

Pepercorn Leaf Disease Detection and Classification model Using Deep Learning Approach

Barcot Bahiru

barcotbahiru5@gmail.com

Debre Markos University

Article

Keywords: Classification, Deep Learning, Pepercorn, Ethiopia, Ultralytics, Roboflow

Posted Date: November 10th, 2025

DOI: <https://doi.org/10.21203/rs.3.rs-7759013/v1>

License: © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Additional Declarations: No competing interests reported.

Abstract

Particularly in the most lowland areas of Ethiopia, Pepercorn is an essential crop that makes a substantial contribution to the country's agricultural economy. However, several diseases that affect crop output and quality provide a barrier to Pepercorn production. Conventional disease detection techniques depend on specialist knowledge and manual inspections, which are frequently time-consuming and ineffective, limiting prompt response. Digital image processing, computer vision, and deep learning technologies have a lot of potential, but their use in Ethiopia's agriculture industry is still unexplored. The need for more sophisticated methods is highlighted by the fact that previous studies primarily used manual feature extraction techniques for disease detection. After a careful analysis of relevant literature, four deep learning architectures were selected: VGG16, VGG19, DenseNet121 and YOLOv11n. Several train-test data splits, such as 70%/30%, 80%/20% and 90%/10% were explored to assess model performance; the VGG19 with 90%/10% split produced the best accuracy 98.05% in case of VGGNet. And the DenseNet121 with 80%/20% achieves better accuracy 98.75% than VGG19. But YOLOv11n is the better model among the entire models researchers used. It achieves a mean average precision (MAP) of 99.03%. When we see the results in terms of performance (speed) the YOLOv11n model performs its preprocessing and post processing tasks in 1Hr, 096 Seconds, while DenseNet121 takes a speed of 2Hr, 35 Seconds. According to the study's finding, out of all the algorithms studied, YOLOv11 is the best model for Pepercorn leaf disease detection and classification.

I. Introduction

The viability of the economies of developing nations depends heavily on agricultural products. Agriculture has a major role in the economies of the majority of emerging nations, including Ethiopia. The Pepercorn crop is one of the most significant agricultural crops in terms of human food security on a global scale [1]. Numerous pests and diseases threaten crops, particularly in temperate, tropical, and subtropical climates. Complex relationships exist between the virus, its vector, and the host plant in plant diseases. Climate change's impact on the atmosphere and how it changes ecosystems are sometimes linked to the background of this issue [2]. A deep learning method for learning various characteristics from leaf images with classifiers for disease detection and classification was examined in this study [3]. The mother of all cultures is agriculture [4]. Pepercorn crop is highly cultivated and cropped around the lowland areas of many Ethiopian regions especially in AMHARA around WEST GOJJAM in JABI TEHNNAN, BURRIE SHIKUDAD, and DEMBECHA woredas and partially in some part of WOLLO, in Oromiya, Harare and some part of southern Ethiopia. It is a very sensitive crop up to its growth lifetime which means it is infected by different disease in different conditions for example it may infected by Herbicide chemicals, bacterial, fungi, leaf Shrinkage, and even infected by leaf burned due to frosty weather and rainy season. Generally the overall cultivation process has two phases that is First phase and Second phase. The first phase is the sowing and breeding on the temporary farm land. In the second phase the trees are dig upping and planting on the real farm land and the rest of cultivation process is applied here up to collecting the product. First it should be sow or breed on a separate area and keep it from floods, weeds and different insects. Then after once the tree gets almost 7 up to 10 number of leaf

it dig upped and planted on the other farm land and the other cultivating process will continue almost for three months. In Ethiopia, vegetable crops are produced in different agro-ecological zones through commercial as well as smallholder farmers both as a source of income and food[5]. When we compare with other crops It is a very life changing crop due to its better market but it needs much more initial budget to crop it successfully since it requires high amount of fertilizer (DAP and UREA/ NPS) to grow up and to Increase its product, it requires digging almost for four times within three months, it requires a special ditch to collect and remove internal floods from the farm and to protect external floods from getting in to the farm and it requires removing weeds almost two times[6]. Pepper is almost an indispensable crop in life and is closely related to human obesity rate cardiovascular disease[7].

At the end, the organizations of the rest part of this paper are as follows: Literature Reviews are seen in the part of "Related works." Next to this the Proposed Model Architecture is going to be discussed with the proposed method and algorithms used. Following this Experimental setup and results are presented. Finally Conclusion and recommendation summarizes the paper and highlights the future works to be done.

II. Related works

The authors Yohannes, beniam, abdella, ayodeji and aleka melese (2019) they focused on pepper leaves and fruit disease classification using a concatenation of convolutional neural network (CNN) models to identify healthy pepper leaves, common rust in pepper leaves, gray-leaf spot in pepper leaves, fruit disease, and blight disease in pepper leaves. They achieve 95.82% accuracy but their limitation was since their scope was on leaf and fruit, they missed other diseases on the leaf and they didn't mentioned how they solve if more than one disease is occurred.

The authors Haile and kedir (2021) evaluated the recognition of bell pepper leaf disease identification using VGG-19 on healthy and diseased pepper leaves and achieved an accuracy of 97.84%. The exclusion of validation and testing accuracy is the authors' limitation. Based on the investigation's results, the authors suggested more research to improve the identification of different diseases by finding characteristics of fruit diseases in addition to their leaves. The study's attempt to isolate only binary bell pepper leaves from both healthy and bacteria-spot disease was one of its limitations.

The authors Bhagat, M., Kumar, D., Mahmood, R., Pati, B. & Kumar, M. (2020) focused on the use of CNN to extract features and perform image recognition and detection to detect bacterial spots on bell pepper leaves that are caused by bacteria. To differentiate between two kinds of pepper leaves healthy and bacterially infected convolution neural networks were employed. For the experiment, 96.78% test accuracy was attained. The process effectively differentiated between healthy and sick leaves out of the twenty leaves that were used for screening. The paper's shortcomings include the use of a tiny dataset and the examination of only two classes. Using a dataset of 2320 photos of capsicum plants at different growth stages and imaging conditions, the authors extracted several deep learning techniques used in the study, such as Inception V3, ResNet50, and VGG16. The results demonstrate that these techniques

performed well in terms of accuracy and stability. Accuracy values of 91.60%, 92.01%, and 91.39% were attained by the pre-trained models (VGG16, ResNet50, and InceptionV3).

The authors in Jana, A. R. S. S. S. (2020) were able to examine various detection methods for pepper leaf disease detection. Examples of detection techniques include recurrent neural networks (RNNs), convolutional neural networks, and deep neural networks. The structure of the algorithm can be confounding, even though neural networks can accommodate chaotic inputs. The model thus achieved DNN 91.386%, CNN 91.436%, and RNN 91.616% in the binary detection of bacterial pepper and healthy pepper. With a score of 91.616%, the RNN performs better in the tests stated above.

The authors Kundu, N., Rani, G. & Dhaka, V. S. A (2020) presented an image processing detection system for identifying bell pepper leaf diseases. To solve the issue, pepper disease discovery was a technique utilized to detect illnesses from the leaves. The processes utilized in the paper's pepper disease detection process include pre-processing, segmentation, feature extraction, and detection. The authors used training models for deep learning. The accuracy test results for the trained models, VGG16, Inception ResNet V2, and DenseNet120, were 91.54%, 96.30%, and 96.99%, respectively.

The authors Yohannes, beniam, abdella, ayodeji and aleka melese (2019) they focused on pepper leaves and fruit disease classification using a concatenation of convolutional neural network (CNN) models to identify healthy pepper leaves, common rust in pepper leaves, gray-leaf spot in pepper leaves, fruit disease, and blight disease in pepper leaves. They achieve 95.82% accuracy but their limitation was since their scope was on leaf and fruit, they missed other diseases on the leaf and they didn't mentioned how they solve if more than one disease is occurred.

III. Proposed Model architecture

Image preprocessing, segmentation, feature extraction, and classification are all part of the design or Architecture of this system, which detects and classifies Pepercorn leaf diseases. Preprocessing includes noise reduction and resizing photos to the same size, commonly referred to as normalizing an image. In the segmentation stage, the Pepercorn leaf picture foreground and background are separated using the edge detection approach, also known as the canny edge detection method. Following segmentation, automatic feature extraction components are applied using a convolutional neural network (CNN) with VGGNet, DensNet and YOLO (You Only Look Once) architecture. Lastly, Pepercorn leaf diseases are categorized using the VGG16, VGG19, DensNet121 and YOLOv11 models with SoftMax and RELU activation function with Adam and AdamW optimizer.

A. Datasets

Pepercorn leaf images are needed to carry out image processing tasks related to the identification and categorization of Pepercorn leaf diseases. By contacting agricultural specialists and researchers, the researchers were able to obtain photographs of healthy Pepercorn leaves and Pepercorn leaves which are infected with various illnesses from various farming locations. These images are necessary for the

detection and classification of Pepercorn leaf disease. The photographs were gathered from the Burie woreda farming area, the Jabi Tehnan woreda farming area, the Dembecha woreda farming area, the West Gojjam agricultural bureau and Mankusa Abdegoma Agricultural sector. 4502 Images of both healthy and infected Pepercorn leaves were collected by the researchers. However, this data was insufficient, so the researchers used image augmentation techniques to increase the dataset and then 6013 Pepercorn leaf images were obtained. After the researchers collect the Pepercorn leaf images from different farming areas by using handheld devices, the agricultural experts classify the photos with Healthy, Anthracnose, Powdery Mildew, Black Spot, White Spot, Yellowing Leaves, Grey Mold and Mixed. Once the Agricultural experts classify the collected images accordingly, the researchers perform the rest of image preprocessing activities.

Table1: Collected Dataset summary

No	Disease Type	Number of images
1	Anthracnose	165
2	Black Spot	58
3	Grey Mold	1412
4	Healthy	1436
5	Mixed	1015
6	Powdery Mildew	189
7	White Spot	167
8	Yellowing Leaves	60
Total		4502

B. Preprocessing

Following the gathering of the Pepercorn leaf images, the researchers used a variety of preprocessing procedures, including augmentation, noise reduction, and image resizing. By eliminating noises, preprocessing aims to enhance the quality of the Pepercorn leaf images, so that the model can analyze images more effectively.

Resizing Images: Resizing an image can assist lower its pixel count, which has a number of benefits. For instance, it can shorten the time needed to train a neural network and make the model less complex. In order to standardize the image, the researchers must resize the image by increasing or decreasing the pixels. Resizing images lowers processing time and computational expenses. Images of Pepercorn leaves that the researchers take have varying resolutions and sizes. In order to process photos using deep learning methods like a convolutional neural network, the images must have a single dimension. This suggests that before supplying the learning system with photographs for this study, the researcher

should be preprocessed and shrunk to have the same proportions or standard sizes. Depending on the method and image size, the resizing can be either up or down. In order to standardize an image, the researchers in this work resize images to 224x224 and 127x127 for standardization issue and compared the final output. In image preprocessing resizing the image is carried out to reduce the training phase time[8].

Augmentation: The process of creating new data from the original dataset in order to increase the amount of training data points in that dataset is known as data augmentation. Furthermore, it facilitates the network's learning of the image's various degree orientations. Making altered versions of photos in the training dataset that fall into the same class as the original image is known as image data augmentation. Data augmentation is typically used by various researchers for data enlargement and to teach the network to recognize various image orientations. Instead of just cropping and randomly horizontally flipping patches, we decided to try more sophisticated data augmentation technique random scaling and rotation[9]. The researchers used the task of augmentation by flipping, rotating, and scaling photos in order to increase the amount of the dataset. Additionally, photos are standardized to the same image size to ensure that the pixel values are within the same range. One of the issues with short datasets is over-fitting, which occurs when a network performs well (with little error) on the training set but performs poorly (with greater error) on the test set [10]. Because there weren't enough photos gathered for eight class datasets in this investigation, the researchers employed the data augmentation technique to extend the dataset and avoid overfitting issues.

Image enhancement: The technique of enhancing an image's appearance and quality is known as image enhancement. It can be used to fix an image's imperfections or flaws or just to improve its aesthetic appeal. Digital pictures, scans, and photographs are just a few of the many types of images that might benefit from image enhancement techniques. Increasing contrast, sharpness, and color; decreasing noise and blur; and fixing distortion and other flaws are some common objectives of picture enhancement. Techniques for improving images can be used automatically with algorithms and software like OpenCV, or manually with image editing tools. The most popular technique for improving data interpretability in image viewers and improving the contribution of other automated image processing techniques is image enhancement. Image enhancement's main goal is to change an image's characteristics or properties so that it makes greater sense than the original for a specific stated task and witnesses. Model performance is significantly impacted by the quality of the images in an image dataset. Because the image collection contains a variety of components, such as illumination, the contrast on the images will be poor, resulting in insufficient features. Techniques for image enhancement can highlight the images' general or surrounding elements or features, visualize blurry images, highlight particular features of interest, suppress features that are not of interest, and increase the contrast between the features of different objects in the photographs. Techniques for improving images can improve their quality and information, improve their identification and interpretation, and make them more suitable for visual frameworks and experiences. An image must be of high quality in order to expand or improve the model's performance. The researchers employed a variety of image enhancement methods in this study, including the median filter, and the Gaussian filter, as a result of the factors

mentioned in the paragraph above. Since the Gaussian filter employs grayscale as a parameter, the researchers should convert the color image to grayscale since the image should be in grayscale rather than RGB.

Segmentation: One crucial technique that makes it possible to apply the same methods to a wide range of image sizes and types is segmentation [11]. To simplify its representation and boost its analytical value, a computer vision technique known as image segmentation splits a picture into many segments, or groups of pixels. The goal of image segmentation, a subset of digital image processing techniques, is to divide an image into distinct areas based on its characteristics. Simplifying the image for a more straightforward study is the main objective of image segmentation. One of the most well-known methods of segmentation in image processing is edge-based segmentation. One of the most important computer vision jobs is image segmentation, which divides a picture into objects, borders, or structures for more insightful analysis. In order for computers to observe and comprehend visual data in a way that is similar to how humans view and comprehend it, image segmentation is crucial to the extraction of meaningful information from images. There are different methods of segmentation like thresholding, edge detection, region based and clustering. Among the number of segmentation methods the researchers used Edge-Based segmentation. The goal of edge-based segmentation is to identify the edges of different objects in a picture. This is an important stage because it helps us to identify the characteristics of the different things in the picture because the border of the image has a lot of useful information and it assists you in removing unnecessary and unwanted information from the image, edge detection is well recognized. It significantly reduces the image's size, which facilitates analysis. In this study, the shape or boundary of a Pepercorn leaf picture is detected using the edge detection image segmentation method, with canny edge detection.

C. Feature extraction

In digital image processing, feature extraction is the process of locating and separating different features or properties from a picture that can be utilized for additional analysis, categorization, or recognition. Because raw image data might be high-dimensional and complicated, making direct manipulation difficult, this stage is essential. Extracting sufficient information to describe the type and nature of the image is one of the key concepts in image analysis. The most popular method for dealing with raw images is feature extraction, which aims to create new features from the current ones and then discard the original features in order to reduce the amount of features in an image. The majority of the information in the original collection of features should then be summarized by these new, smaller feature sets. This allows a mixture of the original set of features to be used to construct a condensed description of the original features. By creating additional features to represent images from raw pixel data, feature extraction aims to improve classification rates. In order to identify and isolate particular target areas, techniques, and procedures are utilized in the feature extraction step, which is crucial for image recognition and classification. Image features extraction step is very important in detecting and separating the desired region of images, which uses image processing techniques and algorithms[12]. An important piece of information extracted from an image that contributes to a deeper understanding

of the image is called a feature. Convolutional neural networks are one application of deep learning for signal processing issues. Its two primary applications in image processing are feature extraction and image classification. Feature extraction involves the use of many convolutional layers, followed by pooling and an activation function. The pooling function, convolution layer, and two cascading layers make up each of the multiple similar steps in the CNN feature extraction process. One highly effective method for automatically extracting the best features from a large number of training datasets is to use CNN for feature extraction. Convolutional neural networks are superior for automatically extracting deep features.

D. Classification

The vital role that picture classification plays in contemporary technologies makes it stand out. It comprises giving an entire image a label or tag by utilizing previously gathered training data from previously labeled images. Grouping the nodules and non-nodules according to the chosen features is the aim of this stage [13]. Even though it may appear simple at first, the process entails pixel-by-pixel image analysis to get the optimal label for the entire image. This provides us with valuable data and insights that enable us to make informed choices and accomplish meaningful outcomes. Following feature extraction and learning of the distinguishing characteristics, classification is carried out using the knowledge gained during the training phase. Classification is the last phase of this investigation. One classification method that is frequently used for image categorization is a convolutional neural network. The VGG-16, VGG-19, DenseNet121 and YOLOv11n classifier, convolutional neural network with SoftMax activation parameters, is employed in this investigation.

IV. Experimental Setup and Result Analysis

A. Experimental Setup

The model was repeatedly trained using a convolutional neural network (CNN) with a dataset of images of varying sizes, including 224x224 and 127x127, and with varying splitting ratios, such as 70%/30%, 80%/20%, and 90%/10%. In addition to this, the researchers check the model by using different hyper parameter values such as batch size 16, 32 and 64 and learning rate 0.1, 0.01 and 0.001 for VGG16, VGG19, DenseNet121 and YOLOv11. After this, the researchers compared the results of the models. The researchers must determine whether the model can generalize data that hasn't been seen yet after it has been developed and trained. This enables us to determine whether the model is doing well solely on trained data and not on test data, or whether it is categorizing well on new data. Using training data to evaluate the model is not recommended. 10%, 20%, and 30% of the overall dataset's Pepercorn leaf photos are utilized as test data by the researchers to evaluate the model's accuracy.

B. Models

The VGG-16, VGG-19, DenseNet121 and YOLOv11n classifier, convolutional neural network with SoftMax activation parameters, is employed in this investigation. Feature extraction and description is the

process of locating and characterizing characteristics in a leaf image. Features can be extracted from the leaf image using its spectral information, texture, intensity, and shape. The traits are then described using mathematical or statistical models. Feature extraction and description is an essential step in many plant image analysis applications, leaf disease detection, and leaf trait extraction. Multilayer feed-forward with back propagation support vector machine classifiers based on artificial neural networks were used in the study to recognize plant disease images, and their behavior was examined in terms of how well the classifiers performed in identifying various plant diseases [14]. To determine if a neuron is active or not, the activation function is employed. This suggests that it will use comparable mathematical tasks to determine whether or not the neuron's input to the network is relevant during the prediction period. The activation function's job is to create output from a set of input values that are supplied to a layer.

Due to the nature of the problem, a convolutional neural network VGG16, VGG19, DenseNet121 and YOLOv11 classifier with SoftMax activation functions were used for this work.

C. Results

DensNet121 Experimental Result

YOLOv11n Experimental Result

V. Discussions

Among those four algorithms the researchers try to make an experiment with data split ratio 70% / 30%, 80% / 20% and 90% / 10% batch size of 16, 32 and 64 learning rate of 0.1, 0.01 and 0.001. According to this experiment, in the case of VGG16 the highest accuracy achieves with batch size 64, data split ratio 80% / 20% and learning rate of 0.001 that is 98.03%. In the case of VGG19 the highest accuracy achieves with batch size 64, data split ratio 90% / 10% and learning rate of 0.001 that is 98.05%. In the case of DenseNet121 the best accuracy achieves with batch size 64, data split ratio 80% / 20% and a learning rate of 0.001 that is 98.75%. In this study the researchers performs a number of experiments on different CNN models like VGG16, VGG19 and DenseNet121, other real time models like YOLOv11 by using different parameters to decide the best model with best accuracy. So according to the experimental results found by the researchers DenseNet121 is greater than VGG19 (98.75% > 98.05%) and VGG19 is greater than VGG16 (98.05% > 98.03%). So DenseNet121 has achieved the best accuracy of 98.75% than that of VGG16 and VGG19 among the CNN models. But the researchers used another real time model called YOLOv11 and this model achieved an accuracy_top1 of 100% and a mean average precision (MAP) of 99.03%. When we see the results in terms of performance (speed) the YOLOv11 model performs its preprocessing and post processing tasks in 1Hr, 096 Seconds, while DenseNet takes a speed of 2Hr, 35seconds.

Table 2: parameters used in this study

Parameters	Values
Learning Rate	0.001, 0.01 and 0.1
Dropout value	0.5
Batch size	16, 32 and 64
Activation	SoftMax
Number of Class	8
Epoch	30, 35, 50
Optimizer	Adam and AdamW

The comparison between CNN models that the researchers used in terms of performance and accuracy.

Table 3: Performance comparisons of the models

Models	Accuracy	Speed(preprocess + post process)
VGG16	98.03%	2Hr, 17 Minutes
VGG19	98.05%	2Hr, 32Minutes
DenseNet121	98.75%	2Hr, 35 Seconds
YOLOv11n	Top_1 up to Top_5=100% , MAP=99.03	1Hr, 096 Seconds

VI. Conclusions

Now a day's agricultural production is suffering from several problems, but the major problem is Pepercorn diseases such as Anthracnose, Black spot, Powdery mildew, White Spot, Yellowing Leaves, Grey Mold and the hybrid of two or more diseases which leads to product loss. Also, the shortage of problem-solving tools in developing countries like our country Ethiopia has a devastating impact on their development and quality of life. Therefore, there is a serious need to detect the disease at an early stage with affordable and easy-to-use technological solutions. To identify the diseases easily, the researchers have developed the model in a deep learning approach by using the convolutional neural network and image processing techniques. The researchers have presented a convolutional neural network model to detect and classify disease from the Pepercorn leaf using a leaf image as an input. The dataset was collected from the west gojjam agricultural bureau Burie zuria woreda farming area, Jabi Tehnan woreda farming area, Dembecha zuria woreda farming area. The Collected data contains healthy and infected/unhealthy Pepercorn leaf images. Originally, the researchers had acquired 4502 healthy and unhealthy Pepercorn leaf images. But, this data is not sufficient, so applied image augmentation techniques to increase the dataset and the researchers got 6013 Pepercorn leaf image data. The researchers have applied the Gaussian blurring and canny edge detection methods to enhance and

segment images. In this study, the researchers used a convolutional neural network with a VGG16, VGG19, DenseNet121 and YOLOv11 deep learning model for feature extraction and classification purposes and SoftMax as an activation function in a fully connected layer. As the researchers have discussed in the previous chapter discussion section, the VGG16 and VGG19 models were trained by changing different hyper parameter values. The hyper parameters with their values are image size with a value of 224x224 batch size with a value of 64, 32 and 16, dataset splitting ratio 70%/30%, 80%/20% and 90%/10% and learning rate with a value of 0.1, 0.01, and 0.001. Therefore, image size 224x224, number of epoch 50, batch size 64, 90%/10% dataset split ratio, and learning rate 0.001 are optimal hyper parameter values for this work and the result achieved 98.05% average accuracy. But the DenseNet121 model achieved more better accuracy of 98.75% and lastly a very proud real time model has used and achieved a 99.03% mean average precision(MAP). When we see the results in terms of performance (speed) the YOLOv11 model performs its preprocessing task in 4.3ms, while DenseNet takes a speed of 35seconds.

Declarations

Author Contribution

the author has contributed a substantial contributions to the conception or design of the work, the acquisition, analysis, or interpretation of data, the creation of new software used in the work and have drafted the work or substantively revised it

Funding Declaration

The authors declare that no funds were received for this study.

Data Availability: -

The data that support the findings of this study are available on request from the corresponding author.

Ethics declarations

Competing interests

The authors declare no competing interests.

References

1. Y. Zeng, Y. Zhao, Y. Yu, Y. Tang, and Y. Tang, "Pepper Disease Detection Model Based on Convolutional Neural Network and Transfer Learning," *IOP Conf. Ser. Earth Environ. Sci.*, vol. 792, no. 1, 2021, doi: 10.1088/1755-1315/792/1/012001.
2. A. Fuentes, S. Yoon, S. C. Kim, and D. S. Park, "A robust deep-learning-based detector for real-time tomato plant diseases and pests recognition," *Sensors (Switzerland)*, vol. 17, no. 9, 2017, doi:

10.3390/s17092022.

3. S. H. Lee, C. S. Chan, P. Wilkin, and P. Remagnino, "Deep-plant: Plant identification with convolutional neural networks," *Proc. - Int. Conf. Image Process. ICIP*, vol. 2015-Decem, pp. 452–456, 2015, doi: 10.1109/ICIP.2015.7350839.
4. S. Savary and B. M. Cooke, "Plant disease epidemiology: Facing challenges of the 21st century: Under the aegis of an international plant disease epidemiology workshop held at Landernau, France, 10-15th April, 2005," *Plant Dis. Epidemiol. Facing challenges 21st Century Under aegis an Int. Plant Dis. Epidemiol. Work. held Landernau, Fr. 10-15th April. 2005*, no. March 2014, pp. 1–138, 2006, doi: 10.1007/1-4020-5020-8.
5. K. Alamerie, M. Ketema, and F. Gelaw, "Kumilachew Alamerie, Mengistu Ketema, Fekadu Gelaw. Risks in Vegetables Production from the Perspective of Smallholder Farmers: The Case of Kombolcha Woreda," *For. Fish. Spec. Issue Agric. Ecosyst. Environ.*, vol. 3, no. 1, pp. 1–5, 2014, doi: 10.11648/j.aff.s.2014030601.11.
6. Y. A. Bezabih, A. O. Salau, B. M. Abuhayi, A. A. Mussa, and A. M. Ayalew, "CPD-CCNN: classification of pepper disease using a concatenation of convolutional neural network models," *Sci. Rep.*, vol. 13, no. 1, pp. 1–13, 2023, doi: 10.1038/s41598-023-42843-2.
7. M. Dai et al., "Pepper leaf disease recognition based on enhanced lightweight convolutional neural networks," *Front. Plant Sci.*, vol. 14, no. August, pp. 1–18, 2023, doi: 10.3389/fpls.2023.1230886.
8. W. Shivali Amit and R. Hari Krishnan, "A deep learning-based approach in classification and validation of tomato leaf disease," *Trait. du Signal*, vol. 38, no. 3, pp. 699–709, 2021, doi: 10.18280/ts.380317.
9. P. S and F. S. D. A, "Image based Plant leaf disease detection using Deep learning," *Int. J. Comput. Commun. Informatics*, vol. 3, no. 1, pp. 53–65, 2021, doi: 10.34256/ijccci2115.
10. E. A. Smirnov, D. M. Timoshenko, and S. N. Andrianov, "Comparison of Regularization Methods for ImageNet Classification with Deep Convolutional Neural Networks," *AASRI Procedia*, vol. 6, pp. 89–94, 2014, doi: 10.1016/j.aasri.2014.05.013.
11. M. Prabhakar, R. Purushothaman, and D. P. Awasthi, "Deep learning based assessment of disease severity for early blight in tomato crop," *Multimed. Tools Appl.*, vol. 79, no. 39–40, pp. 28773–28784, 2020, doi: 10.1007/s11042-020-09461-w.
12. M. Kunaver and J. F. Tasič, "Image feature extraction - An overview," *EUROCON 2005 - Int. Conf. Comput. as a Tool*, vol. I, no. May, pp. 183–186, 2005, doi: 10.1109/eurcon.2005.1629889.
13. Karunia, "No Title," vol. 4, no. June, p. 2016, 2016.
14. J. D. Pujari, R. Yakkundimath, and A. S. Byadgi, "SVM and ANN Based Classification of Plant Diseases Using Feature Reduction Technique," *Int. J. Interact. Multimed. Artif. Intell.*, vol. 3, no. 7, p. 6, 2016, doi: 10.9781/ijimai.2016.371.

Figures

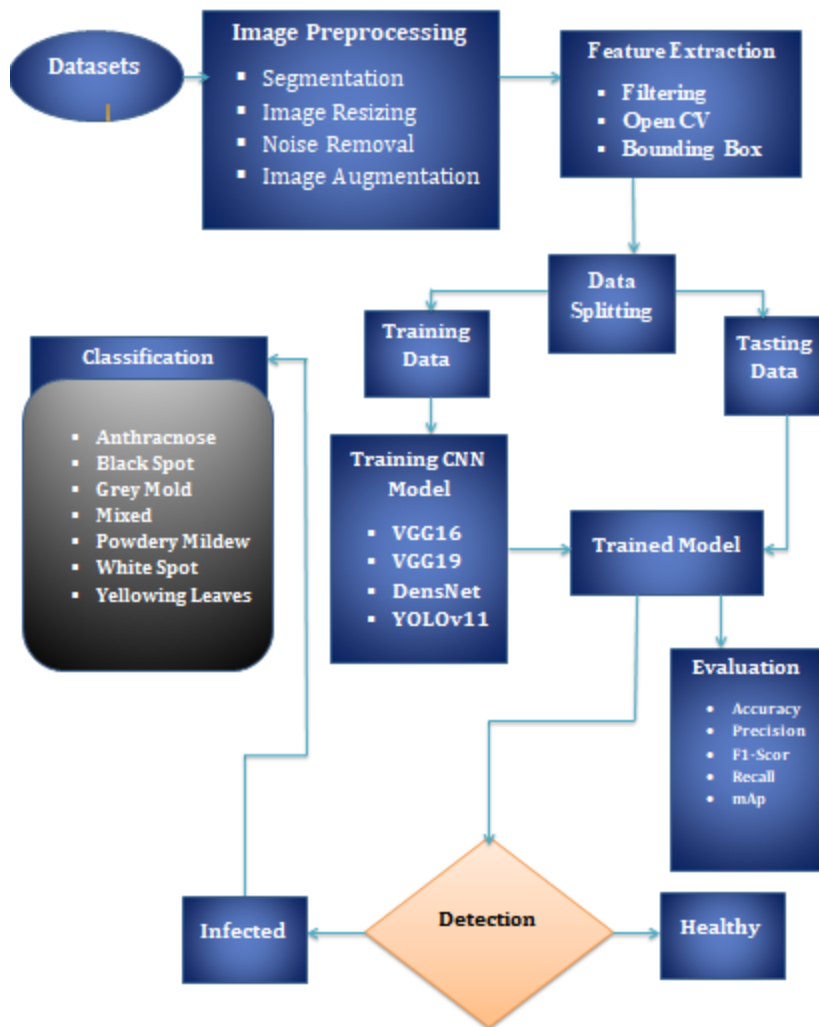


Figure 1

The Proposed Model Architecture

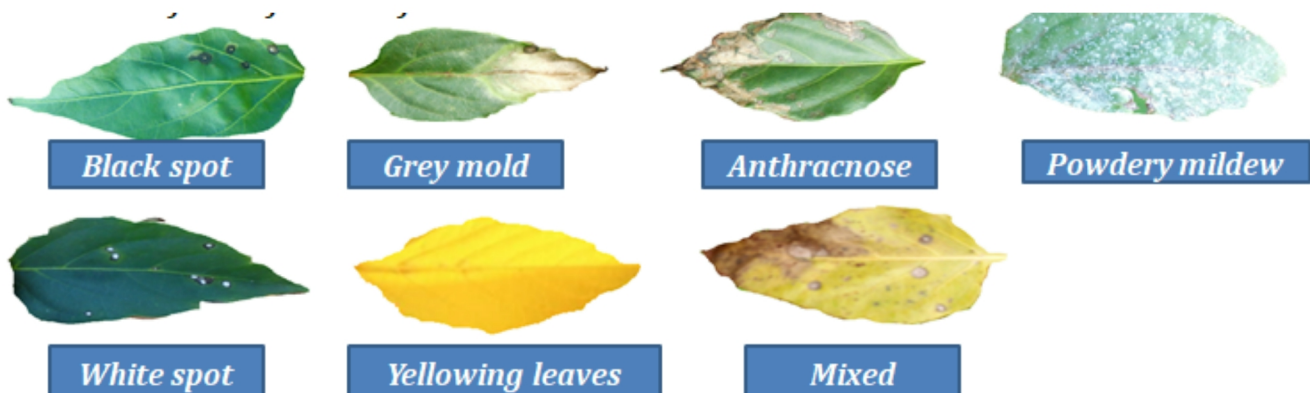
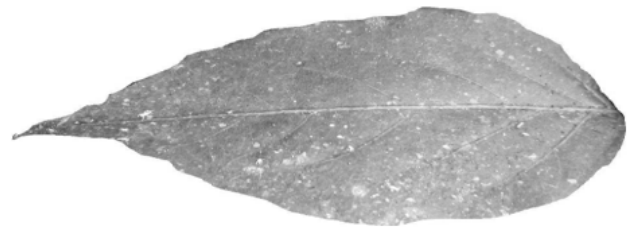
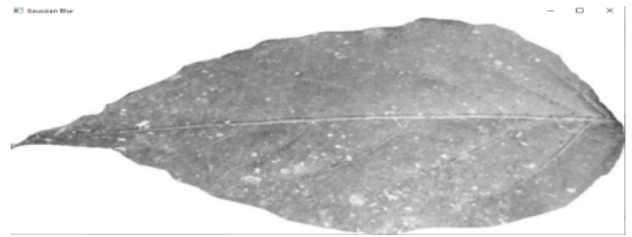
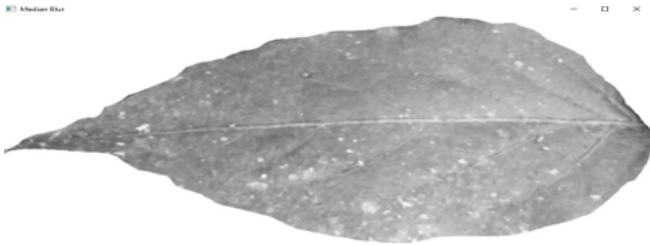


Figure 2

sample images from the dataset

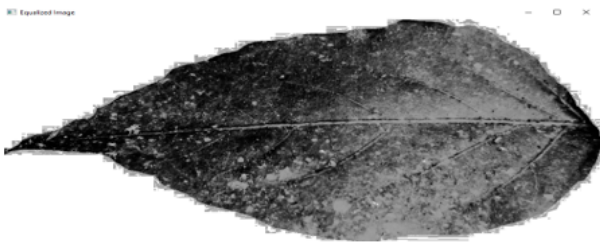


A) RGB Powdery Mildew infected leaf B) Grayscale Powdery Mildew source image



C) Median Blur

D) Gaussian Blur



E) Histogram Equalization

Figure 3

A) RGB image B) Grayscale C) Median Blur D) Gaussian Blur and E) Histogram equalization process

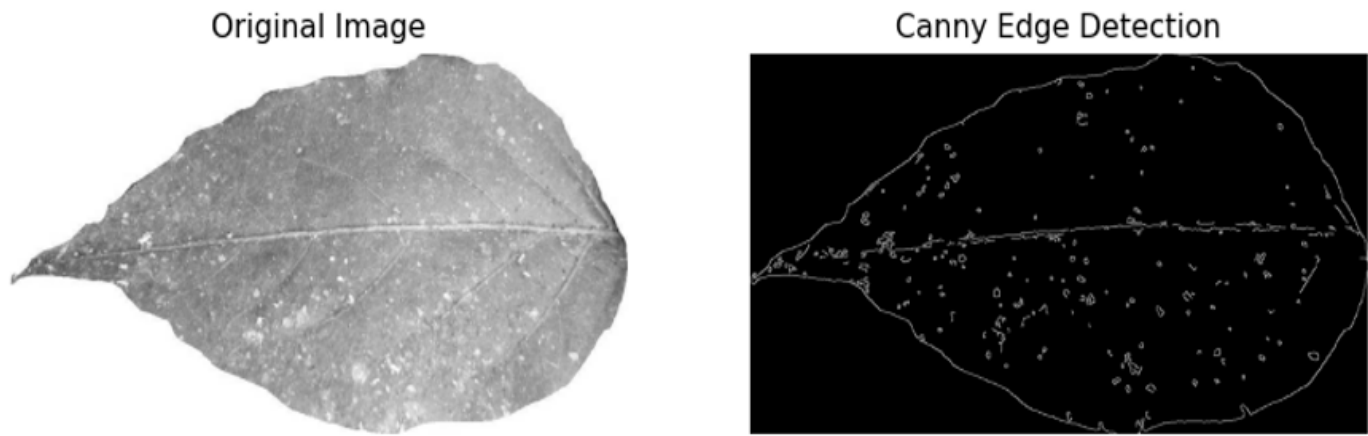


Figure 4

Image Segmentation process with Canny edge Detection

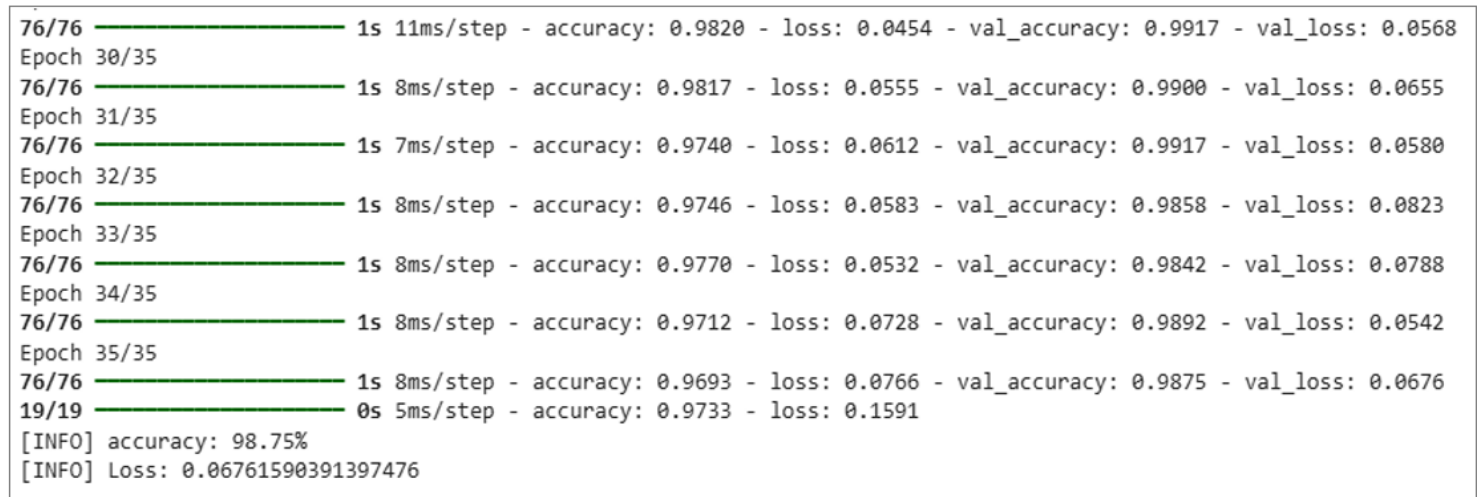


Figure 5

The accuracy and loss information for DenseNet121

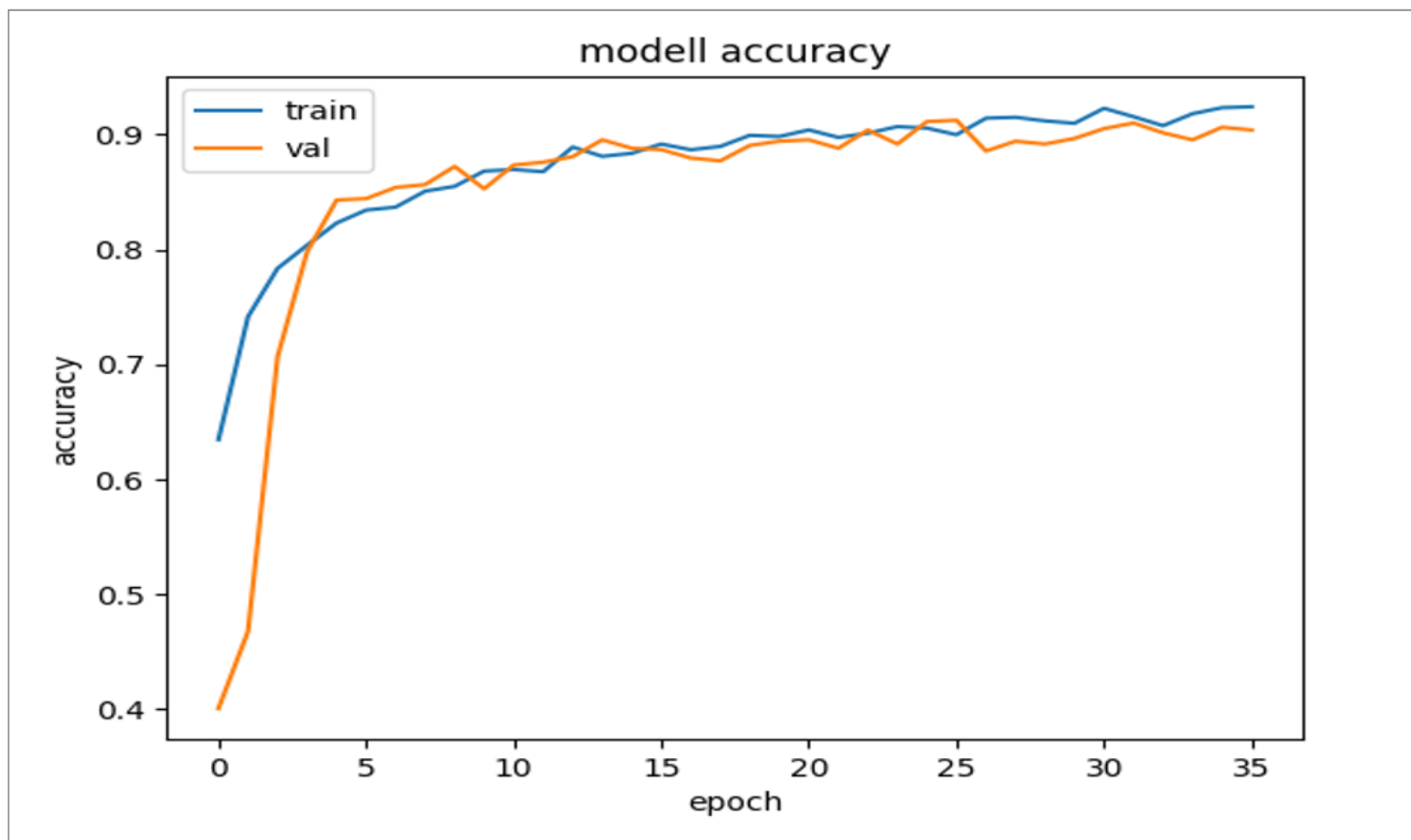


Figure 6

Model accuracy graph for DenseNet121

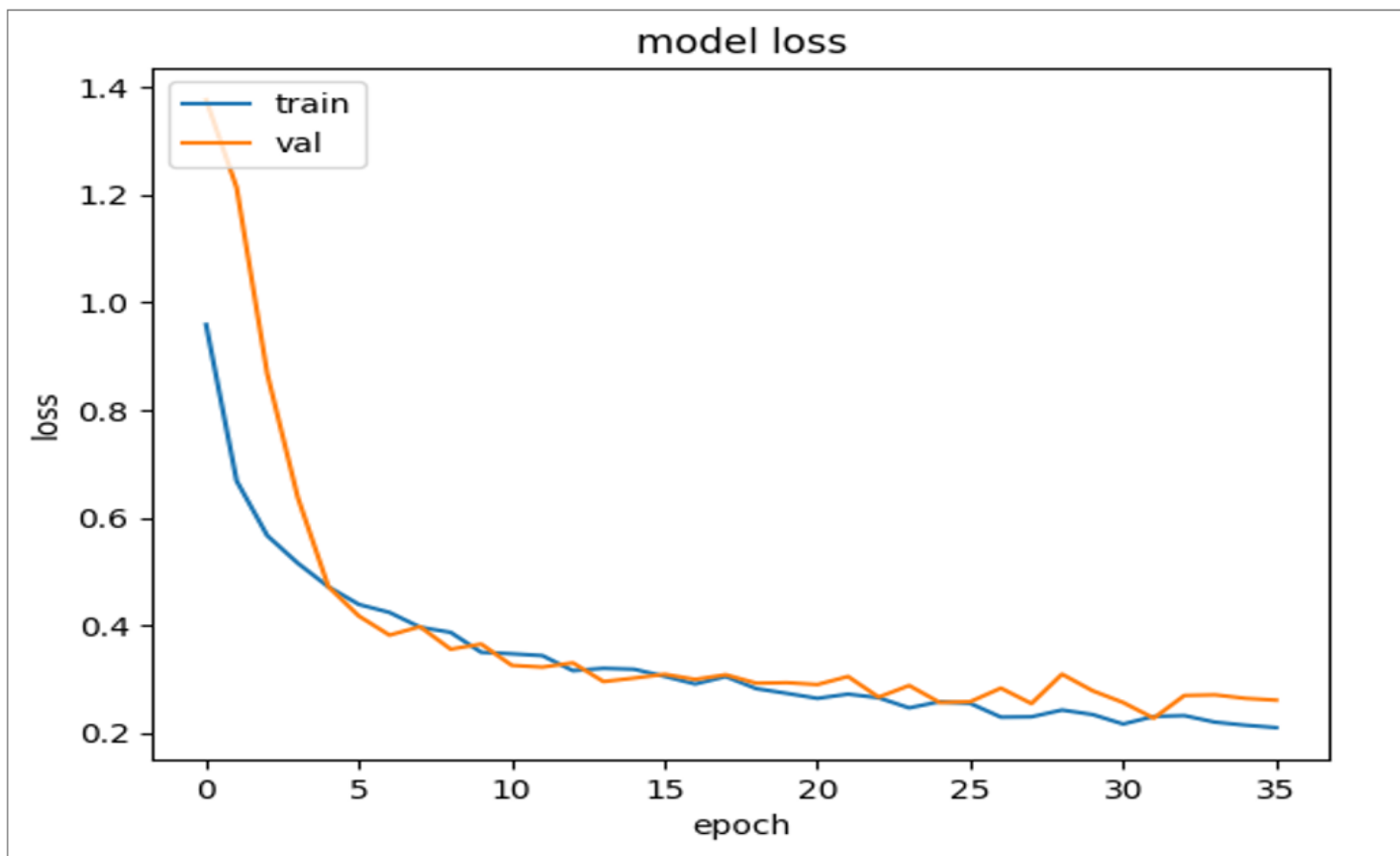


Figure 7

Model loss graph for DenseNet121

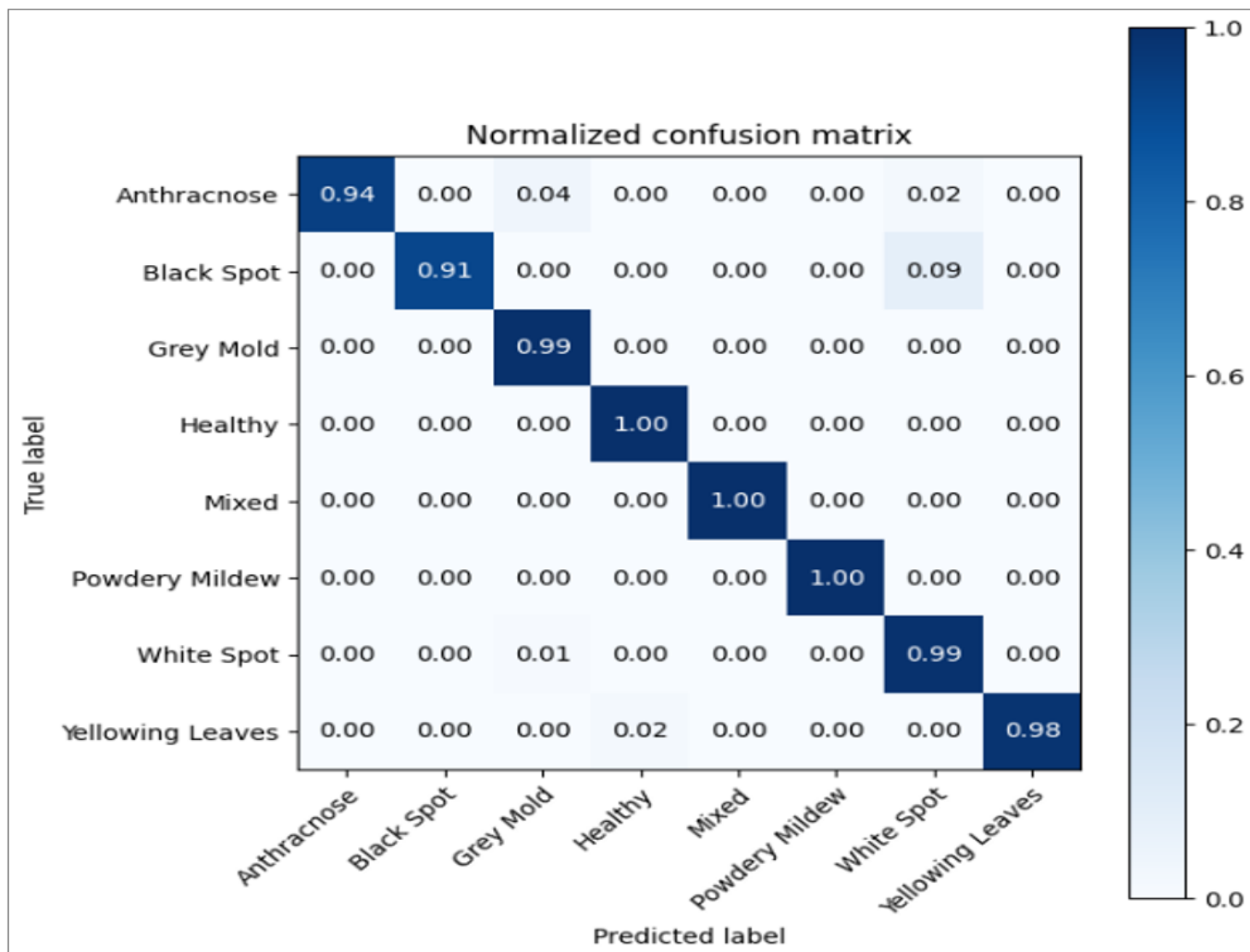


Figure 8

Normalized confusion matrix for DenseNet121

Accuracy: 0.9875104079933389
Macro Average:
precision: 0.9874144502755193
recall: 0.9775042797750061
f1-score: 0.9819504607231468
Weighted Average:
precision: 0.9879969777251749
recall: 0.9875104079933389
f1-score: 0.9874797607823218

Figure 9

Accuracy, Precision and F1-score for DenseNet121

```
50 epochs completed in 1.096 hours.
Optimizer stripped from runs/classify/train3/weights/last.pt, 3.2MB
Optimizer stripped from runs/classify/train3/weights/best.pt, 3.2MB

Validating runs/classify/train3/weights/best.pt...
Ultralytics 8.3.145 Python-3.11.12 torch-2.6.0+cu124 CUDA:0 (Tesla T4, 15095MiB)
YOLO11n-cls summary (fused): 47 layers, 1,536,272 parameters, 0 gradients, 3.2 GFLOPs
WARNING ⚠ Dataset 'split=val' not found, using 'split=test' instead.
train: /content/PepercornLeafDetectionClassficti.v2i.folder/train... found 2179 images in 8 classes ✓
val: /content/PepercornLeafDetectionClassficti.v2i.folder/test... found 293 images in 8 classes ✓
test: /content/PepercornLeafDetectionClassficti.v2i.folder/test... found 293 images in 8 classes ✓
      classes  top1_acc  top5_acc: 100%|██████████| 10/10 [00:04<00:00, 2.27it/s]
           all         1         1
Speed: 1.1ms preprocess, 3.2ms inference, 0.0ms loss, 0.0ms postprocess per image
Results saved to runs/classify/train3
```

Figure 10

Accuracy information for YOLOv11n

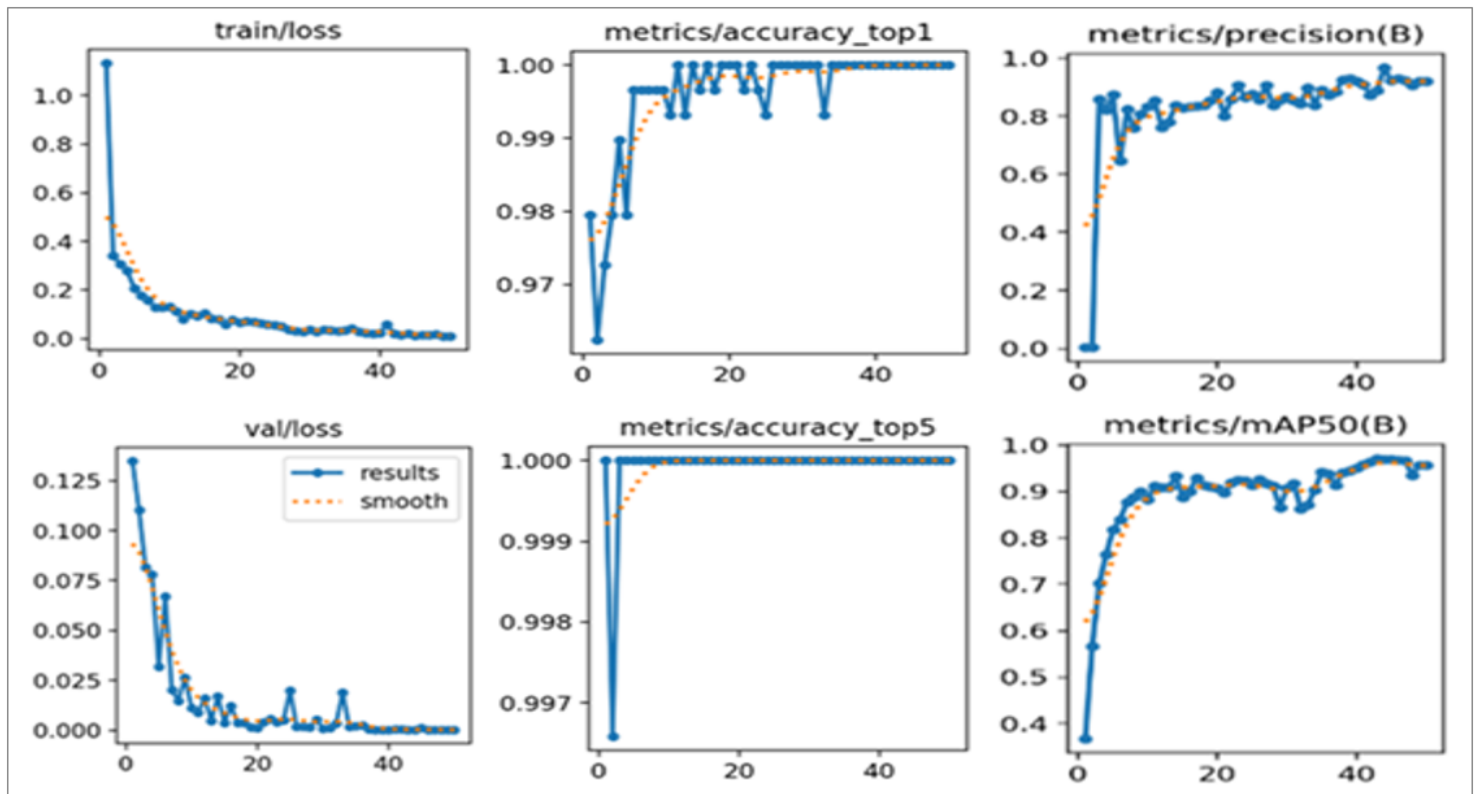


Figure 11

The graph information for YOLOv11n

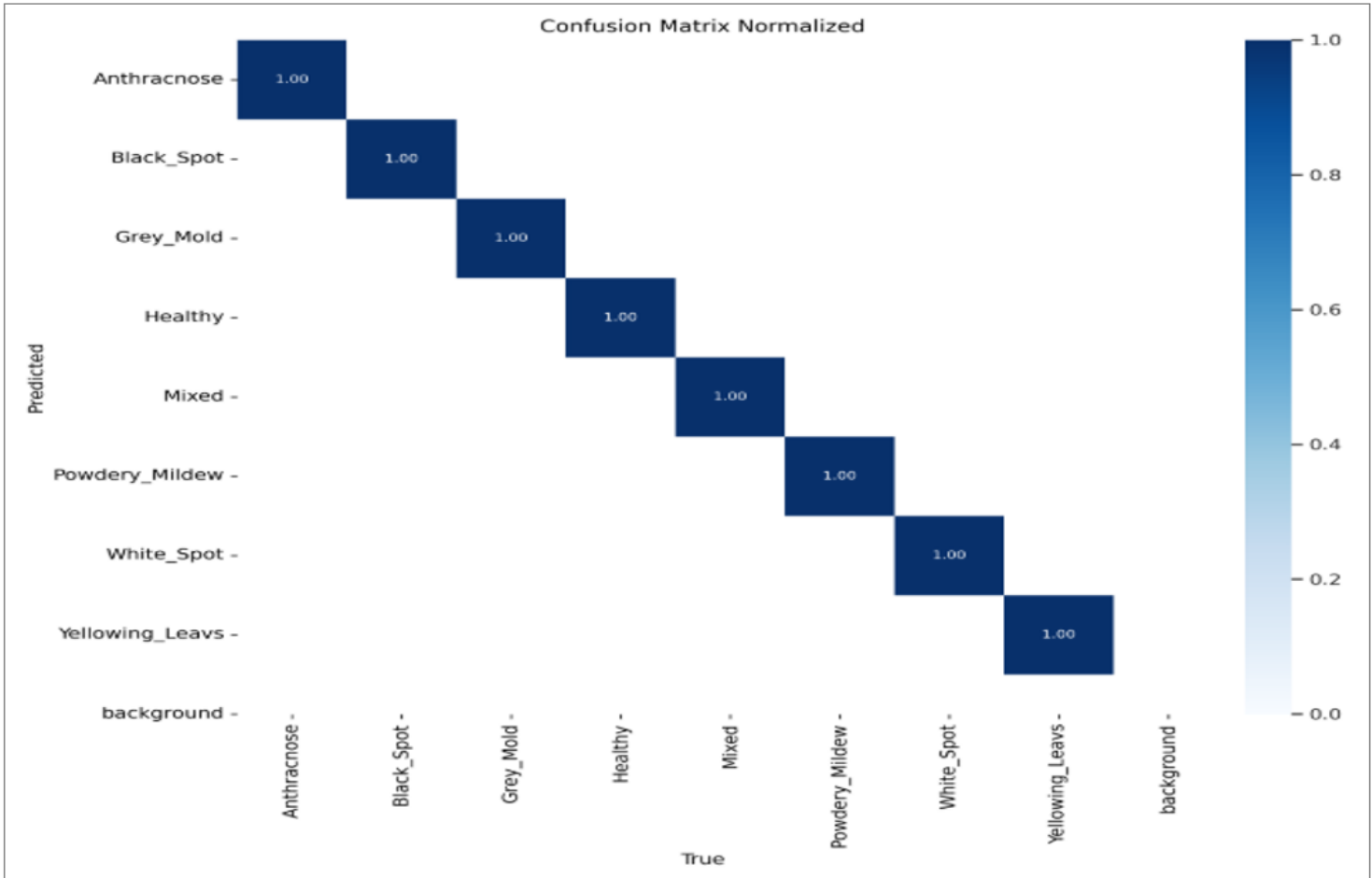


Figure 12

The Normalized Confusion Matrix for YOLOv11n



Figure 13

Sample Test Image results for YOLOv11n

