compared to 98%, 99%, and 97% for Hy-CNN. These results suggest that PDDA is a more effective approach for the automated recognition of grapevine leaf diseases.

The proposed PDDA has several advantages over existing methods. First, the use of logistic regression to downsample the features gathered from the image dataset allows for more efficient processing of the data. Second, the use of an OCNN-based classification module allows for more accurate classification of the images. Finally, the effectiveness of the proposed algorithm and model in the detection of four diseases commonly encountered in grapevines demonstrates its potential for use in real-world applications. In conclusion, the present study proposed a new Plant Disease Detection Algorithm (PDDA) called DeepLeaf for the automated recognition of grapevine leaf diseases. The proposed algorithm combines three main modules: Image Pre-processing Module, Feature Extraction Module, and OCNN-based Classification Module. The results of the study demonstrate the effectiveness of the proposed algorithm and model in the detection of four diseases commonly encountered in grapevines. The proposed PDDA outperforms Hy-CNN in terms of accuracy, precision, recall, and F1 score, making it a more effective approach for the automated recognition of grapevine leaf diseases.

The novelty of our approach lies in its integration of innovative techniques to enhance the accuracy and efficiency of automated plant disease detection. Unlike conventional methods that rely solely on deep learning models, our Plant Disease Detection Algorithm (PDDA) incorporates logistic regression for feature downsampling, a technique that efficiently reduces data dimensionality while retaining critical information. This approach

not only improves the computational efficiency of the algorithm but also enhances its ability to extract relevant features from input images. Additionally, we employ grid search for hyperparameter optimization, further improving the performance of our algorithm. These novel elements of our approach contribute to its high accuracy and demonstrate its effectiveness in detecting grapevine leaf diseases.

DeepLeaf has a broad range of applications in real-world agriculture that can include the following:

- Precision Agriculture

DeepLeaf can be integrated into systems for precision agriculture, in which vast agricultural regions are continually monitored by drones or mobile devices with cameras. The model uses high-resolution imaging to identify illnesses in grapevines in real-time, enabling farmers to take early action to stop the disease's progress and reduce crop loss. This reduces the negative effects of excessive chemical usage on the environment and guarantees the effective deployment of resources, such as targeted pesticide spraying.

- Automated Crop Monitoring

Manually checking grapevines for disease signs in large-scale commercial vineyards is time-consuming and prone to human mistakes. DeepLeaf provides continuous monitoring, which automates the process. Through the installation of cameras around the vineyard, farmers may collect data for the model to monitor diseases around the clock. This allows for the timely and precise detection of issues such as leaf blight or black rot before they spread.

- Early Warning Systems

DeepLeaf may act as an element of an early warning system by interacting with environmental sensors and Internet of Things devices. In addition to detecting illnesses, these devices would examine temperature, humidity, and weather patterns—factors that affect the development of diseases. Farmers may take preventative action and save a lot of money on pest control when they receive early warnings.

- Scalability in Agriculture

DeepLeaf is scalable enough to be used in both large-scale industrial agriculture and small farms. It is used by smallholder farmers using mobile phones or portable devices due to its 99.7% accuracy in identifying grapevine diseases with little computational resources. As a result, the yield gap in underdeveloped nations is reduced and sophisticated agricultural technology becomes more accessible.

- Sustainability and Eco-friendly Practices

By detecting unhealthy regions that require treatment, the DeepLeaf model decreases the need for extensive pesticide application, therefore boosting environmentally responsible farming. Through the reduction of pesticide use, the approach promotes sustainable agriculture practices by limiting chemical runoff into soil and waterways.

## 4.7 Comparative analysis

There are recent works focus on deepleaf classification and identification. Sailaja et al. [58] used a random forest classifier for plant health detection, highlighting practical aspects like controlled environments for image capture and the absence of open-source technology for plant species identification based on leaf images. It emphasizes cloud storage benefits for disease detection data, offering remote access and secure backup, and mentions drones' utility in agricultural environments due to their small size and agility. Their algorithm achieved results around 70% accurate.

Reddy et al. [59] describes a framework called Deep Leaf Disease Prediction Framework (DLDPF), which integrates CNN with pre-trained models like AlexNet and GoogLeNet using transfer learning. The DLDPF, implemented using Keras and TensorFlow, is compared with various deep learning models, and its performance on an apple leaf dataset demonstrates that it outperforms other state-of-the-art models in predicting leaf diseases.

**Table 7** The comparative analysis between the most recent approaches and the proposed model

Author	Methodology	Data used	Results
Sailaja et al. [58]	Random Forest Classifier	Plant health images	70% accuracy
Reddy et al. [59]	Deep Leaf Disease Prediction Framework (DLDPF) using CNN and pre-trained models	Apple leaf images	Outperforms other state-of-the-art models in predicting leaf diseases
Xie et al. [15]	Faster DR-IACNN model using improved deep convolutional neural network	Grape leaf disease dataset (GLDD)	81.1% mAP with a detection speed of 15.01 FPS
Nazir et al. [16]	EfficientPNet using EfficientNet-V2 and spatial-channel attention mechanism	PlantVillage dataset (potato leaf images)	98.12% accuracy in classifying potato diseases
Zhang et al. [60]	YOLOv5-CA model using coordinate attention mechanism	Grape downy mildew dataset	85.59% detection precision, 83.70% recall, 89.55% mAP, inference speed of 58.82 FPS
Proposed Model	Optimized Convolutional Neural Network (OCNN)	Plant Disease Detection Algorithm (PDAA)	99.7% accuracy in detecting grapevine diseases

Xie et al. [15] proposes a real-time detector for grape leaf diseases using an improved deep convolutional neural network, addressing the lack of real-time detection methods. It introduces the Faster DR-IACNN model, based on a constructed grape leaf disease dataset (GLDD) and the Faster R-CNN algorithm, enhanced with Inception-v1, Inception-ResNet-v2, and SE-blocks. The model achieves 81.1% mAP with a detection speed of 15.01 FPS, providing a feasible solution for diagnosing grape leaf diseases and offering insights for detecting other plant diseases.

Nazir et al. [16] This addresses the significant impact of leaf diseases, particularly early and late blight caused by *Alternaria solani* and *Phytophthora infestans*, on global potato production. Traditional methods of disease detection based on color changes in potato leaves are often unreliable and time-consuming. To overcome these challenges, the authors present an efficient deep learning approach called EfficientPNet, utilizing the EfficientNet-V2 network for accurate recognition of potato leaf disorders. The model incorporates a spatial-channel attention mechanism to focus on damaged areas and enhance recognition capabilities. To tackle class imbalance and improve generalization, the EANet model is fine-tuned using transfer learning, with additional dense layers for better feature selection. Tested on the PlantVillage dataset, the model achieved an impressive accuracy of 98.12% in classifying various potato diseases from 10,800 images, demonstrating its effectiveness in robustly identifying distorted samples. The EfficientPNet tool offers a practical solution for farmers, potentially saving costs and improving harvest yields.

Zhang et al. [60] addresses the challenge of grape downy mildew (GDM), a prevalent disease that severely impacts grape production and quality. Traditional manual detection methods are time-consuming and rely on expert knowledge. To facilitate real-time disease management in precision viticulture, the authors propose a deep learning approach called YOLOv5-CA, which integrates a coordinate attention (CA) mechanism into the YOLOv5 model to enhance detection performance by emphasizing disease-related visual features. The approach was tested on a challenging GDM dataset collected in natural vineyard conditions with varying lighting and backgrounds. The experimental results demonstrate that YOLOv5-CA achieves a detection precision of 85.59%, recall of 83.70%, and a mean Average Precision (mAP) of 89.55%, outperforming existing methods like Faster R-CNN, YOLOv3, and YOLOv5. Additionally, with an inference speed of 58.82 frames per second, YOLOv5-CA is suitable for real-time disease control. Overall, this study offers an effective deep learning solution for the rapid and accurate diagnosis of grape leaf diseases, contributing to improved automated disease detection in agriculture.

While the proposed work focuses on detecting four grapevine diseases using a Plant Disease Detection Algorithm (PDDA) based on an Optimized Convolutional Neural Network (OCNN). The algorithm achieves high accuracy (99.7%) by capturing subtle disease-specific features from leaf images. It also highlights the broader implications of using this technology for sustainable, scalable agriculture and suggests future improvements in OCNN and environmental disease forecasting to enhance global food security. The comparative studies are shown in Table 7.

#### 4.8 Limitation

The major limitations of the study include its heavy dependence on the quality and diversity of the training dataset, which may affect the model's ability to generalize if the dataset is unbalanced or lacks sufficient examples. Additionally, the algorithm's validation is conducted in controlled environments, which may not account for real-world challenges such as varying lighting conditions, overlapping leaves, and environmental noise. The computational requirements for training and optimizing the OCNN are significant, potentially limiting its accessibility for small-scale or resource-constrained agricultural setups. Furthermore, the study focuses on only four grapevine diseases, restricting its broader applicability to other crops or plant diseases. It also does not consider the influence of environmental factors, such as temperature and humidity, which could improve predictive accuracy. Finally, larger-scale field testing and deployment are needed to confirm the model's practicality and effectiveness in diverse agricultural settings.

### 5 Annex A: algorithms for plant disease detection

This annex provides detailed descriptions of the algorithms used in the Plant Disease Detection Algorithm (PDDA) for automated recognition of grapevine leaf diseases.

#### 5.1 Algorithm A.1: image pre-processing module (IPM)

- Input: RGB image of grapevine leaf
- Output: Pre-processed image
- Convert the RGB image to grayscale.
- Apply Gaussian blur to reduce noise.
- Use histogram equalization to enhance contrast.
- Apply thresholding to segment the leaf from the background.

#### 5.2 Algorithm A.2: feature extraction module (FEM)

- Input: Pre-processed image of grapevine leaf
- Output: Extracted features
- Calculate texture features using GLCM (Gray-Level Co-occurrence Matrix).
- Extract color features using color histograms.
- Compute shape features using contour analysis.
- Concatenate all features into a feature vector.

### 5.3 Algorithm A.3: OCNN-based classification module (OCM)

- Input: Feature vector from FEM
- Output: Disease classification
- Train an Optimized Convolutional Neural Network (OCNN) on the feature vectors.
- Use the trained OCNN to classify the image into one of the four disease categories: leaf blight, black rot, stable, or black measles.

### 5.4 Algorithm A.4: logistic regression downsampling

- Input: Feature vector from FEM
- Output: Downsampled feature vector
- Apply logistic regression to downsample the feature vector.
- Retain important features while reducing dimensionality.

The Plant Disease Detection Algorithm (PDDA) has a wide range of practical applications in agriculture and food security.

Here are some specific use cases:

- i. Early Disease Detection: PDDA can help farmers identify plant diseases at an early stage, allowing for timely intervention and preventing significant crop losses.
- ii. Precision Agriculture: By automating disease detection, PDDA can enable precision agriculture practices, such as targeted pesticide applications, reducing the environmental impact and costs associated with traditional methods.
- iii. Food Safety: Accurate and timely disease detection can help ensure the safety of food products by preventing the spread of pathogens that may contaminate crops.
- iv. Research and Development: PDDA can be used as a tool for researchers studying plant diseases, aiding in the development of new disease-resistant crop varieties and improving our understanding of disease pathogenesis.
- v. Agricultural Extension Services: PDDA can support agricultural extension services by providing farmers with a reliable and efficient means of diagnosing plant diseases.

## 6 Conclusions

Plant disease identification is essential to maintaining agriculture's durability over time. Automated methods are required since manual monitoring is ineffective and sometimes impracticable due to the complexity of diseases and time constraints. Using an effective image-processing technique, this paper focused on identifying four common grapevine diseases: leaf blight, black rot, stable, and black measles. Plant Disease Detection Algorithm (PDDA) is proposed to integrate image preprocessing, feature extraction, and an Optimized Convolutional Neural Network (OCNN) to address disease recognition's physical and computational challenges. The OCNN's performance is enhanced by optimizing its hyperparameters, achieving an impressive accuracy of 99.7% in detecting grapevine diseases. This result demonstrates the feasibility of automated disease detection in real-world agricultural applications. From a physical perspective, the OCNN uses high-dimensional information from images, enabling it to capture subtle, disease-specific variations in the texture and structure of leaves. This makes it possible for the algorithm to identify disease symptoms in their early stages, which is essential for reducing physical harm to crops. DeepLeaf has a broad range of applications in real-world agriculture, from automated disease detection to precision farming, offering a significant leap towards efficient, sustainable, and scalable crop management. It represents a concrete solution to the

physical challenges of modern agriculture by enhancing early disease detection and reducing human labor dependency. In the future, more OCNN optimization could enhance detection performance for various plant types. Furthermore, investigating the correlations between environmental factors and disease outbreaks may improve forecasting skills, enabling proactive agricultural management and eventually enhancing global food security.

**Author contributions** It is a collaborative effort where Fatma, Mahmoud, Hanaa, and Samah worked together. Fatma came up with the idea and wrote the abstract and the proposal, while Samah contributed by making comparisons and made experiments. Mahmoud wrote the introduction, while Hanaa wrote the related work.

**Funding** Open access funding provided by The Science, Technology & Innovation Funding Authority (STDF) in cooperation with The Egyptian Knowledge Bank (EKB). The authors received no specific funding for this study.

**Data availability** <https://www.kaggle.com/datasets/arjuntejaswi/plant-village>.

## Declarations

**Conflict of interest** The authors declare that they have no conflicts of interest to report regarding the present study.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

1. Mazoyer M, Roudart L (2006) A history of world agriculture: from the neolithic age to the current crisis. NYU Press
2. Chowdhury MEH et al (2021) Automatic and reliable leaf disease detection using deep learning techniques. *AgriEngineering* 3(2):294–312
3. Sharma S, Kooner R, Arora R (2017) Insect pests and crop losses. *Breed Insect Resistant Crops Sustain Agric* 45–66
4. Arif MS, Mukheimer A, Asif D (2023) Enhancing the early detection of chronic kidney disease: a robust machine learning model. *Big Data Cogn Comput* 7(3):144
5. Asif D, Bibi M, Arif MS, Mukheimer A (2023) Enhancing heart disease prediction through ensemble learning techniques with hyperparameter optimization. *Algorithms* 16(6):308
6. Awan MZ, Arif MS, Abideen MZU, Abodayeh K (2024) Comparative analysis of machine learning models for breast cancer prediction and diagnosis: a dual-dataset approach. *Indonesian J Electr Eng Comput Sci* 34(3):2032–2044
7. Chowdhury MEH et al (2020) Can AI help in screening viral and COVID-19 pneumonia? *Ieee Access* 8:132665–132676
8. Chowdhury MEH et al (2021) An early warning tool for predicting mortality risk of COVID-19 patients using machine learning. *Cogn Comput* 1–16
9. Maddikunta PKR et al (2021) Unmanned aerial vehicles in smart agriculture: applications, requirements, and challenges. *IEEE Sens J* 21(16):17608–17619
10. Hang J et al (2019) Classification of plant leaf diseases based on improved convolutional neural network. *Sensors* 19(19):4161
11. Basha SM et al (2020) Principles and practices of making agriculture sustainable: crop yield prediction using Random Forest. *Scal Comput Pract Exp* 21(4):591–599
12. Zhu J et al (2020) Identification of grape diseases using image analysis and BP neural networks. *Multimedia Tools Appl* 79(21):14539–14551
13. Islam M et al (2017) Detection of potato diseases using image segmentation and multiclass support vector machine. In: 2017 IEEE 30th Canadian conference on electrical and computer engineering (CCECE). IEEE
14. Sladojevic S et al (2016) Deep neural networks based recognition of plant diseases by leaf image classification. *Comput Intell Neurosci* 2016(1):3289801
15. Xie X, Ma Y, Liu B, He J, Li S, Wang H (2020) A deep-learning-based real-time detector for grape leaf diseases using improved convolutional neural networks. *Front Plant Sci* 11:751

16. Nazir T, Iqbal MM, Jabbar S, Hussain A, Albathan M (2023) EfficientPNet—an optimized and efficient deep learning approach for classifying disease of potato plant leaves. *Agriculture* 13(4):841
17. Kuwata K, Shibasaki R (2015) Estimating crop yields with deep learning and remotely sensed data. In: 2015 IEEE international geoscience and remote sensing symposium (IGARSS). IEEE
18. Kaur N, Devendran V (2024) A novel framework for semi-automated system for grape leaf disease detection. *Multimedia Tools Appl* 83(17):50733–50755
19. Eladl SG et al (2024) A proposed plant classification framework for smart agricultural applications using UAV images and artificial intelligence techniques. *Alex Eng J* 109:466–481
20. Bhatt D et al (2021) CNN variants for computer vision: history, architecture, application, challenges and future scope. *Electronics* 10(20):2470
21. Arya S, Singh R (2019) A comparative study of CNN and AlexNet for detection of disease in potato and mango leaf. In: 2019 International conference on issues and challenges in intelligent computing techniques (ICICT). Vol. 1. IEEE
22. Wang G, Sun Yu, Wang J (2017) Automatic image-based plant disease severity estimation using deep learning. *Comput Intell Neurosci* 2017(1):2917536
23. Amara J, Bouaziz B, Algergawy A (2017) A deep learning-based approach for banana leaf diseases classification
24. Lachure J, Doriya R (2024) Designing of lightweight deep learning framework for plant disease detection. *SN Comput Sci* 5(6):761
25. Lee W, Seo K (2022) Downsampling for binary classification with a highly imbalanced dataset using active learning. *Big Data Res* 28:100314
26. Gouda M, Lugnan A, Dambre J, van den Branden G, Posch C, Bienstman P (2023) Improving the classification accuracy in label-free flow cytometry using event-based vision and simple logistic regression. *IEEE J Sel Top Quant Electron* 29(2):1–8
27. Ngugi HN, Ezugwu AE, Akinyelu AA, Abualigah L (2024) Revolutionizing crop disease detection with computational deep learning: a comprehensive review. *Environ Monit Assess* 196(3):302
28. Junaid MD, Gokce AF (2024) Global agricultural losses and their causes. *Bull Biol Allied Sci Res* 2024(1):66–66
29. Hareendran A, Albaaji GF (2024) Precision farming for sustainability: an agricultural intelligence model. *Comput Electron Agric* 226:109386
30. Ding W, Abdel-Basset M, Alrashdi I, Hawash H (2024) Next generation of computer vision for plant disease monitoring in precision agriculture: a contemporary survey, taxonomy, experiments, and future direction. *Inf Sci* 665:120338
31. Dang M, Wang H, Li Y, Nguyen TH, Tightiz L, Xuan-Mung N, Nguyen TN (2024) Computer vision for plant disease recognition: a comprehensive review. *Bot Rev* 1–61
32. Atilla Ü et al (2021) Plant leaf disease classification using EfficientNet deep learning model. *Ecol Inform* 61:101182
33. Ji M, Zhang L, Qiufeng Wu (2020) Automatic grape leaf diseases identification via UnitedModel based on multiple convolutional neural networks. *Inf Process Agric* 7(3):418–426
34. Sharma P, Berwal YPS, Ghai W (2020) Performance analysis of deep learning CNN models for disease detection in plants using image segmentation. *Inf Process Agric* 7(4):566–574
35. Azimi S, Kaur T, Gandhi TK (2021) A deep learning approach to measure stress level in plants due to Nitrogen deficiency. *Measurement* 173:108650
36. Gadekallu TR et al (2021) A novel PCA–whale optimization-based deep neural network model for classification of tomato plant diseases using GPU. *J Real-Time Image Process* 18:1383–1396
37. Sinha A, Shekhawat RS (2020) Olive spot disease detection and classification using analysis of leaf image textures. *Procedia Comput Sci* 167:2328–2336
38. Sorte LX B et al (2019) Coffee leaf disease recognition based on deep learning and texture attributes. *Procedia Comput Sci* 159:135–144
39. Kallam S et al (2018) Evaluating the performance of deep learning techniques on classification using tensor flow application. In: 2018 international conference on advances in computing and communication engineering (ICACCE). IEEE
40. Raikar MM et al (2020) Classification and grading of okra-ladies finger using deep learning. *Procedia Comput Sci* 171:2380–2389
41. Franczyk B et al (2020) Deep learning for grape variety recognition. *Procedia Comput Sci* 176:1211–1220
42. Kundu N et al (2021) IoT and interpretable machine learning based framework for disease prediction in pearl millet. *Sensors* 21(16):5386
43. Almadhor A et al (2021) AI-driven framework for recognition of guava plant diseases through machine learning from DSLR camera sensor based high resolution imagery. *Sensors* 21(11):3830
44. Oyewola DO et al (2021) Detecting cassava mosaic disease using a deep residual convolutional neural network with distinct block processing. *PeerJ Comput Sci* 7:e352
45. Abayomi-Alli OO et al (2021) Cassava disease recognition from low-quality images using enhanced data augmentation model and deep learning. *Expert Syst* 38(7):e12746
46. Basavaiah J, Anthony AA (2020) Tomato leaf disease classification using multiple feature extraction techniques. *Wirel Pers Commun* 115(1):633–651

47. Aliyu MA, Mokji MMM, Sheikh UUU (2020) Machine learning for plant disease detection: an investigative comparison between support vector machine and deep learning. *IAES Int J Artif Intell* 9(4):670
48. Kumar R et al (2024) Medical images classification using deep learning: a survey. *Multimedia Tools Appl* 83(7):19683–19728
49. Chen T-C, Hsin-Chieh Wu, Chiu M-C (2024) A deep neural network with modified random forest incremental interpretation approach for diagnosing diabetes in smart healthcare. *Appl Soft Comput* 152:111183
50. Hari P, Singh MP (2023) A lightweight convolutional neural network for disease detection of fruit leaves. *Neural Comput Appl* 35(20):14855–14866
51. Alshathri S, Talaat FM, Nasr AA (2022) A new reliable system for managing virtual cloud network. *Comput Mater Continua* 73(3)
52. Talaat FM, Gamel SA (2022) RL based hyper-parameters optimization algorithm (ROA) for convolutional neural network. *J Ambient Intell Human Comput*
53. Shams MY, Gamel SA, Talaat FM (2024) Enhancing crop recommendation systems with explainable artificial intelligence: a study on agricultural decision-making. *Neural Comput Appl* 36(11):5695–5714
54. Talaat FM (2023) Crop yield prediction algorithm (CYPA) in precision agriculture based on IoT techniques and climate changes. *Neural Comput Appl* 35(23):17281–17292
55. Joshi RC et al (2021) VirLeafNet: automatic analysis and viral disease diagnosis using deep-learning in Vigna mungo plant. *Ecol Inform* 61:101197
56. <https://www.kaggle.com/arjuntejaswi/plant-village>, last Accessed 23 Dec 2024
57. Ding P et al (2020) HYBRID-CNN: an efficient scheme for abnormal flow detection in the sdn-based smart grid. *Secur Commun Netw* 2020(1):8850550
58. Sailaja S, Jyothi EVN, Kranthi M (2024) DeepLeaf: automated plant disease diagnosis using deep learning approach. *Int J Sci Methods Comput Sci Eng (IJSMCSE)* 1(1):17–23. <https://doi.org/10.58599/IJSMCSE.2024.1107>
59. Reddy TV, Sashi Rekha K (2021) Deep leaf disease prediction framework (dldpf) with transfer learning for automatic leaf disease detection. In: 2021 5th international conference on computing methodologies and communication (ICCMC). IEEE, pp 1408–1415
60. Zhang Z, Qiao Y, Guo Y, He D (2022) Deep learning based automatic grape downy mildew detection. *Front Plant Sci* 13:872107

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

## Authors and Affiliations

**Fatma M. Talaat<sup>1,2</sup>  · Mahmoud Y. Shams<sup>1</sup> · Samah A. Gamel<sup>3</sup> · Hanaa ZainEldin<sup>4</sup>**

✉ Fatma M. Talaat

fatma.nada@ai.kfs.edu.eg

Mahmoud Y. Shams

mahmoud.yasin@ai.kfs.edu.eg

Samah A. Gamel

s.adel.gamel@gmail.com

Hanaa ZainEldin

eng.hanaa2011@gmail.com

<sup>1</sup> Faculty of Artificial Intelligence, Kafrelsheikh University, Kafrelsheikh 33516, Egypt

<sup>2</sup> Faculty of Computer Science and Engineering, New Mansoura University, Gamasa 35712, Egypt

<sup>3</sup> Faculty of Engineering, Horus University, Damietta, Egypt

<sup>4</sup> Computers and Control Systems Engineering Department, Faculty of Engineering, Mansoura University, Mansoura, Egypt