JSRT JOURNAL

PAGES: 128-139
8/24/23
JOURNAL OF SCIENTIFIC RESEARCH AND TECHNOLOGY(JSRT) VOLUME-1 ISSUE-5 AUGUST
Registered under MSME Government of India ISSN: 2583-8660

Grape Leaf Disease Detection Using Deep Learning

Deeksha Nargotra[1], Prof. Vinod Sharma[2]

[1]*Student, Department of Computer Science & IT, University of Jammu, Jammu, India*
deekshanargotra3@gmail.com

[2]*Professor, Department of Computer Science & IT, University of Jammu, Jammu. India*

ABSTRACT

An AI function called as deep learning (sometimes deep structured learning or hierarchical learning) attempts to mimic the brain's pattern recognition and data analysis capabilities. Deep learning is a branch of AI's machine learning that makes use of unstructured or unlabeled data to train its networks. Using deep learning to identify diseases in grape leaves has several benefits. First, it facilitates accurate and timely illness diagnosis, which in turn paves the way for efficient and timely disease management. Second, deep learning models are capable of processing massive volumes of complicated picture data, so enabling the capture of subtle patterns and characteristics that may have eluded more conventional approaches. Furthermore, these models may learn from experience and become better with additional data, enhancing their accuracy and precision of predictions over time. Improvements in vineyard management, decreased crop losses, and more support for sustainable agriculture might all result from applying deep learning to identify diseases on grape leaves.

Keywords: Deep Learning, Disease Detection, Grape Disease

I. INTRODUCTION

Grapes are a widely cultivated commercial crop worldwide, primarily for their use in making wine, alcoholic drinks, and dried grapes (raisins). The extent of a disease affecting the grape, and hence the likelihood of economic loss, increases in proportion to the area under grape production. The leaves are the first to show signs of damage from grape diseases. That's why you may use the leaves to spot ailments in their earliest stages and get a proper diagnosis. One of the most common grape diseases is black rot. The grape leaves develop a black mark due to black rot, a fungal disease. When compared to the size of the leaves, this area seems rather little. This illness often shows up in the wet spring and early summer months. The black mark appears on a large number of leaves. Disease diagnosis now relies mostly on manual methods. The farmers will utilize their years of knowledge to make an educated guess as to what ails the crop. It's worth noting, however, that this method not only involves a great deal of human labor but is also open to the influence of subjective variables. The agricultural sector relies on prompt and accurate identification of black rot on grape leaves to protect grape yields and farmer incomes.

1.1 Types of grape diseases

Black Measles:- Brown steaking lesions on any area of the leaf are a telltale sign of this disease, which affects the grape plant. The disease may cause the leaves to dry up and fall off the plant before their time, killing the plant. Grapes affected by this disease have dark spots on their leaves and veins. The symptoms are most common in July and August, however they might appear at any point throughout the growth season. This phenomenon is also known as apoplexy.

Black Rot:- Black rot is a fungal disease that reduces harvest yields and may also degrade wine quality. It appears as irregularly spaced, brown or tan circular dots on the leaf. One of the most devastating grape diseases is black rot. The grape fungus may infect the cluster stem, leaves, shoot, and berries. The effects of a disease on grapes may be severe if not controlled early in the growing season, leading to lost harvests.

Healthy:- A healthy grape leaf has a green color, is spotless, and has a glossy, smooth surface. Medium-sized (around four to five inches) green leaves that let in plenty of sunlight. Detecting healthy grape leaves in photographs and differentiating them from damaged or sick leaves is a computer vision job known as "grape leaf detection." The goal is to create an automated system that can determine whether or not grape leaves are healthy just by looking at them.

Fig.1 Black Measles

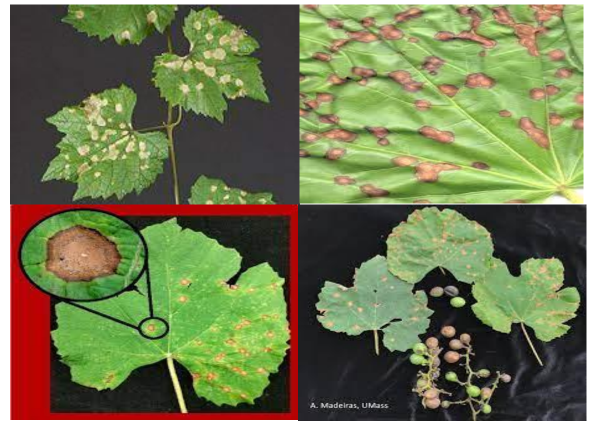


Fig.2 Black Rot

Fig.3 healthy

1.2 Hypothesis

Grape disease classification is very amenable to deep learning approaches.

1.3 Expected contribution to the study

Accurate disease classification in grapes using Deep Learning is a goal of the proposed research.

1.4 Area of Study

To conduct the proposed research, we will use a benchmark dataset of grape leaf diseases and the Python programming language to apply deep learning.

1.5 Justification

To prevent losing market value, reliable identification is a crucial duty. Because of their nutritional significance, grape leaves are of significant interest to humans. Therefore, the wholesale and retail markets, as well as the processing industries, have a vested interest in grape leaf detection, identification, and classification studies. Also, identifying a grape's location of origin will help us tell it apart from varieties from other regions. Therefore, a more effective model is required to properly categorize grape leaf detection.

1.6 Objectives

- Research grape leaf diseases in depth.
- In order to use deep learning for disease classification in grape leaves.
- Use metrics like accuracy, recall, precision, and f1 score to assess how well your deep learning implementation is doing.

II. REVIEW OF LITERATURE

The suggested study's primary objective is to use Deep Learning for plant detection classification. We have researched the relevant literature on the suggested subject and provide a summary of our findings below.

- **Padol et al. (2016) [1]** recommended research into the devastating effects of black rot, black measles, leaf blight, and mites on grape output. The current state of knowledge does not provide a real-time detecting approach for grape leaf diseases, which limits our ability to ensure the robust development of grape plants. In this piece, we present a deep convolutional neural network-based real-time detector for grape leaf diseases. Page 6 Before building the Grape Leaf Disease Dataset (GLDD), this article uses digital image processing to enlarge the photos of grape leaf diseases. Using GLDD and the Faster R-CNN detection technique, we offer a deep learning-based Faster DR-IACNN model with improved feature extraction for identifying grape leaf diseases. The detection speed reaches 15.01 FPS, and the model's accuracy is 81.1% mAP on GLDD, according to the experiments. Based on these findings, we may deduce that the deep learning-powered, real-time detector Faster DR-IACNN is a viable option for diagnosing illnesses in grape leaves and may even serve as a model for detecting diseases in other plant species.
- **Amara et al. (2017) [2]** proposed Plant diseases are significant because they drastically lower agricultural output both in quality and quantity. This emphasizes the need of early diagnosis and screening for such disorders. The suggested method uses deep learning to automatically categorize banana-leaf-related disorders. In specifically, this employs a convolutional neural network using the LeNet architecture for the purpose of picture classification. The first findings show that the suggested method works well even when presented with difficult situations including inconsistent lighting, a busy backdrop, and pictures of varying quality, size, position, and orientation from the actual world. In agriculture and forestry, plant diseases result in significant output and economic losses.
- **Durmuş et al (2017) [3]** project proposed to identify plant diseases in tomato fields and greenhouses. To this end, we used deep learning to identify a variety of tomato leaf diseases. The project attempted to have the robot execute the deep learning algorithm in real time. The robot will be able to identify plant illnesses while it freely roams the field or greenhouse under human control or on its own. Similarly, sensors installed in artificial greenhouses may utilize close-up photos of plants to identify illnesses. Symptoms of the diseases under investigation here include outwardly visible alterations to the leaves of the tomato plant. RGB cameras can detect these color shifts in the leaves. Previous research has relied on the use of conventional feature extraction techniques applied to photos of plant leaves in order to identify illnesses. Disease detection was the focus of this research, which employed deep learning techniques. To train the model, we utilized photos of tomato leaves from the Plant Village dataset. Choosing the right deep learning architecture proved to be the biggest challenge.
- **Zhang et al. (2017) [4]** recommended research into the four most frequent forms of apple leaf disease: rust, brown spot, and alternaria leaf spot. Controlling the spread of infection and ensuring the apple industry's healthy growth depend on prompt and precise detection of illnesses affecting apple leaves. Current studies rely on intricate image preprocessing, however this does not ensure accurate disease detection in apple leaves. Using deep convolutional neural networks, the authors of this research offer a method for accurately recognizing apple leaf illnesses. We develop a deep convolutional neural network model to detect four

prevalent apple leaf illnesses using a dataset of 13,689 photos of damaged apple leaves. The experimental results demonstrate that the proposed disease identification approach based on the convolutional neural network achieves an overall accuracy of 97.62% under the hold-out test set, while reducing the model parameters by 51,206,928 compared to those in the standard Alex Net model and achieving an improvement of 10.83% in accuracy when using generated pathological images.

III. METHODOLOGY AND IMPLEMENTATION

3.1 Research Methodology

The primary goal of this study is to develop a classification system for grape leaf recognition based on a deep learning algorithm. Filtering and scaling the picture to pixels are two examples of preprocessing procedures. After the data has been cleaned and prepared, it will be split into training and testing sets. The plan is to use a deep learning technique to train the model, and then to put it to the test on a separate dataset. A further step involves evaluating the algorithm's effectiveness in practice. Research process flowchart for symptom-based grape leaf disease diagnosis. Black rot, black measles, healthy, and many more are among the numerous common illnesses. Discoloration, stains, lesions, deformations, and powdery growth on the leaves are all symptoms unique to each illness.

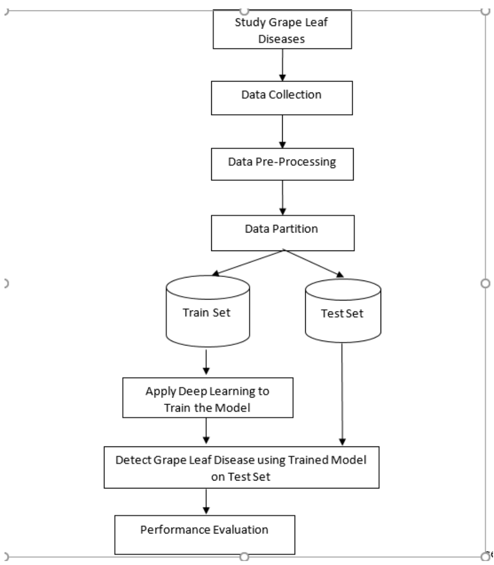


Fig.4 Workflow of the proposed research methodology

3.2 Dataset and its collection:

The Grape Leaf Dystrophy Dataset from Kaggle. Gathering photos and annotations of diseased grape leaves provides a supplemental dataset for disease identification in grapes. There are pictures of healthy leaves, leaves with black rot and black measles, and healthy leaves.

NAME OF DISEASE	BLACK MEASLES	BLACK ROT	HEALTHY
TRAINING DATA	1536	1510	1354
TESTING DATA	384	378	338

PAGES: 128-139
8/24/23
JOURNAL OF SCIENTIFIC RESEARCH AND TECHNOLOGY(JSRT) VOLUME-1 ISSUE-5 AUGUST
Registered under MSME Government of India ISSN: 2583-8660

The total size of dataset includes 5500 images. Attention to the related work is given to all the diseases.

Black Measles

Test Images					
Train Images					

Fig.5 Test and Train Images of Black Measles

Black Rot

Test Images						
Train Images						

Fig.6 Test and Train Images of Black Rot

Healthy

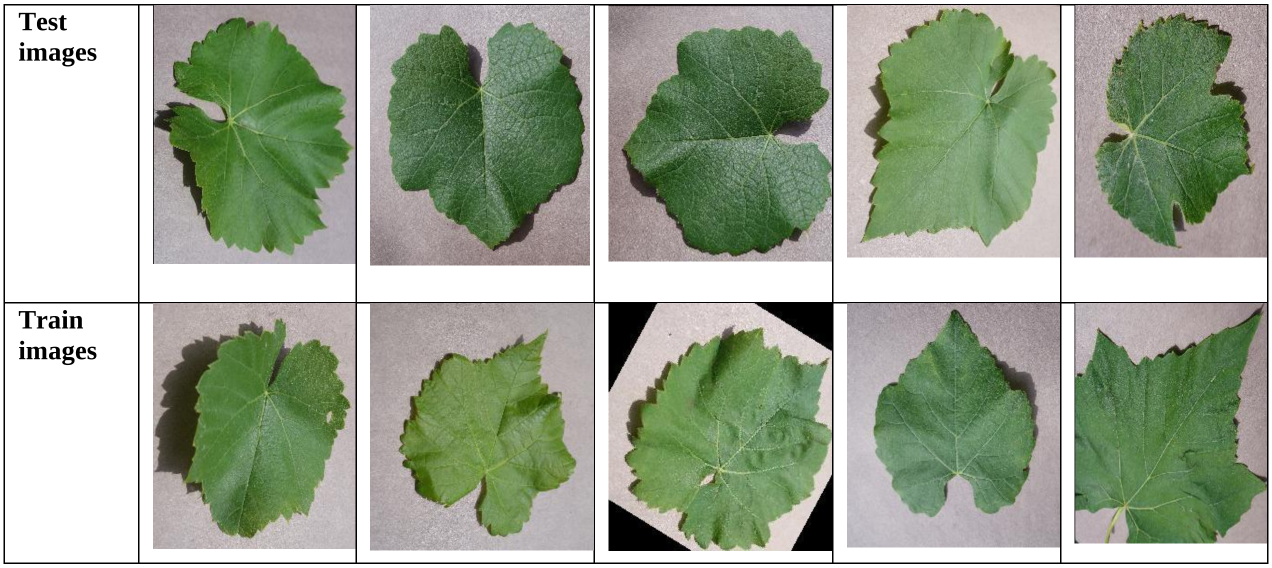


Fig.7 Test and train images of healthy leaf

3.3 Implementation

To get our CNN model up and running, we must first import the necessary libraries. We are use the following libraries:

Keras:

Python's Keras provides a high-level interface for neural networks. TensorFlow, Theano, and CNTK are just few of the major deep learning frameworks that it is based on. Keras provides a simple and straightforward interface for building and training deep learning models like CNNs. It facilitates rapid model prototyping and experimentation by providing a modular and adaptable framework for constructing different kinds of neural networks. Keras is compatible with graphics processing unit (GPU) and central processing unit (CPU) calculations and may be used in tandem with other Python packages.

OS:

Python's OS module allows for communication with the OS. You can manipulate files, folders, and processes in a wide variety of ways. Common uses of the OS module include managing paths and files, reading and writing environment variables, running system commands, and accessing environment variables. Loading datasets from disk, maintaining file directories, and planning the training process are all examples of where the OS module might be helpful in the context of deep learning and CNNs. Keras, scikit-learn, OpenCV, and the OS module are libraries that offer crucial functionality for many facets of machine learning and computer vision projects. Gaining familiarity with their features and putting them to good use may greatly improve your efficiency while developing and deploying CNN models.

Tensorflow:

The Google Brain team created TensorFlow, an open-source machine learning framework. Create, train, and deploy machine learning models, especially deep neural networks, with its help. TensorFlow is well-known in the AI community for its adaptability, scalability, and support of several use cases, such as computer vision, natural language processing, reinforcement learning, and more. Disease identification in grape leaves using the TensorFlow library entails training a deep learning model to distinguish between healthy and unhealthy grape leaves based on visual inputs. When it comes to creating and training neural networks, TensorFlow is a popular choice as a deep learning framework.

Importing All the libraries

```
import tensorflow as tf
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense, Dropout
```

Define the path of dataset

```
# Set the path to your dataset
train_data_dir = r'C:\Users\Hp\Desktop\Dataset1\Train'
validation_data_dir = r'C:\Users\Hp\Desktop\Dataset1\Test'
test_data_dir = r'C:\Users\Hp\Desktop\Dataset1\validation'
```

Data preprocessing

```
# Data preprocessing
batch_size = 32
image_size = (224, 224)  # Adjust the size according to your dataset

train_datagen = ImageDataGenerator(
    rescale=1.0 / 255,
    rotation_range=20,
    width_shift_range=0.2,
    height_shift_range=0.2,
    horizontal_flip=True,
    vertical_flip=True,
    fill_mode='nearest'
)

validation_datagen = ImageDataGenerator(rescale=1.0 / 255)
test_datagen = ImageDataGenerator(rescale=1.0 / 255)

train_generator = train_datagen.flow_from_directory(
    train_data_dir,
    target_size=image_size,
    batch_size=batch_size,
    class_mode='categorical'
)

validation_generator = validation_datagen.flow_from_directory(
    validation_data_dir,
    target_size=image_size,
    batch_size=batch_size,
    class_mode='categorical'
)

test_generator = test_datagen.flow_from_directory(
    test_data_dir,
    target_size=image_size,
    batch_size=batch_size,
    class_mode='categorical'
)
```

Build the CNN model

```
# Build the model
model = Sequential([
    Conv2D(32, (3, 3), activation='relu', input_shape=(224, 224, 3)),
    MaxPooling2D(2, 2),
    Conv2D(64, (3, 3), activation='relu'),
    MaxPooling2D(2, 2),
    Conv2D(128, (3, 3), activation='relu'),
    MaxPooling2D(2, 2),
    Flatten(),
    Dense(512, activation='relu'),
    Dropout(0.5),
    Dense(3, activation='softmax')  # Adjust the output units to match the number of classes
])
```

JSRT JOURNAL
PAGES: 128-139
8/24/23
JOURNAL OF SCIENTIFIC RESEARCH AND TECHNOLOGY(JSRT) VOLUME-1 ISSUE-5 AUGUST
Registered under MSME Government of India ISSN: 2583-8660

compile the Model

```
# Compile the model
model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])
```

Train the model

```
# Train the model
epochs = 5  # Adjust the number of epochs as needed
history = model.fit(
    train_generator,
    epochs=epochs,
    validation_data=validation_generator
)
```

Evaluate the model

```
# Evaluate the model on the test set
test_loss, test_acc = model.evaluate(test_generator)
print("Test accuracy:", test_acc)
```

Model fitting

```
Found 3521 images belonging to 3 classes.
Found 1484 images belonging to 3 classes.
Found 879 images belonging to 3 classes.
Epoch 1/5
111/111 [==============================] - 2504s 22s/step - loss: 0.8009 - accuracy: 0.6924 - val_loss: 13.3308 - val_accuracy:
0.4582
Epoch 2/5
111/111 [==============================] - 2125s 19s/step - loss: 0.3492 - accuracy: 0.8679 - val_loss: 11.0348 - val_accuracy:
0.5216
Epoch 3/5
111/111 [==============================] - 1817s 16s/step - loss: 0.2264 - accuracy: 0.9202 - val_loss: 9.8346 - val_accuracy:
0.6213
Epoch 4/5
111/111 [==============================] - 1919s 17s/step - loss: 0.1987 - accuracy: 0.9310 - val_loss: 11.2857 - val_accuracy:
0.6995
Epoch 5/5
111/111 [==============================] - 1957s 18s/step - loss: 0.2016 - accuracy: 0.9242 - val_loss: 10.5261 - val_accuracy:
0.6732
28/28 [==============================] - 56s 2s/step - loss: 0.2748 - accuracy: 0.9044
Test accuracy: 0.9044368863105774
```

```
In [2]: print("Test accuracy:", test_acc*100)

Test accuracy: 90.44368863105774
```

IV. TESTING AND RESULT

For testing purposes, we can use various measures for testing and some of them are defined below:

Accuracy: The accuracy (ACC) is the proportion of the total number of predictions that were correct. It is given by the relation.

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

Recall: Number of accurately labeled positive instances as a percentage of total positive examples is the definition of recall. When the recall is high, the class is identified accurately. Names for this statistic include True Positive Rate and Detection Rate. That's what the connection guarantees.

$$\text{Recall} = \frac{TP}{TP + FN}$$

Precision: Precision is the ratio of correct diagnoses to total diagnoses (i.e., correct diagnoses + false positives). That's what the connection guarantees. True positives (TP) and false positives (FP) are two different measures of accuracy.

$$\text{Precision} = \frac{TP}{TP + FP}$$

F-score: The F-score is useful since it measures both recall and accuracy simultaneously. Harmonic Mean replaces Arithmetic Mean by being harsher on outlying numbers. Classifiers with high F1 scores tend to have high accuracy and recall. If you need a quick method to evaluate two classifiers, the F1 score combines accuracy and recall into a single statistic.

$$F\text{ - }measure = \frac{2 * \text{Recall} * \text{Precision}}{\text{Recall} + \text{Precision}}$$

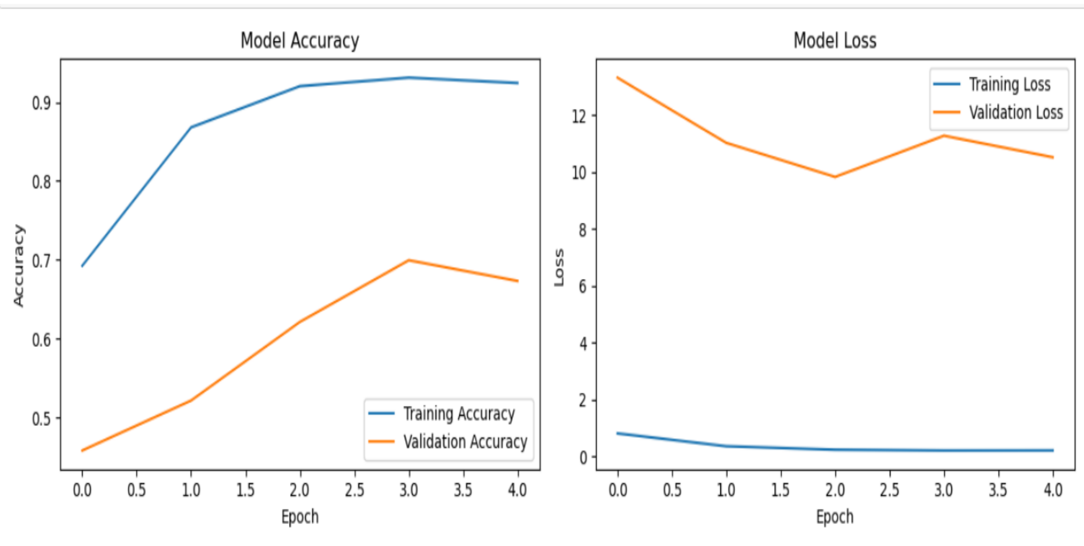
JSRT JOURNAL

PAGES: 128-139
8/24/23
JOURNAL OF SCIENTIFIC RESEARCH AND TECHNOLOGY(JSRT) VOLUME-1 ISSUE-5 AUGUST
Registered under MSME Government of India ISSN: 2583-8660

Model summary:

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 222, 222, 32)      896

 max_pooling2d (MaxPooling2  (None, 111, 111, 32)      0
 D)

 conv2d_1 (Conv2D)           (None, 109, 109, 64)      18496

 max_pooling2d_1 (MaxPoolin  (None, 54, 54, 64)        0
 g2D)

 conv2d_2 (Conv2D)           (None, 52, 52, 128)       73856

 max_pooling2d_2 (MaxPoolin  (None, 26, 26, 128)       0
 g2D)

 flatten (Flatten)           (None, 86528)             0

 dense (Dense)               (None, 512)               44302848

 dropout (Dropout)           (None, 512)               0

 dense_1 (Dense)             (None, 3)                 1539

=================================================================
Total params: 44397635 (169.36 MB)
Trainable params: 44397635 (169.36 MB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

Results and accuracy



V. CONCLUSION AND FUTURE SCOPE

In conclusion, the use of deep learning to identify diseases in grape leaves has tremendous potential to completely transform the viticulture business. Grape leaf diseases are notoriously difficult to detect and categorize, but deep learning models, in particular convolutional neural networks (CNNs), have shown exceptional accuracy and efficiency. These models can tell the difference between healthy and sick leaves and between various kinds of illnesses by making use of vast datasets and sophisticated architectures, therefore assisting producers in making early and targeted treatments.

Using deep learning to identify diseases in grape leaves has several benefits. First, it facilitates accurate and timely illness diagnosis, which in turn paves the way for efficient and timely disease management. Second, deep learning models are capable of processing massive volumes of complicated picture data, so enabling the capture of subtle patterns and characteristics that may have eluded more conventional approaches. Furthermore, these models may learn from experience and become better with additional data, enhancing their accuracy and precision of predictions over time.

There are, however, several obstacles to overcome in order to successfully utilize deep learning for disease detection in grape leaves. To guarantee the models' robustness and generalizability, sufficient and different training datasets are required.

Prospects for even more advanced illness detection and prediction models are bright, thanks to continuous research and development in the area of deep learning. Improvements in vineyard management, decreased crop losses, and more support for sustainable agriculture might all result from applying deep learning to identify diseases on grape leaves.

Future work

The use of deep learning to the problem of disease detection in grape leaves promises significant future improvements in these areas. Several areas of future research may significantly improve the capabilities and uses of grape leaf disease detection as the science of deep learning continues to evolve:

Large-Scale Datasets: In order to train deep learning models, it is necessary to create and curate bigger, more varied datasets. Collecting information from a wide range of vineyards, geographies, and growth circumstances may help models become more generalizable and adaptive.

Semi-Supervised and Active Learning: By exploring semi-supervised and active learning algorithms, which can make effective use of unlabeled data, we may decrease our reliance on completely labeled datasets and speed up the annotation process.

Model Interpretability: Fostering trust and allowing cultivators to better comprehend and act upon model suggestions will need the development of tools to analyze deep learning models and make their decision-making process more transparent.

Robustness to Environmental Factors: Improving the models' effectiveness in real-world circumstances requires looking into how to make them more resilient to changes in lighting, weather, and other environmental variables.

The potential for improved disease detection in grape leaves bodes well for the future of the agricultural industry as a whole.

REFERENCE

[1] Padol, P. B., and Yadav, A. A. (2016). “SVM classifier based grape leaf disease detection,” in Proceedings of the Conference on Advances in Signal Processing, Lisbon, 175–179.

[2] Amara, J., Bouaziz, B., Algergawy, A., et al.: A deep learning-based approach for banana leaf diseases classification. In: BTW (Workshops), pp. 79–88 (2017)

[3] Durmuş, H., Güneş, E.O., Kırcı, M.: Disease detection on the leaves of the tomato plants by using deep learning. In: 2017 6th International Conference on Agro-Geoinformatics, pp. 1–5. IEEE (2017)

[4] Liu, B., Zhang, Y., He, D., and Li, Y. (2017). Identification of apple leaf diseases based on deep convolutional neural networks Symmetry 10:11. doi: 10.3390/sym10010011

[5] Ferentinos, K.P.: Deep learning models for plant disease detection and diagnosis. Comput. Electr. Agri. 145, 311– 318 (2018)

[6] Keke Zhang & Lei Zhang & Qiufeng Wu, 2019. "Identification of Cherry Leaf Disease Infected by Podosphaera Pannosa via Convolutional Neural Network

[7] G., Pandian, A.: Identification of plant leaf diseases using a nine-layer deep convolutional neural network. Comput. Electr. Eng. 76, 323–338 (2019)

[8] Bresilla, K., Perulli, G. D., Boini, A., Morandi, B., and Corelli Grappadelli, L. (2019). Single-shot convolution neural networks for real-time fruit detection within the tree. Front. Plant Sci. 10:611. doi: 10.3389/fpls.2019.00611

[9] Islam, M., Dinh, A., Wahid, K., and Bhowmik, P. (2017). “Detection of potato diseases using image segmentation and multiclass support vector machine,” in Proceedings of the IEEE 30th Canadian Conference on Electrical and Computer Engineering, Windsor, 1–4.

[10] Lu, Y., Yi, S., Zeng, N., Liu, Y., and Zhang, Y. (2017). Identification of rice diseases using deep convolutional neural networks. Neurocomputing 267, 378–384. doi: 10.1016/j.neucom.2017.06.023.

[11] S.Gulavnai and R.Patil, “Deep Learning for Image Based Mango Leaf Disease Detection Detection”,International Journal of Recent Technology and Engineering(IJRTE), Volume.8,2019.

[12] U. Rao, R. Swathi, V. Sanjanaa, L.Arpithaa and K. Chandrasekher, “Deep Learning precision Farming: Grapes and Mango Leaf Disease Detection by Transfer learning”,2021.

[13] R.Khan, K.Khan,W .Albattah and A. Qamar, “Image-based Detection of Plant Disease:from Classical Machine Learning to Deep Learning journey”2021.

[14] G. Geetha l, S.Samundeswari, G.Saranya, K. Meenakshi and M. Nithya, “Plant leaf Disease Classification and Detection System Using Machine Learning”2020.

[15] Juncheng Ma, Keming du, Feixiang Zheng, Lingxian Zhang, Zhihong Gong, Zhongfu Sun, “A recognition method for cucumber diseases using leaf symptom images based on deep learning”2018.

[16] Dian Rong , Lijuan Xie , Yibin Ying , “Computer vision detection of foreign objects in walnuts using deep learning”, computers and Electronics in Agriculture ,2019

[17] https://ipm.ucanr.edu/PMG/IMAGES/U/D-GR-UNKN-FO.003banner.png

[18]https://upload.wikimedia.org/wikipedia/commons/thumb/c/cb/Guignardia_bidwellii_08.jpg/220px-Guignardia_bidwellii_08.jpg