



Spinach fungi guard: A deep learning-based software solution for swift detection and remediation of fungal diseases in spinach leaves

Rajesh Bose^a, Sandip Roy^{a,*}, Shrabani Sutradhar^b

^a Department of Computer Science & Engineering, JIS University, West Bengal, India

^b Department of Computational Sciences, Brainware University, West Bengal, India



ARTICLE INFO

Keywords:

Fungal diseases
Spinach leaf
Convolutional Neural Network (CNN)
Disease classification
Remedies/medications
GUI application

ABSTRACT

We have developed a software solution aimed at assisting farmers in quickly detecting and finding a cure for fungal diseases in spinach leaves. This solution utilizes Deep Learning techniques, specifically a Convolutional Neural Network (CNN), to effectively detect diseases caused by fungi in spinach leaves and suggest suitable remedies to the user. To train our model; we gathered a spinach fungi disease image dataset consisting of over 700 images categorized into six classes, including five diseases and one healthy category. Prior to model training, we resized the images and applied various image augmentation techniques to improve the robustness of the model. Using Keras, we constructed a sequential CNN model for disease classification. The model was trained on the dataset and evaluated on the validation set, achieving an impressive accuracy of 89.86 %. To provide an intuitive interface for end users, we implemented a PySide2-based GUI application that leverages the trained model to classify disease in spinach leaf images provided as input. Our software not only accurately classifies the disease but also suggests appropriate remedies or medications for the specific disease. Furthermore, it provides links to relevant products on various e-commerce sites, enabling users to conveniently purchase the required medications. This comprehensive solution empowers end users to analyze infected spinach leaf images, accurately classify diseases, and take necessary actions by applying appropriate remedies and acquiring the right medications. By swiftly detecting diseases and offering prompt remedies, our software aids in preserving the production of spinach and ensures farmers can effectively combat fungal diseases, ultimately benefiting the food, medicine, and skincare industries.

Introduction

Spinach is a highly valued green leafy vegetable cultivated throughout India, particularly during the cold season. It boasts a rich nutritional profile, containing iron, calcium, magnesium, copper, vitamins, antioxidants, and other essential nutrients [1]. Due to its economic significance, various diseases pose significant threats to spinach cultivation, particularly those caused by fungi. This paper focuses on five economically significant fungi-based diseases [2] that commonly infect spinach plants in India and other South-Asian countries. Traditionally, agricultural experts visually inspect spinach leaves on farms to detect diseases, which are costly, time-consuming, and prone to human errors. To address these challenges, a Deep Learning-based approach utilizing image analysis is proposed in this study. The approach involves taking an image of spinach leaves as input, applying necessary pre-processing techniques, and employing a Convolutional Neural Network (CNN) for

disease classification. This automated approach offers cheaper, easier, and faster disease detection while maintaining comparable accuracy to human experts. The dataset for training the CNN model, named "SFDNet," consists of images representing five leaf-based fungi diseases of spinach: Anthracnose, Cercospora leaf spot, Cladosporium leaf spot, Downy mildew, and Stemphylium leaf spot [3]. Additionally, a class representing healthy spinach leaves is included. Images for the dataset were collected from various sources on the internet and manually resized using an online tool [4]. The dataset was further augmented using an open-source tool [5] and cleaned to ensure quality. The dataset was then divided into training and validation sets. The CNN model, SFDNet, was implemented using Keras. It was trained on the classes in the dataset to create a classifier model, which was subsequently validated using the validation set [6–8]. The model with the highest validation accuracy was selected as the final model. To make the disease classification accessible to users, the model was deployed in a PySide2

* Correspondence author.

E-mail address: sandip.roy@jisuniversity.ac.in (S. Roy).

application. This application enables the classification of diseases and provides suggestions for remedies along with corresponding medication links [9–11]. This paper presents an automated system for identifying spinach fungi leaf diseases using a Deep Learning-based approach. The system offers advantages over traditional visual inspection methods, including cost-effectiveness, efficiency, and reduced human error. The proposed system encompasses the detection and classification of Anthracnose, Cercospora leaf spot, Cladosporium leaf spot, Downy mildew, and Stemphylium leaf spot diseases, providing a valuable tool for spinach farmers and agricultural experts in India and South-Asian countries.

Anthracnose

Spinach is susceptible to a fungal disease known as anthracnose, caused by the pathogen *Colletotrichum Pinaceae*. This disease spreads through wind-dispersed spores, as well as irrigation and rainfall that splashes the spores onto the plants. Anthracnose symptoms manifest as small circular lesions on both young and mature spinach leaves, initially appearing as dark olive in color [20]. Over time, these lesions grow larger and transition to a brown or tan hue, eventually becoming thin, papery, and causing the entire leaves to die off. The development of anthracnose is favored by specific environmental conditions. It thrives in high humidity environments where water accumulates around the leaves. Additionally, temperatures ranging between 26 °C and 30 °C provide optimal conditions for the disease to proliferate.

Cercospora leaf spot

It is one of the most important diseases affecting spinach, it is caused by the fungus *Cercospora beticola* whose spores are spread by rain, winds, insects, etc. and can penetrate the leaf directly through open stomata and survive between crop cycles in infected crop residues, weeds, and seed and in the soil for up to two years. Symptoms include numerous, initially small circular leaf spots having a pale brown to off-white center with a red margin. As the fungus sporulates, these leaf spots expand in size, merge, and turn gray, killing the leaves and resulting in extensive loss of foliage. Favourable conditions include high humidity and temperatures between 25 °C to 30 °C [21].

Cladosporium leaf spot

It is another leaf spot disease affecting spinach caused by the fungal pathogen *Cladosporium variabile*, which survives mainly within the seeds. Its spores are spread by winds, rain, etc. Symptoms include brownish and round leaf spots that are around 0.2 inches in diameter with velvety dark green spores developing later on as the disease develops. Adjacent spots may merge, forming irregular lesions. Favourable conditions include cool, humid environmental conditions with temperatures around 15 °C to 20 °C and relative humidity above 80 % [22].

Downy mildew

A common, contagious and destructive fungal disease affecting spinach which is caused by fungus-like organisms of the phylum Oomycota. The spores produced in the morning are dispersed by the wind throughout the day. Downy mildew begins with uneven yellow spots on top leaf surfaces and purplish-gray sporulation on the undersides of leaves. In later stages, these spots can enlarge and become tan and dry. Favourable conditions include cool and humid environments with temperatures around 15 °C to 21 °C [23].

Stemphylium leaf spot

A fungal leaf spot disease caused by some species of *Stemphylium* such as *Stemphylium botryosum*, *Stemphylium* sp and *Stemphylium*

beticola. Symptoms include light gray to tan green leaf spots 0.15 cm to 1.3 cm in size, the larger spots are irregular in shape, which dries up in the later stages and becomes papery in texture. Favourable situations include prolonged periods of leaf wetness (frequent rains) and moderate temperatures around 17 °C to 23 °C [24,25].

Literature review

Detecting and classifying diseases in plants, particularly in spinach leaves, is crucial for timely treatment and preventing crop failure. Numerous studies and systems have been introduced to understand and identify diseases in spinach leaves, with a focus on visual symptoms caused by fungal species [2]. Safety measures and detailed information regarding fungi diseases associated with spinach have been provided by Satyagopal. et al. [3] and Ekman et al. [12]. Various approaches have been proposed for spinach disease detection. Sankar et al. [13] presented an approach utilizing the k-means clustering algorithm and specific classification algorithms to achieve 85 % accuracy. Farmers can send spinach leaf images via email to an agency, which analyzes the images for diseases and provides precautionary measures. Christopher [14] introduced the use of Histogram Oriented Gradients (HOG) feature extraction and an Artificial Neural Network (ANN) for disease detection in spinach leaves. They achieved promising results using this approach. Cercospora is a common disease affecting spinach leaves and is contagious to humans, potentially leading to kidney stone formation. Ramkumar et al. [15] proposed a transfer-learning-based system using a ResNet-50 based Convolutional Neural Network (CNN) to identify Cercospora levels in spinach leaves. The system provides remedial measures accordingly. Expert systems have also been developed for spinach disease detection. Abu Al-Qumboz et al. [16] proposed an expert system built using CLIPS and the Delphi language, allowing farmers to input symptoms for analysis and diagnosis of spinach diseases. Recent research has focused on disease detection and classification of plant leaf images, often utilizing the PlantVillage dataset. However, this dataset does not include images of spinach-related diseases. Various studies have used CNN-based models trained on subsets of the PlantVillage dataset to achieve high validation accuracies for other crop diseases [17–21]. A survey conducted by Boulet et al. [22] concluded that CNNs provide significant benefits in the classification and detection of crop diseases. In terms of spinach species classification; CNN-based approaches have also been employed. Islam et al. [23] conducted a comparative study using models like Inception3, Xception, VGG19, and VGG16 to classify locally found spinach species in Bangladesh, achieving accuracies between 98.68 and 99.79 %. Sennan et al. [24] implemented image preprocessing techniques, including background removal, along with a novel CNN model to classify spinach leaves, achieving an overall accuracy of 97.5 %. Kamal et al. [25] employed fine-tuned DenseNet121, InceptionV3, and VGG19 models, enhancing accuracy by around 12 % on the PlantVillage dataset through image preprocessing techniques such as edge and morphological-based segmentation and background subtraction. While CNN-based approaches have shown promising results in the detection and classification of crop diseases, the absence of spinach-related diseases in the PlantVillage dataset necessitates tailored solutions for spinach disease detection. This paper aims to propose a cost-effective, accurate, and easy-to-use CNN-based spinach fungi disease classification and cure suggestion system which can be used by spinach cultivators. It is structured in the following manner. Section IV describes the proposed system and section V describes the methodology. Section VI shows the results and discussions regarding the system followed by Section VII and Section VIII which conclude the whole work and discuss the future scope of the work (Figs. 1–5).

Proposed system

A new system is proposed to address the challenges faced by farmers and experts in identifying plant diseases. The system utilizes a cutting-



Fig. 1. Anthracnose-affected spinach leaves.



Fig. 2. Cercospora leaf spots on spinach leaves.



Fig. 3. Cladosporium leaf spots on spinach leaves.



Fig. 4. Downy Mildew affected spinach leaves.



Fig. 5. Stemphylium leaf spots on spinach leaves.

edge CNN-based model to accurately classify images of spinach leaves affected by fungal diseases. This innovative solution aims to reduce the time and effort required for disease identification, while minimizing the potential for human error that can lead to incorrect diagnoses and crop losses. The proposed system is accompanied by a user-friendly application built using PySide2, allowing spinach cultivators to easily utilize its features. Through the application, users can upload images of spinach leaves and receive instant classification results, indicating the specific fungal disease affecting the plants. In addition to disease identification, the system also provides relevant information for diagnosis and offers recommendations for related products that can aid in the treatment or

prevention of the identified diseases. The system is divided into two distinct phases, as illustrated in Fig. 6. The first phase involves the CNN-based model, which has been trained on a large dataset of spinach leaf images affected by various fungal diseases. This model employs advanced image recognition techniques to accurately classify new images and determine the specific disease present. The model's high accuracy and efficiency ensure reliable results for users. In the second phase, the results obtained from the model are seamlessly integrated into the PySide2-based application. The application offers an intuitive interface where users can easily upload their spinach leaf images and receive real-time disease classification results. Upon receiving the results, users can access detailed information regarding the identified disease, including its symptoms, causes, and recommended treatment options. Furthermore, the application provides a comprehensive database of related products, such as fungicides or preventive measures, that users can explore to address the identified diseases effectively. This feature enhances the usability of the system by offering practical solutions to combat and manage plant diseases, ultimately aiding in the preservation and improvement of spinach crop production.

The proposed system introduces a novel approach to plant disease identification, specifically focusing on fungal diseases affecting spinach plants. By leveraging a CNN-based model and a user-friendly PySide2 application, the system aims to streamline the disease identification process, reduce errors, and empower spinach cultivators with the necessary information and tools to combat plant diseases effectively.

Model creation, training, and validation

RGB spinach leaf images belonging to five classes of diseases and the healthy class were downloaded and preprocessed. First, the images were reshaped to a height of 256 and a width of 256. To increase the dataset's size, image augmentation techniques were applied. Additionally, any undesirable images were removed from the dataset. The preprocessed dataset was then converted into a NumPy array and normalized for further processing. The dataset was split into training and validation sets, which would be used to train and validate a CNN-based model. To track the model's performance and save the best version, a model checkpoint callback was implemented. This callback would monitor the validation accuracy and save the model with the highest accuracy as the best version. Overall, the preprocessing steps involved reshaping, augmentation, and cleaning of the RGB spinach leaf images. The resulting dataset was then converted into a NumPy array, normalized, and split into training and validation sets. The model checkpoint callback ensured that the best version of the CNN-based model was saved based on the validation accuracy.

Model deployment through PySide2 application

The Python application is designed with a simple and user-friendly interface, as depicted in Fig. 7. It involves loading a saved model file into the application. The user provides an input leaf image, which is then converted to RGB, reshaped to a height and width of 256, and transformed into a NumPy array. The image is normalized before being used for prediction using the loaded model. The model classifies the input image into one of six disease classes. Once the prediction is made, the proposed application fetches information from a remedy and product database. It then displays relevant information about the disease, if any, along with its associated remedies and product links to the end user. The user interface (Fig. 7a) allows the user to open the application and browse for a spinach leaf image using the "Browse for Picture" button. After selecting the image, the user clicks the "Predict" button to initiate the disease classification process. The application displays the predicted disease class name and the confidence of the CNN model's prediction. Additionally, it provides a description of the predicted disease class. By clicking the "See More" button, the user can access a "Disease Details" window (Fig. 7b) that presents various information about the predicted

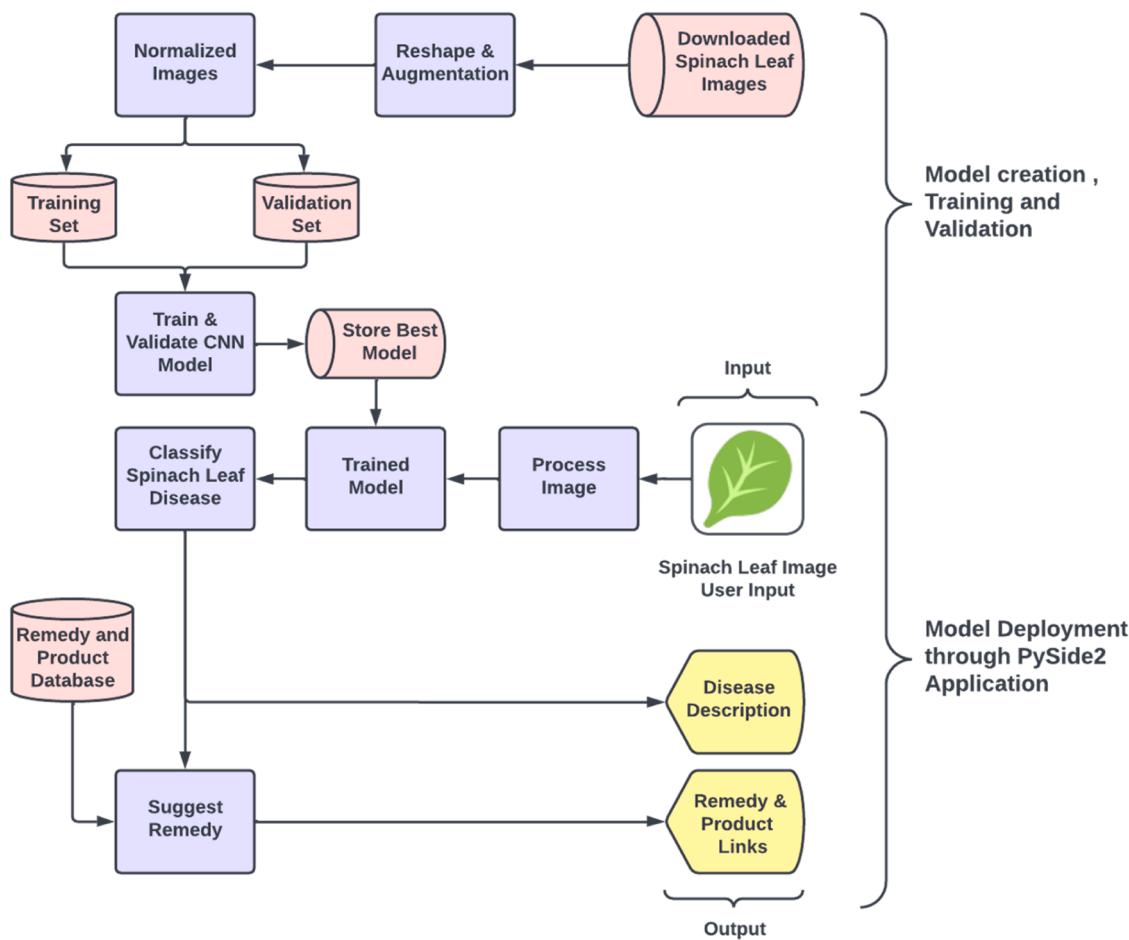


Fig. 6. Overview of the proposed system.

disease. This includes a reference picture, the disease's name, and the fungi/pathogens responsible for it, symptoms, and favorable conditions for the disease to spread and thrive. This window serves to educate the user about the disease. Furthermore, the application offers a "Remedy Details" window (Fig. 7c) when the user clicks the "Cure" button. This window provides information on available remedies for the predicted disease, if any are known. It also displays product links associated with the remedies. Clicking on a product link opens a new tab in the user's default browser, directing them to the specific product page. Overall, the application provides an intuitive and informative experience for users seeking to identify and address fungal leaf diseases in spinach leaves.

Methodology

The proposed system is created by implementing the subsequent modules.

Image acquisition

In the first stage a spinach leaf disease image dataset has been constructed by collecting images from the internet and nearby agricultural land with spinach crops planted. The dataset contains approximately 300 images belonging to six classes (before augmentation). The classes' names are Anthracnose, Cercospora leaf spot, Cladosporium leaf spot, Downy Mildew, Stemphylium leaf spot, and Healthy.

Image pre-processing and augmentation

Data pre-processing is a crucial step in implementing a model, and it

involves converting collected images into RGB format and reshaping them to a height of 256 pixels and width of 256 pixels. To enhance the diversity and quantity of data in our dataset, image augmentation techniques such as horizontal/vertical flip, rotation, and random zoom have been applied [26] using an image augmentation tool [5]. The resulting number of images and their distribution are outlined in Table 1.

Undesirable images have been eliminated from the dataset, and the remaining images have been converted into a NumPy array [27]. This array is then normalized to achieve a normal distribution of pixel intensities, thereby improving computational efficiency by restricting values between 0 and 1. The complete preprocessing and augmentation pipeline are illustrated in Fig. 8.

Model development

The preprocessed NumPy array of images is split into training and validation sets which are used to train and validate a proposed CNN-based model respectively. The CNN model is used for feature extraction and classification purposes and as it receives the pre-processed NumPy array as input, it has an input shape of $256 \times 256 \times 3$.

A CNN is a type of neural network which uses convolutional layers to extract features from images; the features are represented as a set of filters. The dense layers are used to classify images based on the output from the convolutional layers. The architecture of the CNN model to be used in the proposed system has been provided in Fig. 9.

Some of the key terms associated with our proposed CNN are explained below.

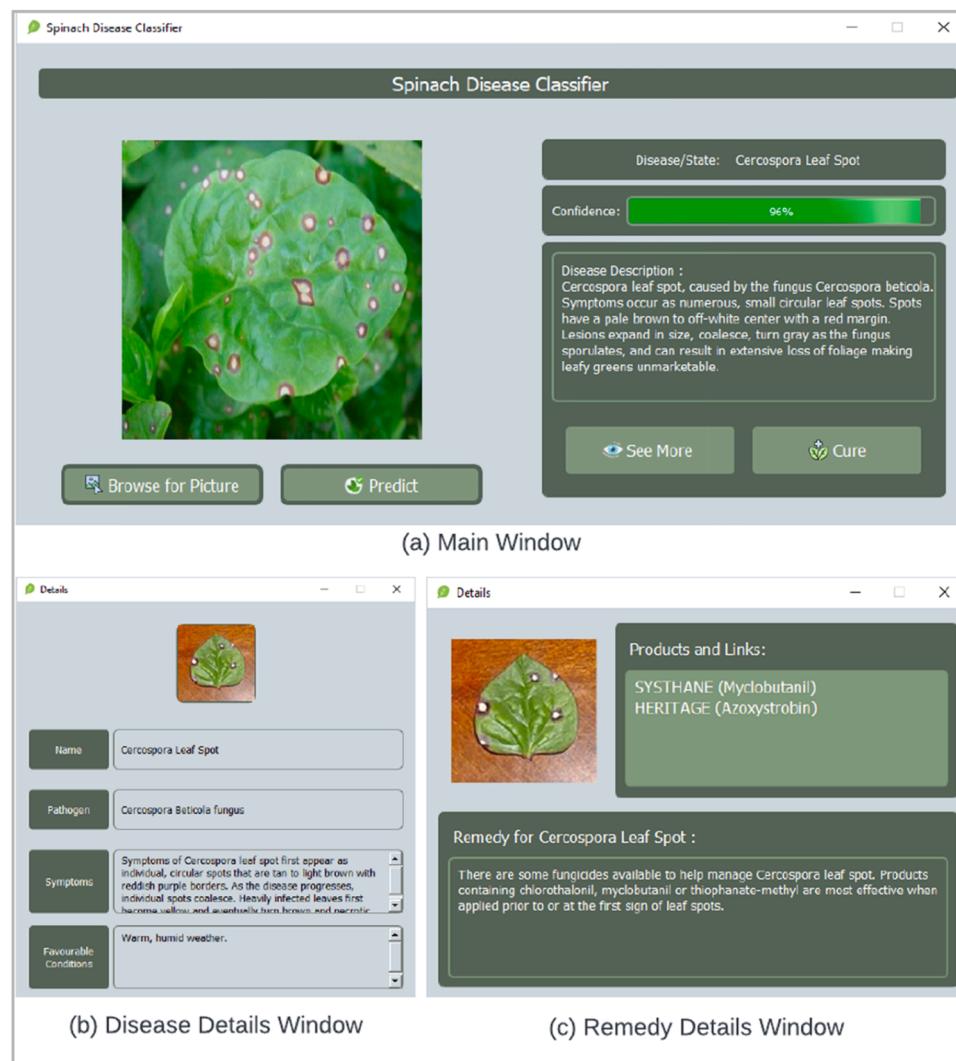


Fig. 7. Windows in the PySide2 Application (a) Application home screen predicting the “Cercospora” disease accurately, (b) Disease Details Window of predicted disease, (c) Remedy Details window of the application about the predicted disease.

Table 1
Image distribution in the dataset.

Disease Name	No. of Images
Anthracnose	130
Cercospora Leaf Spot	130
Cladosporium Leaf Spots	100
Downy Mildew	140
Healthy	140
Stemphylium Leaf Spot	110

- **Kernel** - The two-dimensional structures used to extract features from the images such as edges, corners, etc. [28].
- **Filters** - It is a three-dimensional structure of multiple kernels stacked together and represents the number of unique kernels which need to be generated. The depth of the filter corresponds to the number of channels [28].
- **Convolution** - It is the process of moving the kernels of a filter through the input image data and computing the sum of the dot product matrix for the sub-region of input image data. Convolution operation filters the information and generates the feature maps [28, 29]. For an image with dimension $n \times n$ and a filter with dimension $f \times f$, the dimension of the output is calculated using Eq. (1).

$$FM_{shape} = ((n-f+1), (n-f+1)) \quad (1)$$

where, FM is the output feature map of the convolution process.

- **Stride** - It indicates how many pixels the kernels are shifted across the input image data during the convolution procedure.
- **Padding** - It is the process of adding pixels of a specific intensity (typically 0) to the edges of an image while it is being processed by the CNN kernel. It is performed to prevent the output image from shrinking after the convolution operation.

In a CNN, where the padding size is “same” and the filters are of dimension $f \times f$, the padding shape can be computed using Eq. (2).

$$Padding_{shape} = \left(\left(\frac{f-1}{2} \right), \left(\frac{f-1}{2} \right) \right) \quad (2)$$

For a convolution with padding p , filter size $f \times f$, input image size, $n \times n$ and stride s the shape of the output feature maps can be calculated using Eq. (3).

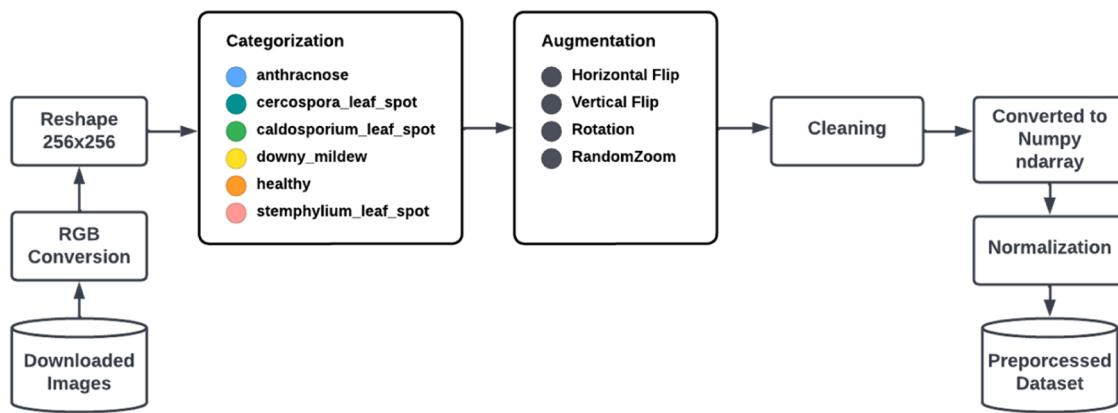


Fig. 8. Image pre-processing and augmentation method.

$$FM_{shape} = \left(\left(\frac{n+2p-f}{s} + 1 \right), \left(\frac{n+2p-f}{s} + 1 \right) \right) \quad (3)$$

- **Activation Function** - It is a function that determines whether or not a neuron should be activated based on the input values [30]. Some activation functions such as ReLU and Softmax have been used in the proposed model.

The rectified linear activation function, often known as ReLU, is a non-linear function that produces the input value if it is positive, otherwise, it produces 0. The ReLU activation function is calculated using Eq. (4).

$$ReLU(z) = \begin{cases} 0 & \text{if } z < 0 \\ z & \text{if } z \geq 0 \end{cases} \quad (4)$$

The Softmax activation function determines the relative likelihood of values in a given vector, which is proportional to the relative scale of all the values in the vector. The Softmax activation function is calculated using Eq. (5) where \vec{z} is the input vector, K is the number of classes in the classifier, and i is the current observation.

$$Softmax(\vec{z})_i = \frac{\exp(z_i)}{\sum_{j=1}^K \exp(z_j)} \quad (5)$$

- **Pooling** - It is a down sampling approach that involves sliding a two-dimensional filter across each channel of a feature map, resulting in a summary of the features within the filter's coverage zone [31]. The proposed model incorporates the max-pooling strategy. It entails picking the maximum value of an area indicated by the pooling filter size and generating a new output matrix with each element representing the maximum value of a region from the original image [32].
- **Flatten** - In order to send the multidimensional pooled feature maps to the fully connected layer, they have to be converted into a one-dimensional structure, the flatten function is utilized for this purpose.
- **Dense Layer** - It is a layer in which every neuron is linked to every neuron in the layer before it. It is used to implement the fully connected layer, which classifies the input images into different categories [33].
- **Dropout Regularization** - It is a regularization approach used to reduce overfitting in neural networks by randomly deleting certain nodes during training depending on a predetermined dropout rate [34].
- **Batch Normalization** - Batch normalization is a transformation performed between the layers of a Neural Network by which data flowing through a layer is rescaled and normalized over the current batch. This makes the training process faster and more stable [35].

We propose a CNN architecture that begins with an additional Augmentation layer to enhance the input images. This layer applies Random Flip, Random Rotation, and Random Zoom operations to augment the images. The augmented images serve as inputs to the first Convolutional layer, named "Conv Layer 1." This layer performs convolution with a kernel size of 3×3 , utilizing 32 filters and a stride size of 1. To maintain the input shape, we use "same" padding. The ReLU activation function is applied, and subsequently, a max pooling operation with a pool size of 3×3 is performed. The feature maps generated by Conv Layer 1 serve as inputs to the next layer, "Conv Layer 2." This layer also employs a 3×3 kernel size, 32 filters, and a stride size of 1, with "same" padding. The ReLU activation function is used, followed by a max pooling operation with a pool size of 3×3 . Moving forward, the subsequent layers, "Conv Layer 3" and "Conv Layer 4," employ a 3×3 kernel size with 64 filters, a stride size of 1, and "same" padding. The ReLU activation function is used, and max pooling with a pool size of 3×3 is applied to each of these layers. The output feature maps from Conv Layer 4 are flattened and passed as inputs to the Feed forward layer, which consists of two dense layers. The first dense layer has 256 neurons, followed by a second dense layer with 64 neurons. The ReLU activation function is used in both dense layers, and each layer incorporates dropout with rates of 0.25 and 0.20, respectively. Batch normalization is performed after each dense layer. Finally, the last layer of the proposed model is a dense layer with 6 neurons, utilizing the softmax activation function for multi-class classification. This enables accurate classification of spinach fungi leaf disease. Refer to Fig. 10 for a summary of the proposed model.

Model deployment

The proposed system utilizes TensorFlow to create a model for classifying spinach leaf diseases. The model is trained and validated in a Google Colab notebook cloud environment, and the best version of the model is saved as a ".h5" file using the Model Checkpoint callback. This file is then downloaded and loaded into a PySide2-based system for deployment. The system follows a data flow diagram, as shown in Fig. 11. The user provides an input image of a spinach leaf, which is then reshaped, converted into a NumPy array, and normalized. The pre-processed image is passed to the trained model, which predicts the type of fungi-based leaf disease that may be affecting the leaf. The predicted disease name is then used by the remedy product recommender to suggest appropriate remedies and related products. To improve clarity, here's a explanation:

The proposed system is designed to classify spinach leaf diseases using a TensorFlow-based model. The model creation, training, and validation processes take place in a Google Collab notebook cloud environment. The best-performing version of the model, based on validation accuracy, is saved as a ".h5" file using the Model Checkpoint

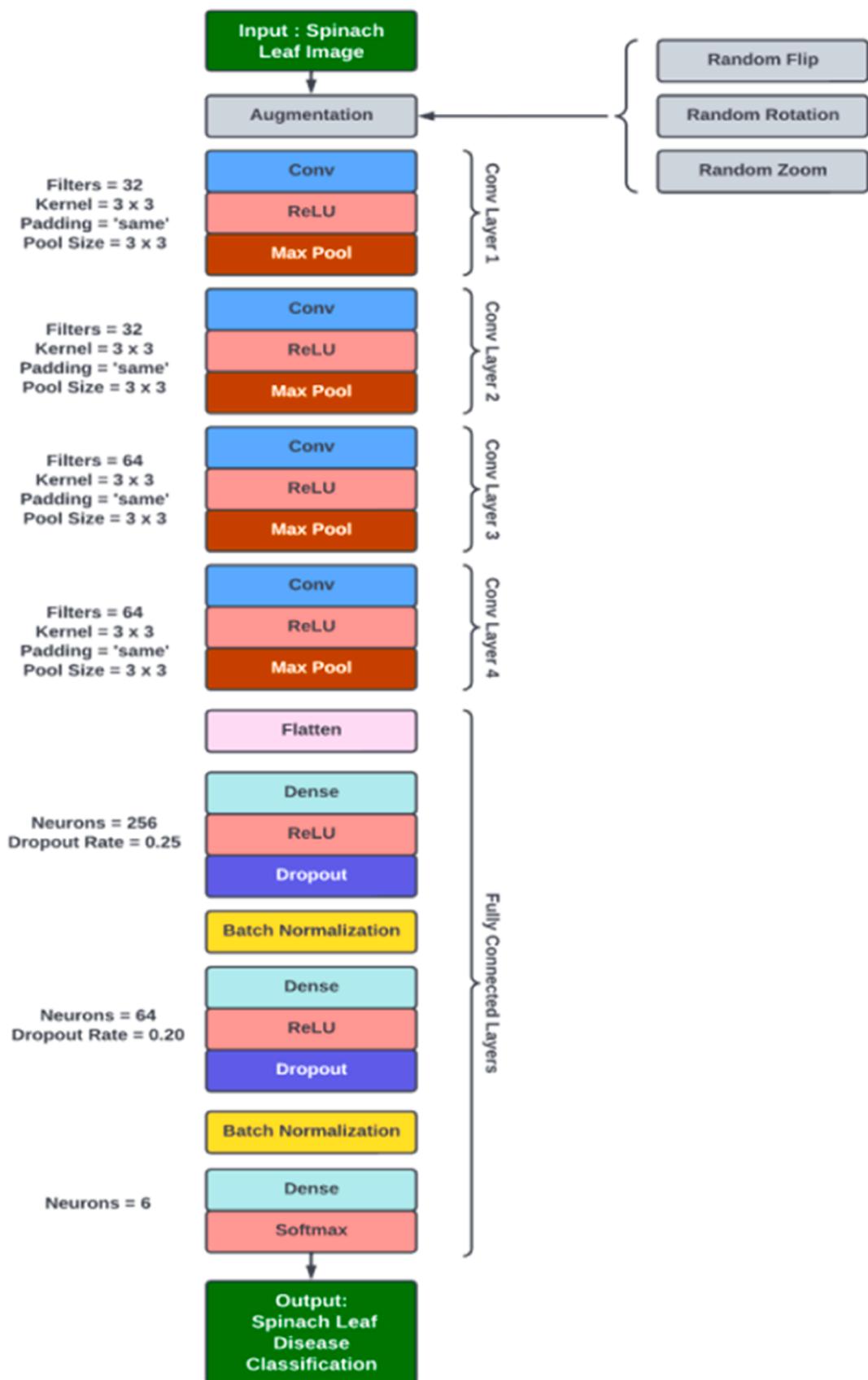


Fig. 9. Proposed CNN architecture.

Model: "SFDNet"

Layer (type)	Output Shape	Param #
<hr/>		
Augmentation_Layer (Sequential)	(None, 256, 256, 3)	0
Input_layer (InputLayer)	multiple	0
Conv_1 (Conv2D)	(None, 256, 256, 32)	896
max_pooling2d_4 (MaxPooling2D)	(None, 85, 85, 32)	0
Conv_2 (Conv2D)	(None, 85, 85, 32)	9248
max_pooling2d_5 (MaxPooling2D)	(None, 28, 28, 32)	0
Conv_3 (Conv2D)	(None, 28, 28, 64)	18496
max_pooling2d_6 (MaxPooling2D)	(None, 9, 9, 64)	0
Conv_4 (Conv2D)	(None, 9, 9, 64)	36928
max_pooling2d_7 (MaxPooling2D)	(None, 3, 3, 64)	0
flatten_1 (Flatten)	(None, 576)	0
Dense_1 (Dense)	(None, 256)	147712
dropout_2 (Dropout)	(None, 256)	0
batch_normalization_2 (BatchNormalization)	(None, 256)	1024
Dense_2 (Dense)	(None, 64)	16448
dropout_3 (Dropout)	(None, 64)	0
batch_normalization_3 (BatchNormalization)	(None, 64)	256
Dense_Output (Dense)	(None, 6)	390
<hr/>		
Total params:	231,398	
Trainable params:	230,758	
Non-trainable params:	640	

Fig. 10. The summary of the proposed model "SFDNet".

callback. This file is later downloaded and integrated into a PySide2-based system for practical use.

The system's functionality is depicted in Fig. 11, which illustrates the flow of data. The user provides an input image of a spinach leaf, which

undergoes reshaping, conversion into a NumPy array, and normalization. These preprocessing steps prepare the image for analysis. The processed image is then fed into the trained model, which leverages its learned knowledge to predict the type of fungi-based leaf disease

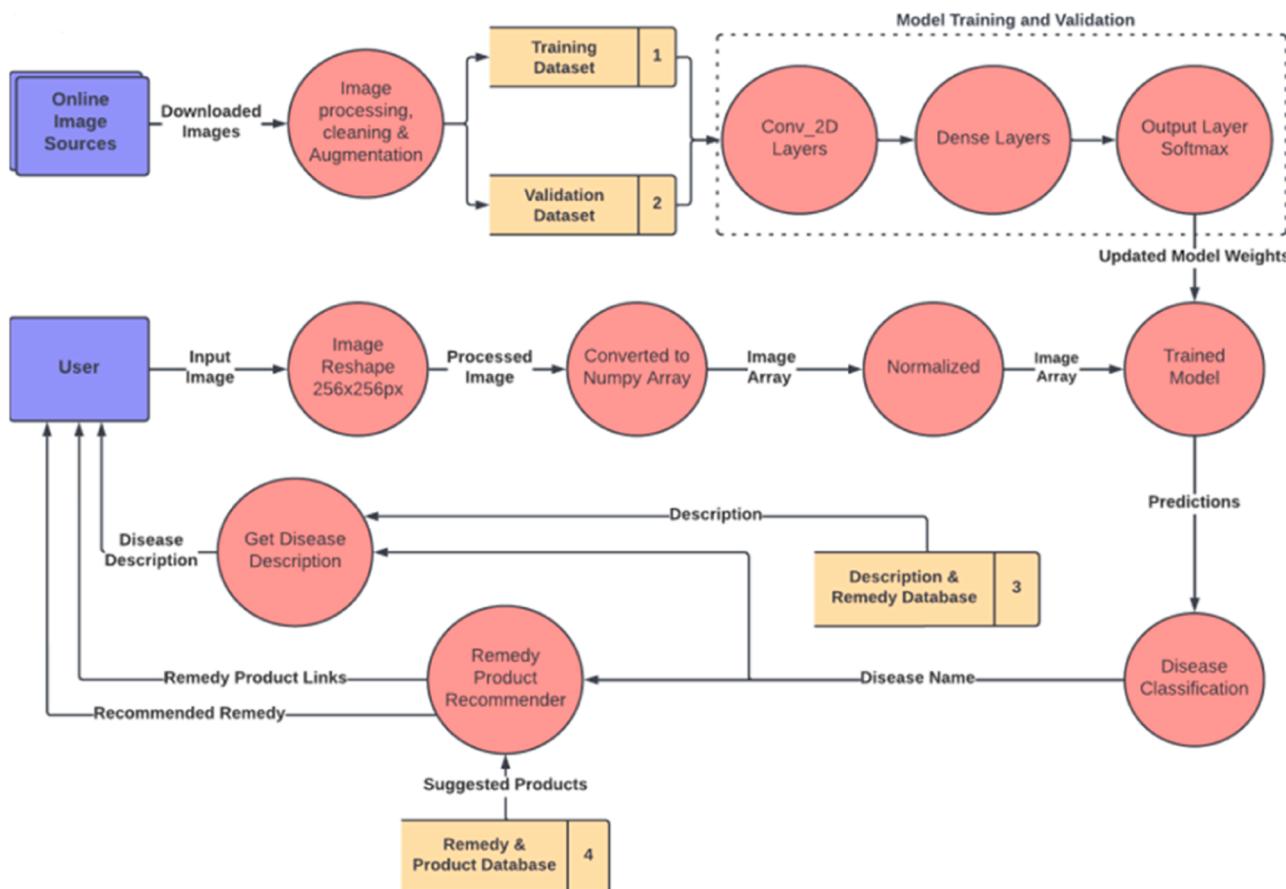


Fig. 11. Data flow diagram of the proposed system.

affecting the spinach leaf.

Upon obtaining the predicted disease name, the remedy product recommender comes into play. It utilizes this information to suggest appropriate remedies and related products from the database. This recommendation system aims to provide the user with relevant information to address the identified leaf disease effectively.

Result and discussion

This section contains a performance analysis of the classifier using training and validation sets of images created from the original dataset.

The proposed CNN model is named "SFDNet", and it uses a learning rate of 0.01, with a batch size of 32. The Softmax activation function and 'Adam' optimizer are used and the model is trained for 300 epochs. The hyperparameters of the model have been mentioned in Table 2.

Training and test data split

Using training and validation data ratios of 80:20, the proposed "SFDNet" model is trained and validated. The random state hyperparameter is set to 20. Table 2 shows the distribution of data into the

training set and validation set. It also shows the target label associated with each class of disease.

Accuracy of the proposed model

The degree to which predictions are correctly made is measured as accuracy, and the discrepancy between expected and actual results is the network's loss or error. The goal of the network is to maximize accuracy and minimize loss. The accuracy and loss graphs in Figs. 12 and 13 are obtained by training and validating the model for using the training and validation datasets.

The accuracy graph in Fig. 12 shows training and validation accuracy vs. Epoch. It can be seen from the figure that the training and validation accuracy of the model increases with the epochs. Our model "SFDNet" achieves a maximum validation accuracy of 89.86 % and does not overfit since the graph of validation accuracy obtained over the epochs is within range of the training accuracy. Another observation can be made that the validation accuracy graph is quite unstable compared to the training accuracy graph.

In the case of the training and validation loss vs. epoch graph shown in Fig. 13, the validation loss and training loss decrease with the epochs and reaches its near minimum value by the 300th epoch.

In a CNN model which works as a classifier, performance is evaluated using a confusion matrix. The matrix comprises the number of predictions produced by the model for True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN). The confusion matrix is used to compute a number of measures, including Accuracy, Precision, Recall, and F1-Score, which are used to assess the performance of the model.

For a given class, accuracy is calculated as the proportion of properly predicted observations to all observations. It can be calculated using Eq.

Table 2
Hyper parameters of "SFDNet" model.

Hyperparameter	Value
Learning Rate	0.001
Optimizer	Adam
Batch Size	32
Activation Function	Softmax
Epoch	300



Fig. 12. Model accuracy graph.



Fig. 13. Model loss graph.

(6).

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (6)$$

Precision is defined as the proportion of accurately predicted positive observations to all predicted positive observations. It can be calculated using Eq. (7).

$$\text{Precision} = \frac{TP}{TP + FP} \quad (7)$$

For any particular class, recall is the proportion of properly predicted positive observations to the actual number of positive cases. It can be calculated using Eq. (8).

$$\text{Recall} = \frac{TP}{TP + FN} \quad (8)$$

The weighted average of recall and precision, which accounts for both FP and FN, is known as the F1-Score. It is calculated using Eq. (9).

$$F1 = 2 \times \left(\frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \right) \quad (9)$$

The confusion matrix for spinach leaf disease prediction on the validation set has been provided in Fig. 14. The predicted classes and actual classes are represented via their respective target labels from Table 3.

The classification report of our model "SFDNet" on the validation set has been provided in Fig. 15. It contains the values of evaluation metrics for all classes of disease. It can be observed that the average accuracy of the CNN model utilized in the proposed system is 89.86 % with the average f1-score, recall, and precision being 0.90, 0.90, and 0.90 respectively.

Conclusion

In conclusion, this paper presents a novel approach for identifying economically significant fungi-based diseases in spinach leaves using

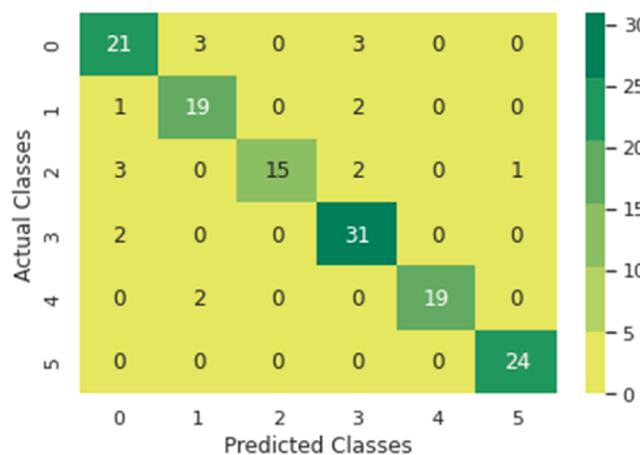


Fig. 14. Confusion Matrix of “SFDNet” model on validation set.

Table 3
Distribution of image data into training and validation set.

Disease Name	Total Images	For Training Set	For Test Set	Target Label
Anthracnose	130	103	27	0
Cercospora Leaf Spot	130	108	22	1
Cladosporium Leaf Spots	100	79	21	2
Downy Mildew	140	107	33	3
Healthy	140	119	21	4
Stemphylium Leaf Spot	110	86	24	5

deep learning techniques. The proposed system utilizes a Convolutional Neural Network (CNN) model that has been trained and validated on a dataset of spinach leaf disease images sourced from various online platforms. The CNN model achieves an impressive accuracy of 89.86 % on the validation set, demonstrating its effectiveness in accurately classifying spinach leaf diseases. To address the challenge of limited data availability, image augmentation techniques, Dropout layers, and Batch Normalization have been employed to prevent overfitting of the model on the training set. This ensures the generalization of the model's performance on unseen spinach leaf images. Furthermore, the CNN model has been seamlessly integrated into the Spinach Disease Classification and Cure Recommendation System application, developed using PySide2. This integration enables spinach cultivators to easily identify specific diseases caused by fungi in spinach leaves and obtain appropriate remedies for the identified diseases. By providing an efficient and accessible solution for disease identification in spinach cultivation, this system offers significant benefits to the agricultural domain. It empowers cultivators with a user-friendly tool to swiftly identify spinach leaf diseases, enabling timely interventions to prevent the spread of

infections and mitigate economic losses. The proposed system demonstrates promising results in identifying economically significant fungi-based diseases in spinach leaves. It represents a valuable contribution to the agricultural sector, enhancing disease management practices and ultimately supporting sustainable spinach cultivation. Future research can focus on expanding the dataset, incorporating additional disease categories, and exploring other deep learning models to further improve the accuracy and applicability of the system.

Future scope

In future work, there are several potential areas of improvement and expansion for the proposed spinach disease detection and cure recommendation system. These areas can further enhance the accuracy, functionality, and applicability of the software solution.

Firstly, expanding the dataset is crucial to improve the generalization and robustness of the trained model. By acquiring a larger and more diverse dataset of spinach leaf images, including additional disease categories and samples, the model can learn to classify a wider range of diseases accurately. Collaboration with agricultural research institutions, farmers, and online platforms can facilitate the collection of a comprehensive dataset.

Secondly, exploring other deep learning models and architectures can provide insights into alternative approaches for disease classification. While the Convolutional Neural Network (CNN) model used in this study achieved impressive accuracy, other models such as ResNet, Inception, or Efficient Net may yield even better results. Comparing and evaluating different models can help identify the most suitable architecture for spinach disease classification.

Thirdly, incorporating real-time disease detection and monitoring capabilities can enhance the practicality of the software solution. By leveraging computer vision techniques and integrating with imaging devices or drones, the system can provide farmers with instant feedback on disease outbreaks and enable continuous monitoring of spinach crops. Real-time data can enable timely interventions and prevent the spread of infections more effectively.

Additionally, incorporating geographical and climate factors into the disease detection system can improve the accuracy of disease diagnosis. By considering location-specific environmental conditions, such as temperature, humidity, and rainfall, the model can adapt its predictions based on the prevalent diseases in a particular region. This personalized approach can enhance the precision of disease identification and recommendation of suitable remedies.

Furthermore, integrating feedback and user interactions into the system can contribute to its continuous improvement. Allowing farmers to provide feedback on disease classifications and remedies can help refine the model and enhance its performance over time. User feedback can also be used to identify any limitations or challenges faced in real-world scenarios, enabling the development of targeted solutions.

Lastly, exploring partnerships with agricultural organizations, governmental bodies, or e-commerce platforms can expand the reach and impact of the software solution. Collaborations can facilitate the

	precision	recall	f1-score	support
anthracnose	0.82	0.85	0.84	27
cercospora_leaf_spot	0.87	0.91	0.89	22
cladosporium_leaf_spot	0.83	0.95	0.89	21
downy_mildew	0.91	0.88	0.89	33
healthy	1.00	0.86	0.92	21
stemphylium_leaf_spot	1.00	0.96	0.98	24
accuracy			0.90	148
macro avg	0.91	0.90	0.90	148
weighted avg	0.90	0.90	0.90	148

Fig. 15. Classification Report of “SFDNet” model on Validation Set.

integration of the system with existing agricultural management platforms, making it easily accessible to a larger number of farmers. Integration with e-commerce platforms can enable direct links to verified products and provide farmers with convenient access to recommended remedies.

In conclusion, future work on the spinach disease detection and cure recommendation system can focus on expanding the dataset, exploring alternative deep learning models, incorporating real-time monitoring capabilities, considering geographical and climate factors, integrating user feedback, and establishing partnerships for wider adoption. By addressing these areas, the software solution can continue to evolve and provide valuable support to farmers in combating fungal diseases in spinach cultivation.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The data that has been used is confidential.

References

- [1] A. Roughani, S.M. Miri, Spinach: an important green leafy vegetable and medicinal herb, in: The 2nd International Conference on Medicinal Plants, Organic Farming, Natural and Pharmaceutical Ingredients, 2019, pp. 1–6.
- [2] J.C. Correll, T.E. Morelock, M.C. Black, S.T. Koike, L.P. Brandenberger, F. J. Dainello, Economically important diseases of spinach, *Plant Dis.* 78 (7) (1994) 653–660.
- [3] D.P. Singh, Integrated Pest Management in Diverse Cropping Systems, CRC Press, 2023.
- [4] P. Begum, B. Fugetsu, Phytotoxicity of multi-walled carbon nanotubes on red spinach (*Amaranthus tricolor* L.) and the role of ascorbic acid as an antioxidant, *J. Hazard. Mater.* 243 (2012) 212–222.
- [5] A. Martínez-Sánchez, J. Guirao-Martínez, J.A. Martínez, P. Lozano-Pastor, E. Aguayo, Inducing fungal resistance of spinach treated with preharvest hormetic doses of UV-C, *LWT* 113 (2019), 108302.
- [6] H.O. Awika, K. Cochran, V. Joshi, R. Bedre, K.K. Mandadi, C.A. Avila, Single-marker and haplotype-based association analysis of anthracnose (*Colletotrichum dematium*) resistance in spinach (*Spinacia oleracea*), *Plant Breed.* 139 (2) (2020) 402–418.
- [7] M.M. El-Fawy, R.M. El-Sharkawy, K.A. Abo-Elyousr, Evaluation of certain *Penicillium* frequentans isolates against *Cercospora* leaf spot disease of sugar beet, *Egypt J Biol Pest Control* 28 (2018) 1–7.
- [8] A.M. Batson, K.A. Spawton, R. Katz, L.J. Du Toit, Cladosporium leaf spot caused by *Cladosporium* variable in winter high tunnel production of spinach (*Spinacia oleracea*) in Maine, United States, *Plant Dis.* 106 (8) (2022) 2260.
- [9] S.L. Kandel, B. Mou, N. Shishkoff, A. Shi, K.V. Subbarao, S.J. Klosterman, Spinach downy mildew: advances in our understanding of the disease cycle and prospects for disease management, *Plant Dis.* 103 (5) (2019) 791–803.
- [10] P.G. Falloon, L.M. Falloon, R.G. Grogan, Etiology and epidemiology of *Stemphylium* leaf spot and purple spot, *Phytopathology* 77 (1987) 407–413.
- [11] W.H. Wadlington, G.V. Sandoya-Miranda, C.F. Miller, J. Villegas, R.N. Raid, *Stemphylium* Leaf Spot in spinach: chemical and breeding solutions for this threatening disease in Florida, in: Proceedings of the Florida State Horticultural Society 131, Florida State Horticultural Society, 2018, pp. 151–158.
- [12] J. Ekman, L. Tesoriero, S. Grigg, Pests, diseases and disorders of babyleaf vegetables: a field identification guide, *App. Horticult. Res.* (2014) 1–62.
- [13] Akshara R Sankar, Sneha S, Disease detection in spinach leaves using image processing and machine learning, *Int. J. Innovat. Res. Electr. Electron. Instrum. Control Eng.* 8 (1) (2020) 33–36.
- [14] C.S. Christopher, Disease identification in spinach leaves, in: International Conference on Recent Advances in Energy-efficient Computing and Communication (ICRAECC), IEEE, 2019, pp. 1–7.
- [15] M.O. Ramkumar, S.S. Catharin, V. Ramachandran, A. Sakthikumar, Cercospora identification in spinach leaves through resnet-50 based image processing, *J. Phys. Conf. Ser.* 1717 (1) (2021), 012046. IOP Publishing.
- [16] M.N. Abu Al-Qumboz, S.S Abu-Naser, Spinach expert system: diseases and symptoms, *Int. J. Acad. Inf. Syst. Res. (IJAISR)* 3 (3) (2019) 16–22.
- [17] G. Shrestha, M. Das, N. Dey, Plant disease detection using CNN, in: IEEE Applied Signal Processing Conference (ASPCON), IEEE, 2020, pp. 109–113.
- [18] A. Tripathi, U. Chourasia, P. Dixit, V. Chang, Plant disease detection using sequential convolutional neural network, *Int. J. Distrib. Syst. Technol. (IJDST)* 13 (1) (2022) 1–20.
- [19] A. Ashqar, S. Abu-Naser, Image-based tomato leaves diseases detection using deep learning, *Int. J. Eng. Res.* 2 (12) (2019) 10–16.
- [20] I. Sarker, S. Karforma, P. Bhattacharya, R. Bose, S. Roy, Agronomy with IoT devices: the smart solution for detection of diseases of betel leaves, *Univ. J. Agricult. Res.* 11 (1) (2023) 98–109.
- [21] M. Sardogan, A. Tuncer, Y. Ozen, Plant leaf disease detection and classification based on CNN with LVQ algorithm, in: 3rd International Conference on Computer Science and Engineering (UBMK), IEEE, 2018, pp. 382–385.
- [22] J. Boulent, S. Foucher, J. Théau, P.L. St-Charles, Convolutional neural networks for the automatic identification of plant diseases, *Front. Plant Sci.* 10 (2019) 941.
- [23] M. Islam, N.J. Ria, J.F. Ani, A.K.M. Masum, S. Abujar, S.A. Hossain, Deep learning based classification system for recognizing local spinach, in: Advances in Deep Learning, Artificial Intelligence and Robotics: Proceedings of the 2nd International Conference on Deep Learning, Artificial Intelligence and Robotics, (ICDLAIR) 2020, Springer International Publishing, 2022, pp. 1–14.
- [24] S. Sennan, D. Pandey, Y. Alotaibi, S. Alghamdi, A novel convolutional neural networks based spinach classification and recognition system, *Comput. Mater. Continua* 73 (1) (2022) 343–461.
- [25] K. Kc, Z. Yin, D. Li, Z. Wu, Impacts of background removal on convolutional neural networks for plant disease classification in-situ, *Agriculture* 11 (9) (2021) 827.
- [26] A. Mikolajczyk, M. Grochowski, Data augmentation for improving deep learning in image classification problem, in: International Interdisciplinary PhD Workshop (IIPhDW), IEEE, 2018, pp. 117–122.
- [27] T.E. Oliphant, Guide to Numpy, 1, Trelgol Publishing, USA, 2006, p. 85.
- [28] K. Bai, A comprehensive introduction to different types of convolutions in deep learning, *Towards Data Sci.* 1 (2019) 1–2.
- [29] Khandelwal, R. (2020). Convolutional neural network: feature map and filter visualization. 1, 1–2.
- [30] S. Sharma, S. Sharma, A. Athaiya, Activation functions in neural networks, *Towards Data Sci.* 6 (12) (2017) 310–316.
- [31] N. Akhtar, U. Ragavendran, Interpretation of intelligence in CNN-pooling processes: a methodological survey, *Neural Comput. Appl.* 32 (3) (2020) 879–898.
- [32] M. Ahmadi, S. Vakili, J.P. Langlois, W. Gross, Power reduction in CNN pooling layers with a preliminary partial computation strategy, in: 16th IEEE International New Circuits and Systems Conference (NEWCAS), IEEE, 2018, pp. 125–129.
- [33] X. Glorot, Y. Bengio, Understanding the difficulty of training deep feedforward neural networks, in: Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, JMLR Workshop and Conference Proceedings, 2010, pp. 249–256.
- [34] A. Poernomo, D.K. Kang, Biased dropout and crossmap dropout: learning towards effective dropout regularization in convolutional neural network, *Neural Netw.* 104 (2018) 60–67.
- [35] B. Manifold, S. Men, R. Hu, D. Fu, A versatile deep learning architecture for classification and label-free prediction of hyperspectral images, *Nature Mach. Intell.* 3 (4) (2021) 306–315.