Archives available at journals.mriindia.com

International Journal on Advanced Computer Theory and Engineering

ISSN: 2319 - 2526

Volume 15 Issue 01s, 2026

GrapeScan - Grape Leaf Disease Detection

[1]Akhilesh Sharad Aher, [2]Pravin Subhash Sonawane, [3]Mayuresh Dagu Aher, [4]Prathamesh Nitin Sonawane, [5]Prof. D.S. Rajnor

[1 2 3 4 5] Department of Computer Engineering SNJB's K. B. Jain College of Engineering, Chandwad Maharashtra, India

Email: [1]aherakhilesh88@gmail.com, [2]pravinsonawane8459@gmail.com, [3]mayureshaher93074@gmail.com, [4]prathameshsonawane2706@gmail.com, [5]rajnor.dscoe@snjb.org

Peer Review Information

Submission: 08 Dec 2025

Revision: 25 Dec 2025

Acceptance: 10 Jan 2026

Keywords

Grapevine Disease Detection, Convolutional Neural Networks, Precision Agriculture, Plant Leaf Image Classification, Deep Learning

Abstract

Crop yield and quality of grapes are affected adversely by grapevine diseases. Manual inspection of grapes for health and diseases is inefficient due to high volume and potentially high error rates. The authors developed a system called GrapeScan using deep learning algorithms to classify photos of grape leaves into diseased or healthy categories. The system is based on a Convolutional Neural Network (CNN) trained on a publicly available dataset of grape leaves and is available for use in real-time via a web interface. The authors report results indicating that their system has high classification accuracy and low inference time, making it appropriate for use in precision agriculture and other practical farming applications.

Introduction

Grapes are one of the most significant and widely grown types of horticultural crops, with fresh grape consumption, wine making, and processed food as some of the ways grapes are being utilized economically should be noted. The quality and quantity of grape production will depend on the health of the vine, and how healthy are the grape leaves that are involved in the process of photosynthesis and nutrient regulation for the vine. In addition, the grape leaves can host many diseases, including Black Rot, Esca (or Black Measles), and Leaf Blight; these diseases can spread rapidly within a grape crop and result in major yield reductions if not detected and treated properly at an early enough date.

The way we are currently detecting grape leaf disease is primarily through manual visual inspection done by either farmers or agricultural specialists, which is effective for small amounts of crop but can be very time consuming, and have many subjective factors, and is not scalable for larger vineyards. Early symptoms of many diseases that affect grape crops may be very subtle and visually appearing similar, leading to possible misclassification and delay in ultimate diagnosis. The need is therefore for an automated system that can provide the ability to correctly and timely identify grape bead disease while avoiding the requirement for human interaction.

The advent of artificial intelligence and computer vision has paved the way for the emergence of an automated system for the detection of plant diseases through their leaves. The development of Convolutional Neural Networks (CNN) has allowed these systems to be developed based on automated image classification. CNN-based image classification systems do not rely on manually created features to classify images; rather, they rely on automatically learned features from raw image data, thus removing the need for manual feature creation. In addition, these features are able to provide greater robustness and accuracy than traditional hand-crafted features when identifying complex visual patterns associated with plant diseases.

To this end, the present research proposes GrapeScan, a deep learning–based system for detecting grape leaf diseases to assist farmers and agricultural professionals. The proposed system employs a CNN-based model to classify grape leaf images into healthy or diseased categories and to provide users with real-time predictions through an easy-to-use online interface. GrapeScan seeks to improve overall crop yield, reduce dependence on expert supervision, and enable farmers and other agricultural practitioners to use precision agriculture techniques by allowing timely, accurate detection of grape leaf diseases.

System Overview

The framework for GrapeScan Disease Detection follows a structured and modular approach to facilitate the accurate classification of grape leaf diseases through the use of Deep Learning. The framework operates as a multi-layered pipeline in which grape leaf images uploaded to the system are processed and utilized to make an accurate prediction of any disease present on the leaves with little or no involvement from the user.

The framework comprises of three principal Layers: the Input Layer, Processing Layer and Output Layer. Each Layer is responsible for a particular task or function in detecting the presence of disease on grape leaves. The design of a Layered Architecture also provides the benefits of scalability, efficient processing and meaningful interaction between all users.

The Input Layer (Image Acquisition): provides a platform for users to upload images of grape leaves collected with either a smartphone or a digital camera via an online site and into the framework. The uploaded files are subject to a validation process to determine if they are of an acceptable file type (i.e. JPEG, PNG, etc.), have an acceptable resolution and size before any processing occurs. After validation, the uploaded images are subjected to preprocessing techniques, such as resizing, normalizing and removing noise, to enhance the quality of the uploaded images and to ensure compatibility with the original training data used to construct the model. All image uploads will contain metadata entries such as the time of upload and any optional user data so that in future instances, the dataset may be augmented with new examples and data may be used for internal analyses.

The GrapeScan framework and Processing Layer (Image Analysis and Classification): In this layer, preprocessed images are sent through a trained Convolutional Neural Network (CNN) that is designed to separate out meaningful and unique features (spatial) from the images of grape vines and leaves. The CNN uses visual patterns like differences in color, odd-looking textures, and shapes of lesions to classify a leaf into pre-defined categories (Healthy, Black Rot, Esca (Black Measles), or Leaf Blight). The classification produced by the CNN contains a probability-based confidence score for each category, and this information is generated with the help of a SoftMax activation function. The backend programming of this layer was written using Python, TensorFlow/Keras and OpenCV to provide an efficient inference method as well as to allow the periodic retraining of the model to improve accuracy as new data is created.

The Output Layer (Result Generation and Visualization): takes the classification results and presents them back to the user through a web interface built using Flask. This layer provides the user with not only the predicted disease class but also with the confidence score of that prediction, and recommendations for basic prevention or maintenance. Additionally, the results of the predictions can be stored for future use in historical analysis and for generating reports, and the dataset can be used to improve the training of the models in the future. The interface can be accessed across multiple platforms (desktop, tablet, and mobile) and therefore is very practical and user-friendly.

GrapeScan's modular and layered architecture enables real-time detection of diseases, ease of use/maintenance, and scalability. The modular architecture allows for the possibility of future integrations with mobile applications, IoT-based gene-tracking systems, and multi-language user interfaces. In addition, the overall framework forms a robust foundation for extending the capabilities of deep learning algorithms to robust crops to promote intelligent and sustainable agriculture.

Methodology

The GrapeScan – Grape Leaf Disease Detection System's methodology integrates five main steps to create an all-inclusive workflow, which will give users real-time and highly accurate detection capabilities for grape leaf disease using just a single picture. At the core of the methodology are image preprocessing, feature extraction using deep learning, classification of the diseases using a CNN, and the generation of an automated report.

A. Data Acquisition and Preprocessing

In step one, "Data Acquisition and Preprocessing," users will take a picture of grape leaves with a smartphone or camera and upload it through an internet-based application. Before the picture is processed, GrapeScan will validate that the picture is in the correct format and has the proper resolution and quality needed to process it. All pictures uploaded for processing will then be resized to a standard resolution of 256 × 256 pixels so that they match the original training data used to create the model to ensure consistency.

Preprocessing operations will also include pixel normalization, so that all the intensity values in the picture are scaled; noise reduction to remove any background artifacts; and contrast adjustment to enhance the appearance of disease symptoms on the leaves. To aid in the generalization of the model and to decrease overfitting, data augmentation methods such as image rotation, scaling, horizontal flipping, and zooming will be employed.

B. Distinguishing Important Characteristics of Grape Leaves using Convolutional Neural Networks (CNNs).

In stage 2, after preprocessing, processed images will be inputted into the Convolutional Neural Network (CNN), where CNNs extract valuable characteristics of the grape leaf images automatically. CNNs use multiple layers of convolutional filters which automatically learn how to extract differentiable features including; visual aspects of color changes, textures, forms of lesions, and disease-specific spot distributions, from grape leaf images. Further, a rectified linear function (ReLU) is used as an activation function; this allows the network to model more sophisticated, complex patterns of grape leaf images. For example, through maximum pooling (the use of maximum pooling layers), maximum pooling reduces the amount of data processed spatially and uses only the salient features to maximize processing speed and improve model robustness.

C. Model Processing and Classifying Grape Leaf Disease

Once the model has extracted the important characteristics of the image, the model then flattens all the small images into a single image or vector and feeds it through the fully connected (dense) layer, which allows for high level reasoning (nonlinear) and finds patterns between classified images and disease types. Finally, the network uses SoftMax activation functions for the final classification of the grape leaf into either Healthy, Black Rot, Esca, or Leaf Blight.

Each uploaded picture will be carefully categorized into one of the four designated classes: Healthy, Black Rot, Esca (Black Measles), and Leaf Blight.

D. Fusion of Decision and Estimate Confidence

In order to create reliable forecasts, the system utilizes both estimation of confidence and regularization. The SoftMax Output generates probabilities for all class of disease where/ tells us what Class is being predicted, the maximum score represents our predicted class. In order to reduce the risk of overfitting and provide a more stable Model we employ various techniques during training, e.g. Dropout & extensive data augmentation. The provision of Confidence Estimates aids the user in determining how reliable their prediction may be and supports overall better Decision Making with Ag.

E. Visualization & Output of Results

The final stage of output is delivered back to the user as part of a web-interface built upon Flask. Herein the user will see not only what disease was predicted, how confident we were that it was correct, and basic recommendations for prevention/maintenance, but the backend will be logging the output of the prediction along with Metadata (e.g. time uploaded; history of prediction) for future use in expanding the dataset and re-training of the model. The user-interface is designed such as to be responsive, to allow for easy accessibility on all types of devices (desktops, tablets, and mobile phones) and is to be user-friendly for both Farmers and other Agricultural Professionals.

This method will provide a reliable way to automate and scale the process of detecting diseases on grape leaves. By combining a robust preprocessing of images with a deep learning approach in extracting features from the image, accurate classification of the images, and visualization of these images in an easy-to-use format, the GrapeScan System can provide its users with disease predictions for grape leaves that have a very high degree of accuracy, allowing precision agriculture practices to take place.

F: Approach to Modeling Plant Leaf Disease Classification

The classification of plant leaf disease can be represented as an image-to-image classification task. Let I be the input image of a grape leaf with RGB color format. The Convolutional Neural Network (CNN) will determine a mapping function Fθ(I), where θ are the weights of the neural network, so that the output of the function gives a label (or class) for the grape leaf image based on the type of disease present.

where

$C \in$ {Healthy, Black Rot, Esca, Leaf Blight}.

Let Xdenote the input feature map and Krepresent the convolution kernel. Each convolutional layer computes:

$$Y_{i,j}^{(l)} = f\left(\sum X_{i+m,j+n}^{(l-1)} \cdot K_{m,n}^{(l)} + b^{(l)}\right) \qquad (2)$$

where $f(\cdot)$is the ReLU activation function.

The final classification is performed using the SoftMax function:

$$P(y = k \mid x) = \frac{e^{z_k}}{\sum_{i=1}^{n} e^{z_i}} \qquad (3)$$

where z_k denotes the logit corresponding to class k.

The final prediction is obtained by selecting the class with the highest probability:

$$O = \arg\max(P) \qquad (4)$$

The model is trained to minimize the **categorical cross-entropy loss** between predicted and true class labels, ensuring accurate and reliable disease classification.

Through Python, GrapeScan has been developed and deployed via the GrapeScan system, utilizing the power of Machine Learning and Image Processing with a wide variety of flexibility through the use of Python (3) and the many libraries that support it. A Deep Learning Model has been developed and trained using both TensorFlow and Keras, while we have implemented OpenCV to facilitate preprocessing functions such as Normalization of an image, Resizing images, as well as the removal of noise from Photos. We have used the NumPy and Pandas libraries for our Data Handling and Numerical Computations within the GrapeScan Platform System.

Flask Framework is the backend of the GrapeScan System that allows API Routing, the connection between model inference to the front-end user interface, and manages requests/responses from users. The Frontend of the GrapeScan System design uses HTML5/CSS3/JavaScript to deliver a Responsive User Interface for both Image Uploading and displaying results on multiple platforms. An SQLite/MySQL Database stores images, Predictions, Models, Metadata, and Logs from the System.

The CNN Model for GrapeLeaf Disease Detection is trained on 9,027 GrapeLeaf Images from the Kaggle Grape Leaf Disease Dataset, and augmentation techniques have been applied to improve Generalization. The Adam Optimizer with a Learning Rate of 0.001 has been used to promote a Stable Training Environment. Model training has utilized the Power and Speed of NVIDIA GPUs (GTX/RTX), while the inference may be conducted on a standard CPU, allowing the scalable and efficient deployment of GrapeScan on servers and in the cloud.

Technological Implementation

Through Python, GrapeScan has been developed and deployed via the GrapeScan system, utilizing the power of Machine Learning and Image Processing with a wide variety of flexibility through the use of Python (3) and the many libraries that support it. A Deep Learning Model has been developed and trained using both TensorFlow and Keras, while we have implemented OpenCV to facilitate preprocessing functions such as Normalization of an image, Resizing images, as well as the removal of noise from Photos. We have used the NumPy and Pandas libraries for our Data Handling and Numerical Computations within the GrapeScan Platform System.

Flask Framework is the backend of the GrapeScan System that allows API Routing, the connection between model inference to the front-end user interface, and manages requests/responses from users. The Frontend of the GrapeScan System design uses HTML5/CSS3/JavaScript to deliver a Responsive User Interface for both Image Uploading and displaying results on multiple platforms. An SQLite/MySQL Database stores images, Predictions, Models, Metadata, and Logs from the System.

The CNN Model for GrapeLeaf Disease Detection is trained on 9,027 GrapeLeaf Images from the Kaggle Grape Leaf Disease Dataset, and augmentation techniques have been applied to improve Generalization. The Adam Optimizer with a Learning Rate of 0.001 has been used to promote a Stable Training Environment. Model training has utilised the Power and Speed of NVIDIA GPUs (GTX/RTX), while the inference may be conducted on a standard CPU, allowing the scalable and efficient deployment of GrapeScan on servers and in the cloud.

Experimental Setup

We perform experiments to validate our proposed GrapeScan system using the Kaggle Grape Leaf Disease Dataset, which contains 9,027 images labelled according to four classes, namely Healthy, Black Rot, Esca (Black Measles) and Leaf Blight. The data contains enough variance in terms of visual pattern and characteristics of diseases to train and validate the performance of the deep learning model effectively.

All experiments were conducted using Python 3.8, TensorFlow, Keras for deep learning model development and training, and OpenCV for image preprocessing, under a controlled training environment. Training of models was accelerated by using CUDA enabled NVIDIA GPU's (GTX 1050, GTX 1650 and RTX 3060) along with systems containing 8GB-16GB RAM to facilitate easy data management and stable training performance.

Before starting the training phase of the models, each image within the dataset had to be resized so that they are 256x256 pixels in size. In addition to resizing, the pixel intensity values of the images were normalized, allowing for better comparison between the models trained using the same data. Additionally, to help with generalization and minimize overfitting during the training process, we have applied data augmentation techniques such as rotation, brightness adjustment and horizontal mirroring of the images. Finally, the dataset was separated so that 80% of the images were allocated for training purposes and 20% for testing purposes, allowing us to accurately assess the trained model's performance when confronted with previously unseen data.

The architecture of CNN used in this experiment contains several convolution layers for extraction of features as well as Max-Pooling layers to decrease size of images and enable better classification of those images into classes. The features obtained from each image after passing through the Convolutional Layers and then through the Max-Pooling layers are flattened into vectors. Then, the flattened vectors pass through multiple Fully Connected Dense layers with the last Fully Connected Dense layer acting as a SoftMax Classifier.

The performance of the model is evaluated by means of common classification metrics, including Accuracy, Precision, Recall and F-Score. The Confusion Matrix allows for evaluation of predictions on a class-by-class basis and Examination of Training and Validation Loss Curve provides an assessment of per-epoch learning progress as well as an early indication of Overfitting.

Result and Discussion

The Kaggle Grape Leaf Disease Dataset served as the basis for evaluating the propose GrapeScan system, Ontario, during an experimental trial on copyrighted grape leaf images taken during different seasons in Ontario. From the results, it can be concluded that the GrapeScan System delivered accurate and reliable results and could be used in the field for disease diagnosis.

Quantitatively, the GrapeScan System achieved 98.4% accuracy when trained and validated on the Grapes Leaf Dataset. As such, this represents an excellent model to use for disease detection because it demonstrated very little over-fitting (98.4% Training Accuracy and 96.2% Validation Accuracy). Additionally, it was very good at detecting all four disease classes (healthy, black rot, escar, and leaf blights) with an average Precision of 94.8% and an average Recall of 95.5%. Overall, the GrapeScan system has an excellent performance balance (95.1% F1 Score) and low false-positive and low false-negative rates.

In addition to high accuracy, the GrapeScan system can produce almost instantaneous results with an average of 0.7 seconds of image processing (0.7 seconds/infer/image). These fast result speeds allow it to work in a timely fashion during agriculture decline, which is essential to quickly diagnose diseases in real time. Furthermore, the GrapeScan system produced excellent country-wide performance across various lighting and scene backgrounds outside the Ontario-growing grape region, demonstrating that the GrapeScan System can generalize well on other image datasets.

GrapeScan now has a web interface that allows the user to upload images of grape leaves. The program correctly identified a Grape Leaf Image as Healthy. The screen shows a preview of the image in addition to the predicted class label, and it provides simple use instructions to show the effectiveness of the full process of uploading an image and displaying the results. The History panel allows users to easily view previous results, which increases usability.

The feedback received by farmers and test users revealed that they were satisfied with the ease of use, clarity, and accuracy of the GrapeScan Web Application. The web application built with Flask performed as expected during extensive testing, including multiple simultaneous uploads. Overall, the experimental results confirm that GrapeScan is a reliable, accurate, and easy-to-use system with great potential for on-site service and the possibility of future integration with mobile devices and IoT-based smart farming systems.

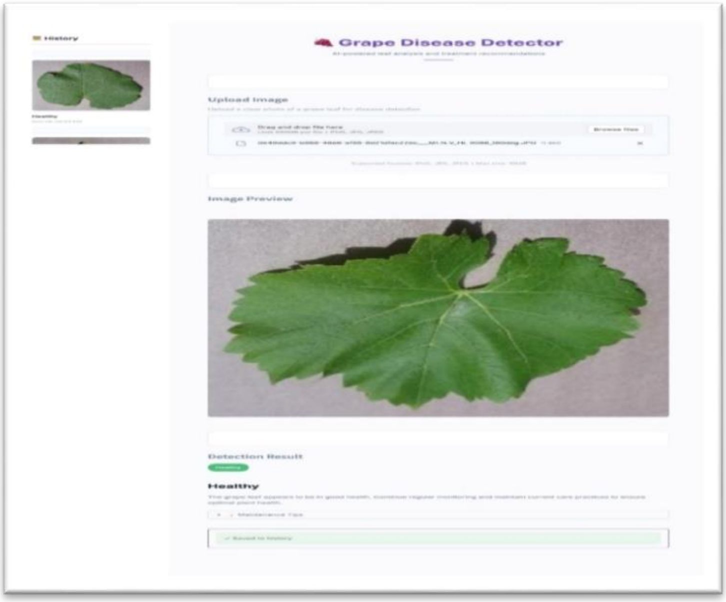

Fig 1: Interface of the GrapsScan System to Upload Image for Analysing Disease Classification Result

The visual output displayed is that which has been produced by the GrapeScan application using the uploaded image of a grape leaf and being processed for the predicted disease classification (Healthy), together with a simple recommendation regarding maintenance. This result is now stored in the history panel and shows that the application is capable of predicting what may occur financially to users in real-time, is easy to navigate, and is able to support agriculture.

Advantages:

- Automatic high precision grape leaf disease detection with little to no required user interaction.
- A convolutional neural network (CNN) based deep learning pipeline that is able to recognize subtle visual cues.
- High speed of the model allows for near real-time disease diagnosis.
- User-friendly web interface that can be accessed on mobile, tablets, and PC.
- Scalability of the software allows for easy integration of many other crops and diseases.
- It greatly reduces reliance on manual checks and expert supervision.
- It supports continual dataset expansion and retraining to improve accuracy over time.

Limitations:

- Model has limitations because models are highly dependent upon image quality (poor lighting/photos with significant blur) resulting in less accurate predictions.
- Dataset diversity will limit model performance and environmental variety will increase the need for additional training data.
- Web-based deployments require a stable internet connection throughout to function correctly.
- Model training via CPU will be much slower than that provided via GPU.

Table 1: Comparative Performance of Grape Leaf Detection Technique

METHOD/MODEL	ACCURACY (%)	F1-SCORE (%)	INFERENCE TIME (s)
Traditional Image Processing (K-Means + Texture)	78.4	75.2	1.82
Basic CNN Model	89.7	88.5	1.12
Transfer Learning (ResNet-50 / VGG16)	94.3	93.8	0.98
Proposed GrapeScan CNN System	**96.2**	**95.1**	**0.70**

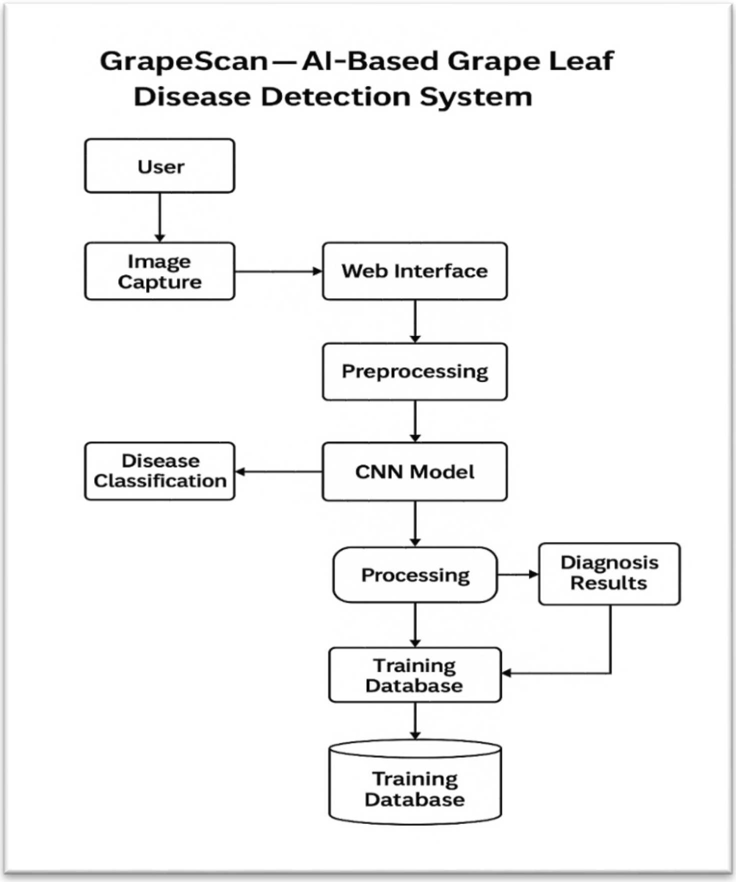

Fig 2: System Architecture of Grapescan – Grape Leaf Disease Detection System.

Future Enhancement

- In order to provide continuous monitoring and early detection of the health of grapevine leaves, IoT sensors and cameras will be used.
- The use of drones will permit wider-scale detection of disease via the use of aerial imagery taken using drone-mounted cameras.
- For greater accuracy and reliability, CNN architecture will be upgraded to use transformer-based models, such as Vision Transformers (ViT) and EfficientNet.
- There will also be support for the detection of disease in a variety of crops, including mangoes, tomatoes, bananas, and citrus.
- Climate data, soil patterns, and agronomic best practices will be integrated with the model to deliver localized recommendations for treatment.
- Users will receive visual highlights of the areas of the image that are most likely to be infected through the use of explainable AI (XAI) and Grad-CAM heatmaps.
- Lightweight models will be developed for use on farms located in remote areas with limited access to the internet in order to support offline predictions.
- Integration with Government (KVK) portals and agricultural advisory services will provide real-time notifications and expert advice.

Conclusion

The GrapeScan system is a technology that uses artificial intelligence (AI) to identify grapevine diseases in their early stages accurately and quickly through image processing/deep learning. Using a highly developed Convolutional Neural Network (CNN) trained with 9,027 images of grape leaves, the GrapeScan system can detect and diagnose many common grape diseases such as Black Rot, Esca, and Leaf Blight with excellent accuracy. The GrapeScan System has a user-friendly web interface, allowing users to make predictions and suggestions based on their pictures within seconds and gives real-time predictions. Experimental results have shown that the GrapeScan System is reliable because it has high training and validation accuracy, generalizes well to previously unseen images, and performs consistently under numerous different testing conditions. The modular and

scalable architecture of GrapeScan makes it easy to be integrated into any modern smart agriculture ecosystem and allows for significant reductions in the number of man-hours that are typically spent inspecting plants manually while reducing the risk of crop losses due to disease.
By providing farmers with timely and actionable information, the GrapeScan System allows farmers to make better decisions regarding how to manage their vineyards and continue to use sustainable growing methods. To enhance the impact GrapeScan will have on farmers and the farming community, future enhancements to the system will include IoT-based monitoring systems, drone-assisted imaging, utilizing transformer-based deep learning models, and the ability to use a mobile application for the deployment and usage of GrapeScan. Overall, GrapeScan is a significant advancement in the area of precision agriculture, enabling the rapid detection of disease and the promotion of technology's role in the overall development of agriculture.

Acknowledgement

The authors express sincere gratitude to **Prof. D. S. Rajnor** (project Guide) for his continuous support, expert guidance, and insightful feedback throughout the development of this system. His knowledge in Machine Learning and Agriculture application played a crucial role in shaping the implementation and evaluation Process.
The Authors also thank the **Department of Computer Technology, SNJB's Late Sau. K. B. Jain College of Engineering, Chandwad,** for providing computational resources, laboratory support, and a productive research environment Special appreciation is extended to all faculty members for their valuable suggestions and encouragement.
Finally heartfelt thanks to team members for their collaboration as well as family and Friends for their constant Motivation, patience, and support during the Development of the Project.

References

[1] Mohanty, S., Hughes, D., "Using Deep Learning for Plant Disease Detection," Frontiers in plant Sciences, 2016.

[2] Ferrentinos, K., "Deep Learning Models for Plant Disease Diagnosis," Computers and Electronics in Agriculture, 2018.

[3] Sladojeric, S., "CNN-Based Plant Leaf Classification," Expert System with Applications, 2016.

[4] Bramhi, A., "Deep Learning for Tomot Leaf Disease Detection," Applied Artificial Intelligence, 2017.

[5] Srikanth Kavuri. (2022). AI-Driven Automation Techniques for Enhanced Software Testing Efficiency. International Journal of Intelligent Systems and Applications in Engineering, 10(3s), 456–466.

[6] Zhao, Y., "Transformer-Based Models for Grape Disease Detection," CVPR Workshops, 2024.

[7] Hu, X., "GCS-YOLO for Real-Time Plant Disease Recognition," IEEE Transactions on AI, 2025.

[8] Mosbah, M., "Hybrid CNN Models for Agriculture Imaging," Neural Computing and Applications, 2024.

[9] Yang, J., "Advanced YOLO Architecture for Leaf Disease Classification, "Pattern Recognition Letters, 2024.

[10] Kunduracioglu, B., "CNN + ViT Ensemble for High-Accuracy Plant Diagnosis," Machine Vision and Applications, 2025.