REPUBLIC OF TURKEY
ADANA ALPARSLAN TÜRKEŞ SCIENCE AND TECHNOLOGY
UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
DEPARTMENT OF ELECTRIC AND ELECTRONIC ENGINEERING

CLASSIFICATION OF PEPPER LEAF DISEASES USING VGG16.NET

SÜLEYMAN ÇETİNKAYA
MASTER OF THESIS

ADANA 2025

REPUBLIC OF TURKEY
ADANA ALPARSLAN TÜRKEŞ SCIENCE AND TECHNOLOGY UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
DEPARTMENT OF ELECTRIC AND ELECTRONIC ENGINEERING

CLASSIFICATION OF PEPPER LEAF DISEASES USING VGG16.NET

SÜLEYMAN ÇETİNKAYA
MASTER OF THESIS

SUPERVISOR
Dr.Öğr.Üyesi AMİRA TANDİROVİÇ GÜRSEL

ADANA 2025

ÖZET

VGG16.Net Kullanarak Biber Yaprağı Hastalıklarının Sınıflandırılması

Süleyman ÇETİNKAYA
Elektrik Elektronik Mühendisliği Anabilim Dalı
Danışman: Dr.Öğr.Üyesi AMİRA TANDİROVİÇ GÜRSEL

Ocak 2025, 63 pages

Tarım, insan yaşamını sürdürebilmek ve ulusların kalkınmasını ve verimliliğini desteklemek için en önemli kaynaklardan biridir. Artan sebze ve meyve taleplerini karşılamak için üretim verimliliğini artırmaya yönelik araştırmalar yapılması ve önleyici tedbirler alınması hayati önem taşımaktadır. Bu tedbirler arasında bitki hastalıklarının ele alınması öncelikli bir konudur. Bitki hastalıklarının tanınması ve teşhis edilmesi, tarımsal verimliliğin artırılması, hastalıkların yayılmasının önlenmesi ve ekonomik kayıpların en aza indirilmesi açısından kritik bir öneme sahiptir. Bu çalışma, biber yaprağı hastalıklarını sınıflandırmayı ve teşhis etmeyi hedeflemekte, böylece bu hastalıkların etkilerini azaltmak için erken müdahale imkanı sağlamaktadır. Küf, akar, tırtıl, kurt, yaprak biti ve yaprak yanıklığı gibi hastalıklar, özellikle seralarda yetiştirilen biberlerin yapraklarını etkileyerek karakteristik izler bırakmaktadır. Bu belirgin özellikler, biber yaprağı hastalıklarının daha hızlı ve doğru teşhisini sağlayan 7 sınıflı bir sınıflandırma modelinin geliştirilmesinde temel alınmıştır.

Geleneksel doğrudan ölçüm yöntemleri basit ve güvenilir olsa da genellikle zaman alıcı ve emek yoğun olmaktadır. Bu sınırlamaların üstesinden gelmek için görüntü işleme konusunda Evrişimli Sinir Ağı (CNN) algoritmaları kullanılmaktadır. Ana ön işleme adımı, yeşil ve sarı tonlarındaki farklılıkları artırarak hastalık sınıfları arasındaki ayrımı kolaylaştırmak için renk geliştirme işlemini içermektedir. Görüntülerin canlılığını, kontrastını ve genel renk özelliklerini iyileştiren yenilikçi bir algoritma tasarlanmıştır. Sınıflandırma süreci, karmaşık veri setleriyle yüksek doğruluk oranı elde etmesiyle tanınan 19 katmanlı VGGNet mimarisi ile gerçekleştirilmiştir. Eğitim süresini kısaltmak için VGGNet'in önceden eğitilmiş katmanları dondurulmuş ve modelin ince ayarı için ek katmanlar eklenmiştir.

Önerilen model, bu çalışma için özel olarak derlenmiş bir veri seti üzerinde test edilmiştir. Bildiğimiz kadarıyla, bu çalışma yaprak biti ve tırtıl sınıflarının teşhisine odaklanan ilk çalışmadır. Model, 7 sınıflı teşhis görevi için oldukça tatmin edici bir performans olarak kabul edilen %92.00 ortalama doğruluk oranına ulaşmıştır. Yanlış sınıflandırmaların ana kaynağı, yaprak biti sınıfında, eğitim için kullanılan örneklerin sınırlı sayıda olmasından kaynaklanmaktadır.

Anahtar Kelimeler: Derin öğrenme, Biber, Bitki Yaprak Hastalığı, Görüntü Sınıflandırma, Renk İyileştirme, Önceden Eğitilmiş VGG16 Modeli, Evrişimsel Sinir Ağı (CNN)

ABSTRACT

Classification of Pepper Leaf Diseases Using VGG16.Net

Süleyman ÇETİNKAYA
Department of Electric and Electronic Engineering
Supervisor: Dr.Öğr.Üyesi AMİRA TANDİROVİÇ GÜRSEL

January 2025, 63 pages

Agriculture is one of the most essential resources for sustaining human life and fostering the development and productivity of nations. To meet increasing demands for vegetables and fruits, it is imperative to conduct research and implement preventive measures to enhance production efficiency. Among these measures, addressing plant diseases stands out as a top priority. Recognizing and diagnosing plant diseases is critical for improving agricultural productivity, preventing the spread of infections, and minimizing economic losses. This study aims to classify and diagnose pepper leaf diseases, enabling early intervention to mitigate their impact. Diseases such as mildew, mites, caterpillars, worms, aphids, and leaf burn predominantly affect greenhouse peppers, leaving distinctive marks on their leaves. These unique features form the basis for developing a 7-class classification model that facilitates faster and more accurate diagnosis of pepper leaf diseases.

While traditional direct measurement methods are straightforward and reliable, they are often time-consuming and labor-intensive. To overcome these limitations, Convolutional Neural Network (CNN) algorithms for image processing have been employed. A key preprocessing step involves color enhancement to amplify variations in green and yellow hues, improving the differentiation between disease classes. An innovative algorithm has been designed to enhance the vibrancy, contrast, and overall color properties of the images, ensuring optimal feature extraction. The classification process leverages the 19-layer VGGNet architecture, renowned for its high accuracy in handling complex datasets. To reduce training time, the pre-trained layers of VGGNet were frozen, and additional layers were added for fine-tuning.

The proposed model was evaluated on a proprietary dataset specifically compiled for this study. To the best of our knowledge, this is the first study to focus on diagnosing aphid and caterpillar classes within this context. The model achieved an average accuracy of 92.00%, considered highly satisfactory for a 7-class diagnostic task. The primary source of misclassification was observed in the aphid class, attributed to the limited number of samples available for training.

Keywords: Deep learning, Pepper, Plant Leaf Disease, Image Classification, Color enhancement, Pretrained VGG16 model, Convolutional Neural Network (CNN)

ACKNOWLEDGEMENTS

Primarily, I would like to thank my supervisor, Dr.Öğr.Üyesi AMİRA TANDİROVİÇ GÜRSEL, for providing guidance and feedback throughout this thesis.

I would also like to thank my thesis committee members, Doç. Dr. Gökay DİŞKEN and Doç. Dr. Ahmet AYDIN for their constructive comments.

Finally, I would like to thank my esteemed professor and my beloved family, who always supported and encouraged me while writing this thesis, and express my deepest gratitude.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

RBG : Red Green Blue
CNN : Convolutional Neural Network
LVQ : Learning Vector Quantization
DNN : Deep Neural Network
RELU : Rectified Linear Unit
KNN : K- Nearest Neighbors
SVM : Support Vector Machine
AI : Artificial Intelligence
DL : Deep Learning

1. INTRODUCTION

1.1. Agriculture in Turkey

Agriculture is essential for the sustainability of human societies, as it provides fundamental resources and generates economic opportunities. As an industrial practice encompassing the cultivation, management, and harvesting of plant and animal products, agriculture has significantly shaped economic, social, and environmental dimensions throughout history [1]. In Turkey, the agricultural sector constitutes a critical foundation of the national economy, serving as a primary source of livelihood for a substantial proportion of the population. Agriculture influences society in multiple ways, including the provision of food, living spaces, employment opportunities, and raw materials. Among the notable agricultural commodities, the pepper plant holds particular importance due to its extensive applications across various industries. In Figure 1 below, there is an image of a pepper garden cultivated in Turkey. The rapid growth of the country’s population has amplified the demand for more efficient and higher-quality agricultural products. However, the widespread occurrence of plant diseases poses a considerable challenge to agricultural productivity, resulting in diminished yields and significant economic losses [2].

Environmental factors play a pivotal role in determining the susceptibility of pepper plants to leaf diseases. Notably, imbalances in soil pH weaken the plant's defence mechanisms, thereby creating an environment conducive to pathogen attacks. An acidic or alkaline pH disrupts the plant's capacity to absorb essential nutrients, rendering it more vulnerable to diseases. Furthermore, climatic conditions such as humidity and temperature exert a profound influence on the proliferation and progression of plant diseases. For instance, elevated humidity levels accelerate the growth of fungal infections such as mold, while fluctuations in temperature can expedite the advancement of certain diseases [3]. These environmental conditions increase the susceptibility of pepper plants to diseases, adversely affecting their overall health and productivity.

Figure 1. Obtained From the Pepper Garden

In addition to environmental factors, biotic influences also contribute to the onset of leaf diseases in pepper plants. Pathogens such as viruses, bacteria, fungi, and other harmful organisms are widely recognized for causing various plant diseases, often targeting specific plant parts, including the leaves. Pests such as aphids and worms are particularly detrimental, as they damage the leaves and compromise the plant’s defenses, increasing its vulnerability to infections. Aphids, for instance, feed on the undersides of leaves, disrupting nutrient absorption and resulting in symptoms such as leaf yellowing and premature leaf drop. Similarly, worms damage the plant's root system, inducing stress and weakening its immune responses. These pests not only inflict direct harm but also facilitate the spread of pathogens by creating favorable conditions for infections. Figure 2 below depicts an image illustrating the manual diagnosis of pepper disease in an agricultural context. This combined impact of environmental and biotic factors significantly undermines the plant's ability to resist diseases, ultimately leading to reduced crop yields and inferior product quality [4].

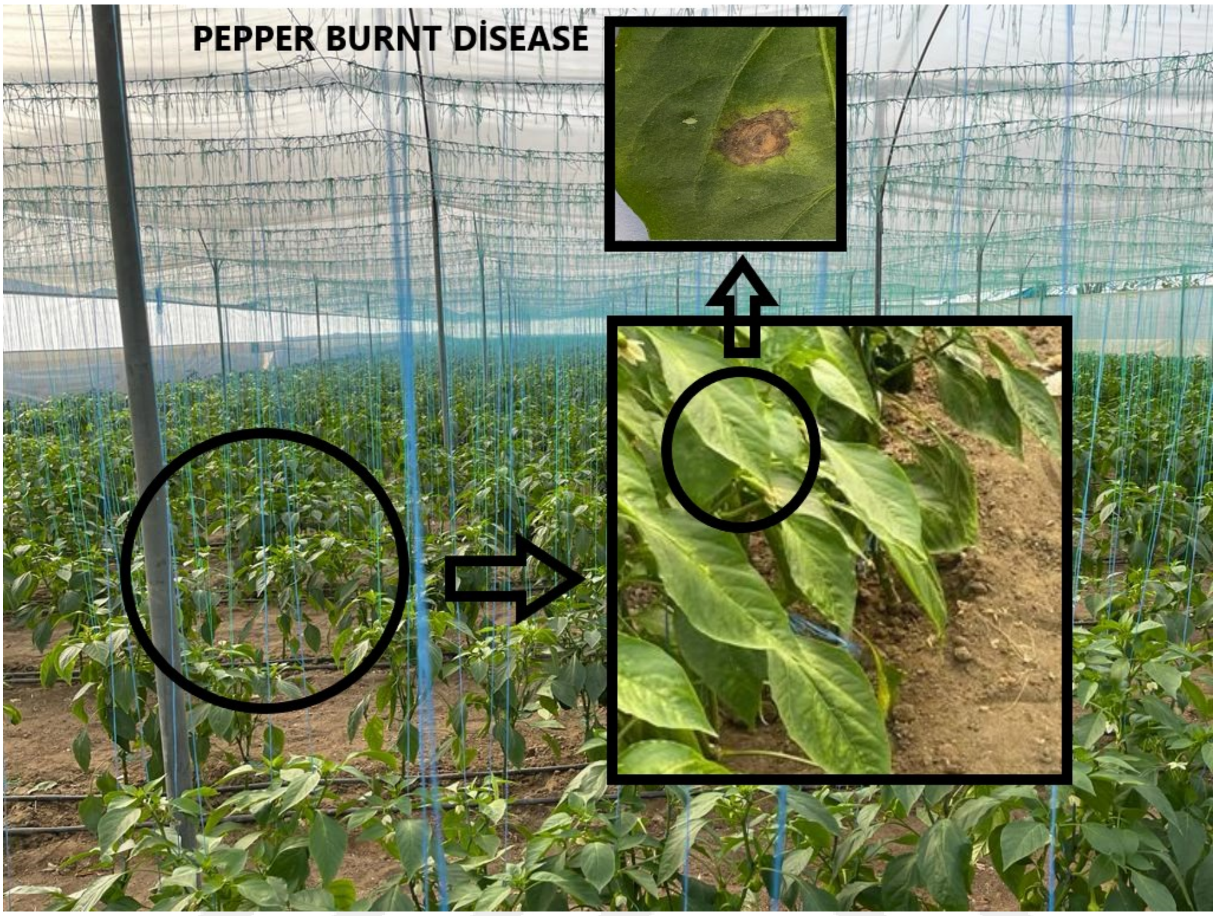


Figure 2. Burnt Pepper Leaf Disease

1.1.1. Pepper Diseases Classes

Plant leaf diseases significantly affect the external appearance of leaves. Viral, bacterial, and infectious diseases often exhibit overlapping symptoms, making accurate detection and diagnosis challenging. Healthy pepper leaves are characterized by a vibrant, uniform, light green color with a shiny appearance. When an optimal environment is provided for the pepper plant, both healthy leaves and fruits develop, resulting in efficient production. Figure 3 below shows the distribution of diseased and healthy classes of pepper leaf disease images used for model training. The dataset used in this study includes various types of pepper leaves, such as healthy, burnt, mildew, mite, caterpillar, worm, and aphid leaves. Representative sample images from the dataset are presented in the figure for reference [5]. The content of the dataset is shown by a bar chart in Figure 3.

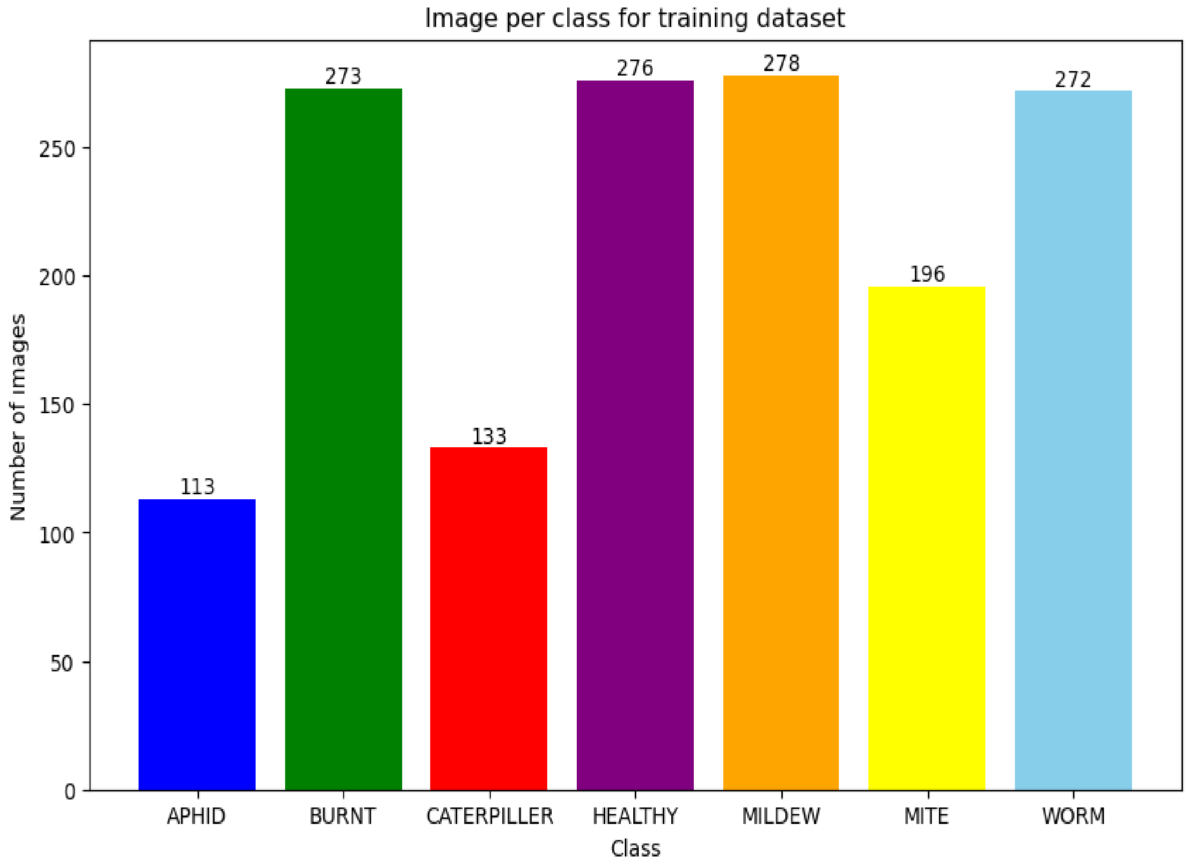


Figure 3. Grouping of Pepper Leaf Diseases Number

1.1.1.1. Mite

Wrinkling observed on pepper leaves is often associated with stress conditions, infections, or nutrient deficiencies. Notably, deficiencies in calcium or potassium can result in the wrinkling of leaves. Additionally, irregular irrigation and lighting are significant contributing factors to pepper leaf curl disease. This condition is primarily induced by aphids, mites, and whiteflies. Figure 4 below shows an image of mite pepper leaf disease.

Figure 4. Mite Pepper Leaf

1.1.1.2. Burnt

Yellowing and brown spots on pepper leaves are commonly observed symptoms associated with several diseases affecting pepper plants. These diseases can result in significant deformations in both the leaves and fruits, adversely impacting plant health and yield. Furthermore, deficiencies in essential nutrients, such as potassium or magnesium, can contribute to similar symptoms, highlighting the importance of accurate diagnosis and effective management strategies. Figure 5 is an image of the burnt-pepper leaf disease class.

Figure 5. Burnt Pepper Leaf

1.1.1.3. Mildew

Yellowing near the veins of pepper leaves is often associated with deficiencies in essential nutrients such as potassium or magnesium. Furthermore, infections caused by the mosaic virus frequently result in a mottled appearance and discolouration on the leaves. This condition is primarily attributed to fungal infections. Infected fruits typically exhibit discolouration in shades of light grey or white, accompanied by silvery spots on their surfaces. Grey spots can also be observed on the underside of the pepper leaves. Upon cutting infected plants, discolouration of internal tissues may become apparent; however, the leaves remain unaffected and retain their green colouration. Figure 6 below is an image belonging to the mildew-pepper leaf disease class.

Figure 6. Mildew Pepper Leaf

1.1.1.4. Worm

This disease is one of the most devastating threats to pepper plants. The most distinctive symptom is pallor, with wilting initiating on the leaves and progressing to the surface of the fruits. Another characteristic symptom is the appearance of dark white lines on the fruit surface. The damage observed on the pepper leaf is typically caused by a pest known as the "leaf miner" (*Liriomyza spp.*). The larvae of this pest feed within the leaf tissue, forming distinctive curved galleries (tunnels). Additionally, discolouration and spots are evident on various parts of the leaf, further indicating the presence of this pest. Figure 7 below is an image of the worm pepper leaf disease class.

Figure 7. Worm Pepper Leaf

1.1.1.5. Aphid

The disease manifests as water blister-like symptoms on pepper plants' leaves and fruit surfaces. Numerous lesions are observed on leaves and fruits, which are covered with wet, salmon-colored fluid spores. This condition is commonly associated with pests and pathogens such as red spider mites (*Tetranychus spp.*) or powdery mildew. The presence of dark spots on the leaves and general wilting are indicative of these pests or fungal infections. If not diagnosed early and treated appropriately, the disease poses a significant threat to plant growth and can severely impact pepper production. [6]. Figure 8 below is an image of the zinc pepper leaf disease class.

Figure 8. Aphid Pepper Leaf

1.1.1.6. Caterpillar

Upon examining the appearance of the leaf, it is evident that holes and irregular damage are present on its surface. This type of damage is typically associated with leaf pests or insect infestations, such as leafworms or caterpillars. Such injuries are caused by the feeding activity of leaf-eating pests, which consume portions of the leaf tissue. Figure 9 below is an image of the caterpillar pepper leaf disease class.

Figure 9. Caterpillar Pepper Leaf

1.2. Artificial Intelligence and Image Classification

Artificial intelligence (AI) refers broadly to human-like behaviors exhibited by a machine or system. AI is not merely a technological advancement but also a multidisciplinary field that transforms human life and enables innovation across various domains [7]. At its most basic level, AI involves programming computers to "mimic" human behavior utilizing extensive datasets derived from past examples of similar behaviors. For example, AI systems can be trained to distinguish a cat from a bird using thousands of images or to coordinate robotic arms with precision in a manufacturing facility. This process encompasses data analysis, pattern recognition, and the automation of decision-making.

More specifically, AI refers to the ability of a machine to learn from raw data, extract meaningful insights to generate understanding and utilize the acquired knowledge to achieve specific objectives. For example, a medical AI system can detect cancerous cells in imaging data, or an e-commerce platform can analyze user behavior to provide personalized product recommendations. The learning capability of AI continues to evolve, depending on the volume and quality of data. AI spans various fields, including image processing, speech recognition, natural language processing, decision-making, and robotics, offering solutions that replicate or exceed human cognitive abilities [8].

The impact of AI in some sectors extends beyond technological innovation. In healthcare, AI accelerates disease diagnosis and drug development, saving lives. In agriculture, AI is applied to enhance crop yield and optimize resource usage through applications ranging from weather prediction to pest detection. In education, AI provides personalized learning experiences by tailoring programs to students’ abilities. Meanwhile, the financial sector utilizes AI for risk analysis, fraud detection, and automated trading systems [9].

The future potential of AI is vast. As human-machine interactions deepen, AI could revolutionize smart cities, environmental sustainability, and energy management. Additionally, ethical considerations, security concerns, and the implications of AI on the workforce are becoming increasingly prominent topics of discussion. While AI simplifies daily life on the one hand, it continues to contribute to solving complex global challenges on the other. In summary, artificial intelligence is poised to remain at the forefront of technological advancement, shaping both the present and the future world, As Figure 10 shows, AI encompasses subfields such as machine learning (ML) and deep learning (DL). ML allows systems to learn directly from data without explicit programming, while DL employs multi-layered artificial neural networks to interpret complex data structures.

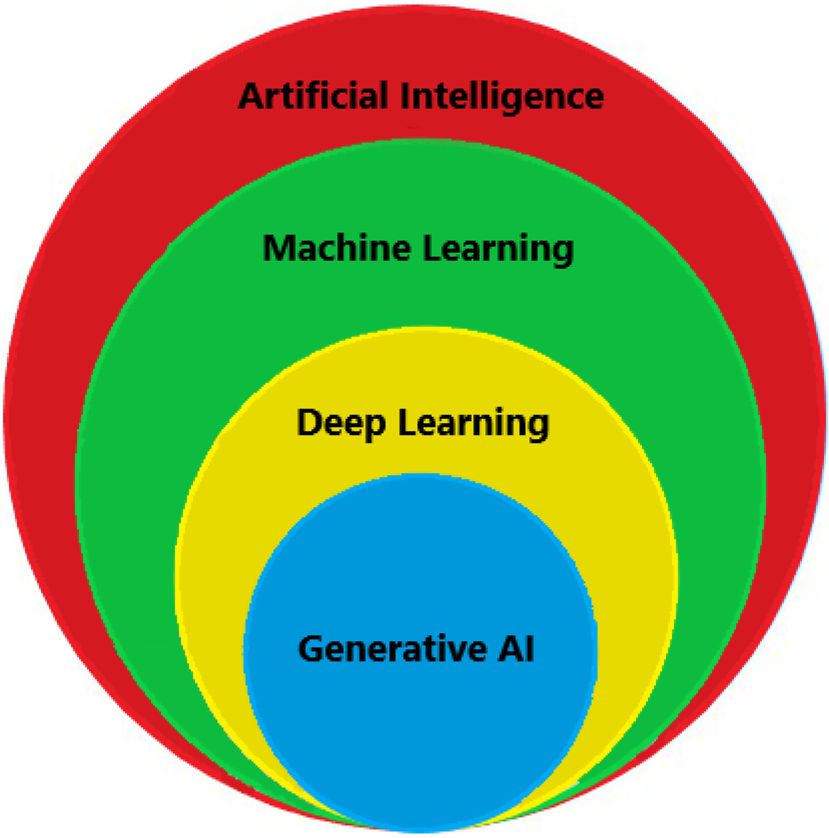


Figure 10. Artificial Intelligence

AI algorithms are widely used in image processing, especially in image analysis, because of their powerful simultaneous identification and recognition properties, according to multiple criteria. The most commonly used AI models in this field are the Ml and DL models. Subsections 2.1.1 and 2.1.2 explain these in more detail.

1.2.1. ML-based Algorithms

ML is a subfield of artificial intelligence that focuses on enabling computers to learn from data and enhance their performance without human intervention. The learning process typically begins with experiences or observations, to identify patterns within the data and make more informed decisions in future scenarios. The primary objective is to develop systems capable of autonomous learning and adaptation, leveraging acquired information to optimize actions and eliminate the need for manual oversight. During this process, systems analyze data to extract meaningful patterns and relationships, utilizing these insights to generate accurate predictions or make data-driven decisions about future events [10]. The most prominent ML-based algorithms used in image processing are Decision Tree, LightGB, Logistic Regression, KNN, SVM, XGBoost, Naive Bayes Classification, and Random Forest. They are explained in more detail below.

1.2.1.1. KNN

K-Nearest Neighbor (KNN) is a machine learning algorithm frequently employed for efficient solutions in small datasets. This algorithm identifies existing instances and classifies new ones based on a similarity measure. KNN was introduced in the early 1970s and has since been widely applied in statistical prediction and pattern recognition. Its simplicity and effectiveness make it a popular choice for various classification and regression tasks, particularly in scenarios where computational efficiency and interpretability are prioritized. Figure 11 shows the algorithm for two different classes in K-Nearest Neighbor.

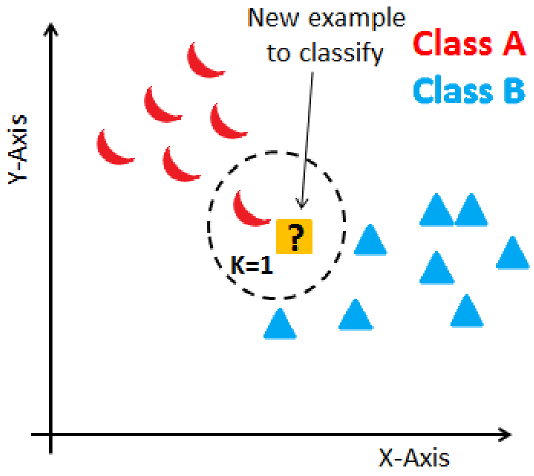


Figure 11. KNN Algorithm

1.2.1.2. SVM

Support Vector Machine (SVM) is a supervised machine learning technique employed for both classification and regression tasks, with a predominant application in classification problems. In this approach, each data point is represented as a point in a high-dimensional space, where the value of each feature corresponds to the value of a specific coordinate. SVM aims to identify a hyperplane that optimally separates the data points of different classes, maximizing the margin between them. This method has achieved high accuracy, particularly in small to medium-sized datasets, making it a powerful tool in various ML applications [11]. Figure 12 illustrates two different classes schematized for the Support Vector Machine algorithm.

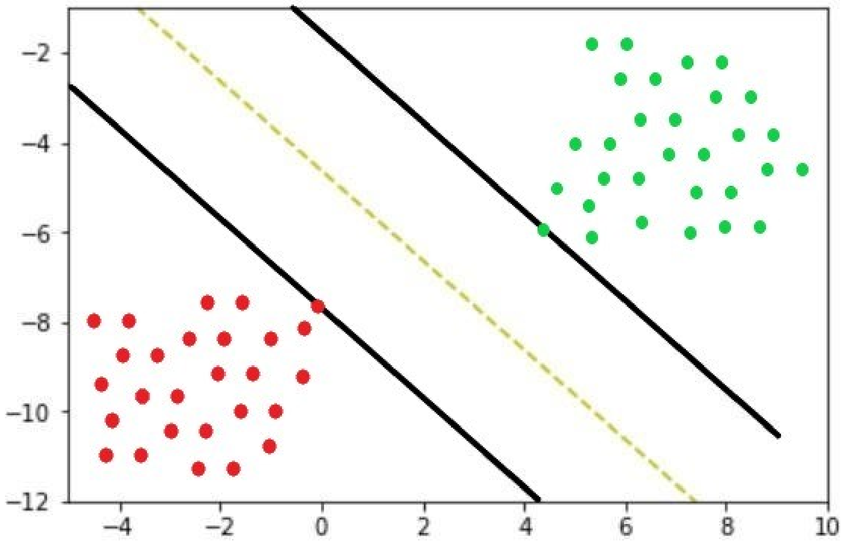


Figure 12. SVM Algorithm

1.2.1.3. Decision Tree

A decision tree is a supervised learning algorithm frequently used in classification studies. It works for continuous input and output variables. This algorithm divides the population into two or more homogeneous clusters based on the most important feature in the input variables. Figure 13 schematizes the predictions of diseased and healthy classes for two pepper disease classes in the decision tree algorithm, followed by sub-predictions.

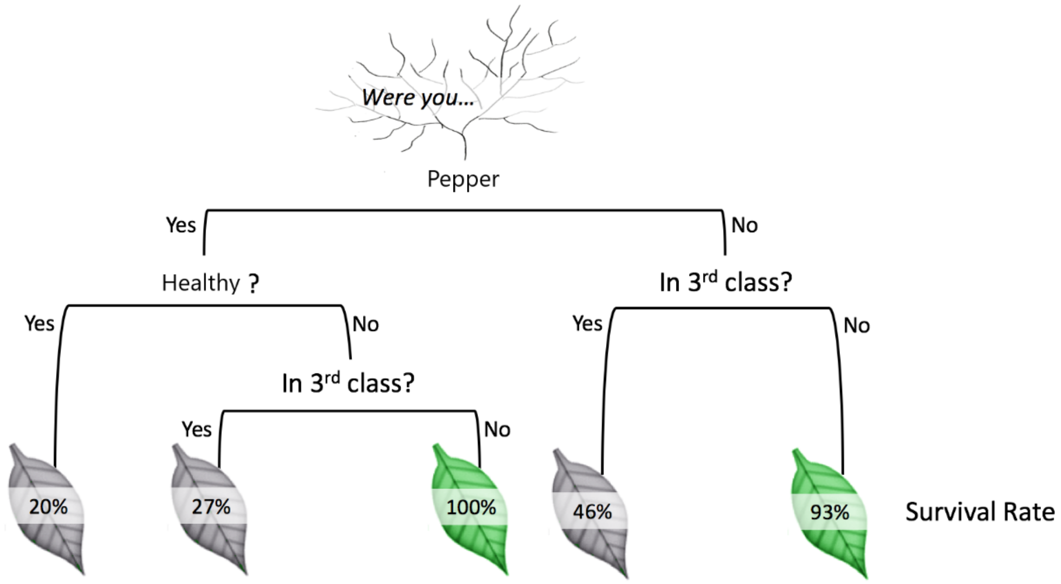


Figure 13. Decision Tree Algorithm

1.2.1.4. Naïve Bayes Classification

Naive Bayes is a highly effective and widely utilized machine learning classifier, grounded in Bayes' theorem. The term "naive" refers to the assumption that all features are independent of each other, an assumption that simplifies calculations and model development. Despite this simplification, Naive Bayes has demonstrated remarkable performance in various applications. It operates by applying the maximum a posteriori decision rule within a Bayesian framework, making it a probabilistic classification method. Furthermore, it can be interpreted as an implementation of a Bayesian network. Naive Bayes classifiers are particularly prominent in text classification tasks, such as spam detection and sentiment analysis, due to their simplicity, efficiency, and effectiveness in handling large-scale data.

$$\boldsymbol{P(c/x) = \frac{P(x/c)P(c)}{P(x)}} \tag{1}$$

- (c|x): The probability of c given that x has occurred.
- (x|c): The probability of x given that c has occurred.
- *P*(c): The prior probability of c.
- *P*(x): The prior probability of x.

1.2.1.5. Random Forest

Random Forest is a robust ensemble learning method widely used in machine learning, where multiple weak models, specifically decision trees, are combined to form a strong predictive model. This technique builds an ensemble of decision trees and aggregates their outputs to achieve higher accuracy and reliability in predictions. Recognized as a versatile solution to various data science problems, Random Forest can address regression and classification tasks effectively. Furthermore, it excels in handling missing values, outliers, and other critical aspects of data preprocessing. It also demonstrates high efficiency in dimensionality reduction, making it a valuable tool for tackling complex datasets with impressive performance outcomes. Figure 14 below schematizes the predictions of diseased and healthy classes for two pepper disease classes in the random forest algorithm, followed by sub-predictions.

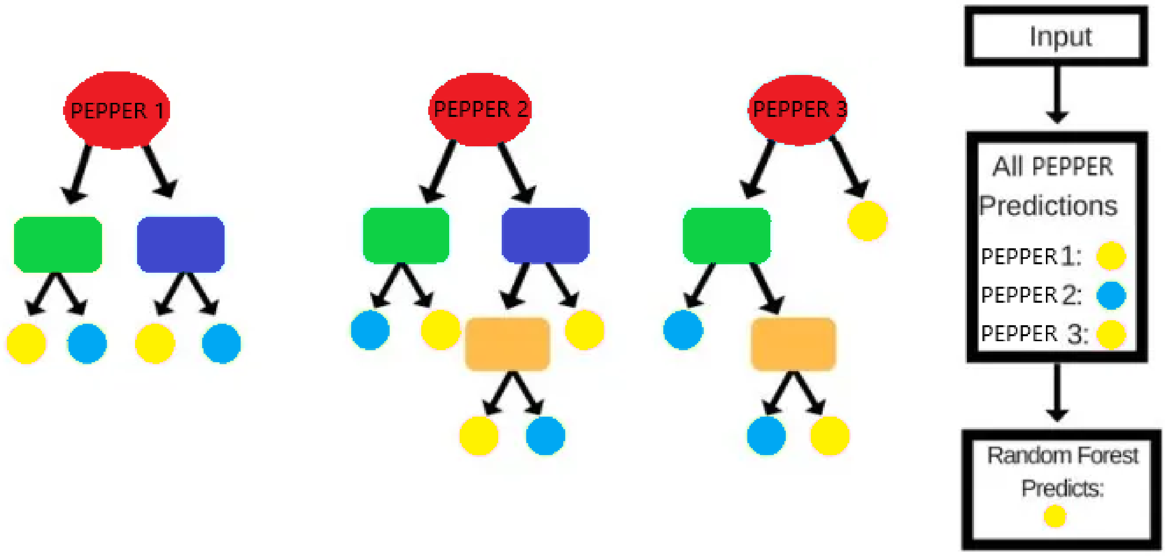


Figure 14. Random Forest Algorithm

1.2.2. Image Classification and Deep Learning

In the literature, various approaches have been proposed, such as the automated scanning of vegetables, fruits, flowers, and leaves. In recent years, image processing methods and machine learning algorithms have been increasingly utilized for the rapid detection of plant diseases [12]. Deep learning, a technique that enables the representation of multilayered data abstractions, consists of multiple processing layers. Notably, it does not explicitly define which features are used or how they are extracted. Instead, deep learning can automatically extract features directly from the raw input data [13]. According to studies, deep learning methods have effectively identified and classified diseases, despite various external factors such as lighting, visibility, background, size, and orientation. Convolutional Neural Network (CNN)-based deep learning (DL) algorithms, renowned for their exceptional performance in object detection and classification, are widely employed across various medical fields, including radiology.

The structure of this paper is organized as follows: The second section provides a concise review of previous research regarding the datasets, algorithms, and methodologies utilized in the detection of leaf diseases. The third section outlines the methods employed in this study, while the fourth section summarizes the findings. Lastly, the fifth section presents the discussion and conclusion.

Convolutional Neural Networks (CNNs) architecture consists of multilayer perceptrons, distinguishing them from traditional artificial neural networks. In conventional neural networks, the input is transformed through a series of hidden layers, with each layer fully dependent on the previous one. In contrast, a Convolutional Neural Network (CNN) [14] is a deep learning algorithm composed of sequential convolutional, pooling, and fully connected layers. These three types of layers have distinct characteristics, with the output of each preceding layer serving as the input to the subsequent layer. This structure has proven highly effective in achieving success in plant disease classification. Advances in CNN-based architectures have notably enhanced the efficiency of plant disease recognition.

CNNs are widely used in image recognition tasks, including the detection of diseases and the identification of disease features. A large volume of images or data is used to train the CNN model, leading to improved training outcomes and classification performance [15]. CNNs have demonstrated remarkable performance in image classification and various other visual processing applications, such as the diagnosis of lung diseases, cancer imaging, object detection, visual parsing, and other classification tasks.

Prominent CNN models include:

- AlexNet,
- ResNet,
- GoogleNet,

- MobileNet,
- DenseNet,
- VGGNet

In this study, the input layers of the CNN model are trained using images from the dataset specific to this research.

1.2.2.1. AlexNet

AlexNet is a Convolutional Neural Network (CNN) architecture that represents a significant milestone in the fields of deep learning and computer vision. Developed in 2012 by Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton, AlexNet highlighted the transformative potential of deep learning by achieving exceptional performance in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [16].

The architecture of AlexNet comprises **eight layers**, including **five convolutional layers** and **three fully connected layers**. Max pooling is applied after the convolutional layers to reduce spatial dimensions and improve computational efficiency. A notable feature of AlexNet is the use of the **ReLU (Rectified Linear Unit)** activation function after each convolutional layer. The nonlinear nature of ReLU accelerates the learning process and facilitates the training of deeper networks by mitigating issues such as the vanishing gradient problem. To address overfitting, AlexNet employs the **dropout** regularization technique in the fully connected layers, which enhances the model's generalization capabilities by randomly deactivating a subset of neurons during training. The network is designed to process **RGB images** with a resolution of **224x224 pixels**, accepting input data in three channels to accommodate color images [17].

AlexNet's success demonstrated the robustness of deep learning techniques, achieving high accuracy rates on large-scale datasets and excelling in tasks such as image classification. This accomplishment inspired subsequent research and development of more advanced architectures, including **VGG**, **ResNet**, and **Inception**, which expanded upon AlexNet's foundational principles. In summary, AlexNet played a pivotal role in establishing deep learning as a dominant paradigm in computer vision and continues to serve as a cornerstone for modern approaches in the field.

1.2.2.2. ResNet

Residual Networks (ResNets) represent a Convolutional Neural Network (CNN) architecture specifically designed to address challenges encountered in the training of deep neural networks. Developed in 2015 by Kaiming He and his team at Microsoft Research, ResNet is particularly effective in mitigating the vanishing gradient problem commonly observed in very deep networks, thereby facilitating their training [18].

The primary innovation of ResNet lies in its **residual learning mechanism**. Unlike traditional neural networks, where each layer processes input data directly for the next layer, ResNet introduces a mechanism where the output of a layer is combined with a copy of its input through skip connections. This approach allows the network to focus on learning residual mappings rather than complex transformations, thus enabling the effective training of extremely deep networks without performance degradation.

The residual learning function is mathematically represented as:

$$\mathbf{H(x) = F(x) + x} \qquad \mathbf{(2)}$$

- **H(x):** The desired output of the network.
- **F(x):** The residual function representing the learned transformation.
- **x:** The input data.

- **Convolutional Layers:** Extract key features from input data.
- **Batch Normalization:** Enhances stability and accelerates training.
- **Activation Function:** Utilizes the Rectified Linear Unit (ReLU) for non-linearity.
- **Skip Connections:** Bypass the intermediate transformation by directly passing input to subsequent layers.

ResNet models: ResNet18, ResNet34, ResNet50

ResNet demonstrated its exceptional performance in the 2015 ImageNet competition, securing first place with a 152-layer model. This achievement established that very deep architectures could be effectively employed in deep learning tasks. Furthermore, ResNet models have gained prominence for their lower error rates and higher accuracy in comparison to conventional architectures [19].

1.2.2.3. GoogleNet

GoogLeNet, developed by Google in 2014, represents a significant advancement in deep learning architecture and constitutes the inaugural version of the Inception family. Specifically engineered to enhance computational efficiency in large, deep networks, GoogLeNet garnered significant attention upon winning the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) in 2014. This architectural innovation not only optimized model performance but also effectively reduced computational costs.

The central innovation of GoogLeNet lies in adopting Inception modules, which incorporate parallel convolution filters of varying sizes (1x1, 3x3, 5x5) alongside pooling layers. This design facilitates extracting more complex, diverse features, thereby enabling deeper networks without disproportionately increasing the parameters. Consequently, GoogLeNet achieves a more efficient network structure. Despite comprising only 22 layers, GoogLeNet attains remarkable accuracy, demonstrating superior performance to traditional networks with a significantly lower parameter count [20].

Furthermore, the incorporation of 1x1 convolutional layers in GoogLeNet reduces computational demands while simultaneously augmenting the network's depth and representational capacity. This architecture has become a pivotal reference point for subsequent advancements in deep learning, laying the groundwork for the evolution of models such as Inception V2 and Inception V3.

1.2.2.4. MobileNet

MobileNet is a deep learning architecture specifically designed for environments with limited resources, such as mobile devices and embedded systems. Developed by Google, the architecture prioritizes computational efficiency by minimizing model size and reducing computational costs.

The primary innovation of MobileNet lies in the use of **depthwise separable convolutions**. Unlike traditional convolutions, which apply filters across all input channels simultaneously, depthwise separable convolutions apply separate filters for each channel, significantly reducing the number of computational operations. This reduction leads to a smaller model size, faster processing, and enhanced portability, making MobileNet particularly well-suited for mobile applications [21].

MobileNet is widely utilized in a range of tasks, including image classification, object detection, and segmentation, and is optimized for deployment on mobile devices. The architecture's subsequent iteration, MobileNetV2, introduces further improvements, enhancing both efficiency and performance. MobileNet has made a significant contribution by enabling the deployment of deep learning models on low-cost, resource-constrained devices

1.2.2.5. DenseNet

DenseNet (Densely Connected Convolutional Networks) represents an innovative deep learning architecture that introduces a novel connection pattern within Convolutional Neural Networks (CNNs). Initially proposed by Gao Huang, Zhuang Liu, and Kilian Q. Weinberger in 2017, DenseNet’s primary innovation lies in its dense connectivity scheme. In conventional CNN architectures, each layer receives input exclusively from the preceding layer. In contrast, DenseNet establishes a connection where each layer receives input from all prior layers, thereby facilitating a feed-forward structure where all layers are interconnected. This arrangement ensures a more efficient flow of information throughout the network.

The dense connections in DenseNet contribute to improved gradient propagation, mitigating the vanishing gradient problem, and promoting the reuse of features. Consequently, the architecture achieves greater efficiency, with each layer benefiting from the features of preceding layers. This feature reuse enables DenseNet to operate with fewer parameters while maintaining high performance, making it particularly effective for deeper networks.

DenseNet variants: DenseNet-121, DenseNet-169, DenseNet-201, and DenseNet-264.

In conclusion, DenseNet’s distinctive approach to connectivity and feature reuse substantially enhances the training process and performance of deep neural networks, providing an efficient and robust architectural framework that reduces the number of parameters needed while enhancing model efficacy.

1.2.2.6. VGGNet

VGGNet is a deep learning architecture developed in 2014 by the Visual Geometry Group (VGG) at the University of Oxford, which has achieved significant success, particularly in the field of visual recognition [30]. The architecture is primarily recognized by its two main versions: VGG-16 and VGG-19, with these names referring to the number of layers (16 and 19 layers, respectively).

The primary innovation of VGGNet lies in its ability to achieve success through deeper and more complex structures, with the network's layers being homogeneously structured. This design utilizes 3x3 convolutional filters in each layer, enabling the extraction of deeper features. The purpose of employing smaller filters, as opposed to larger ones, is to facilitate learning more abstract features by passing through more layers, thereby efficiently increasing the number of parameters.

Typically, VGGNet is presented in two versions: VGG-16, which has 16 layers, and VGG-19, which contains 19 layers. These layers consist of convolutional, pooling, and fully connected layers. VGGNet exclusively employs 3x3 convolutional filters and 2x2 max-pooling layers, balancing the depth of the network while controlling the number of parameters [22].

VGGNet is commonly utilized as a pre-trained model, having been trained on large datasets such as ImageNet. It has demonstrated efficacy in various visual recognition tasks. The architecture's success has not only illustrated the efficiency of deploying deeper networks in deep learning but has also inspired the development of subsequent modern architectures, such as ResNet and Inception.

2. LITERATURE REVIEW

This literature review focuses on prior research that employs deep learning (DL) algorithms, such as AlexNet, DenseNet, VGGNet, MobileNet, and ResNet, for the detection of plant diseases in recent years. These studies have made significant contributions to the development of the field, and several of them have been comprehensively summarized below due to their methodological or thematic similarities.

First proposed in 1998 by LeCun et al. for image recognition tasks, deep learning has significantly advanced and diversified over the years [23]. These advancements have been made possible by improved computational power, algorithmic designs, and access to large-scale labeled datasets. However, applying DL to agricultural tasks initially faced considerable challenges, the most significant of which was the lack of sufficient and reliable datasets specific to this domain.

Until recently, DL applications for plant disease diagnosis were primarily limited and addressed only through sporadic practical implementations [24]. However, the past decade has witnessed a paradigm shift in this field. Researchers have increasingly focused on enhancing the accuracy of plant disease detection by leveraging DL techniques and employing diverse datasets across a wide range of plant species. The utilization of various algorithms and the application of extensive datasets have improved classification accuracy and paved the way for innovative approaches to solving agricultural challenges. These advancements highlight the critical role of DL in overcoming longstanding obstacles in plant pathology.

One significant milestone in the field was the development of the **PlantVillage** database, which contains 38 classes representing 14 plant species and remains one of the most comprehensive datasets for plant disease detection. This database has served as a critical resource, accelerating the application of deep learning algorithms in agricultural technologies. Sardogan et al. employed a CNN model using the PlantVillage dataset to identify tomato leaf diseases [25]. To optimize the classification process, the researchers integrated the Learning Vector Quantization (LVQ) algorithm, achieving an accuracy rate of 88%. This result demonstrated the effectiveness of CNN-based classification techniques in detecting plant leaf diseases. The study conducted classification into a total of five categories, including four disease classes and one healthy class.

In the same year, Rangarajan and his team evaluated pre-trained AlexNet and VGGNet models to classify tomato leaf images from the same dataset into seven distinct classes [26]. Their experimental findings revealed a remarkable accuracy rate of 97.4%, primarily attributed to the superior performance of the AlexNet architecture. Despite methodological similarities with other studies, this research achieved significantly higher accuracy compared to prior findings. Another notable study focusing on VGGNet was conducted by Ferentinos et al., who analyzed and compared the performance of various architectures trained on an extensive dataset of approximately 87,000 images of healthy and diseased plant leaves across 25 plant species, including peppers [27]. Among the architectures evaluated, VGGNet achieved an impressive accuracy rate of 99.53%, outperforming all other models on this dataset. This study further underscored the remarkable efficiency of VGGNet as a deep learning-based tool for plant disease diagnosis.

The following year, Kaya et al. introduced a combined methodology utilizing VGGNet and AlexNet, which was evaluated across four distinct databases, including the PlantVillage dataset [28]. The study classified various plant species, including peppers, as either healthy or diseased. Rather than providing plant-specific accuracy rates, the authors reported the accuracy rates in a more generalized manner for each individual dataset. This approach reflects the broader applicability of the proposed model. Notably, the best score achieved for binary classification on the PlantVillage dataset was 99.80%, underscoring the high efficacy of the methodology in distinguishing between healthy and diseased plant leaves, with potential implications for early disease detection in agricultural practices.

Sardogan et al. utilized a Convolutional Neural Network (CNN) model to identify tomato leaf diseases using the PlantVillage dataset [29]. To further enhance the classification process, they implemented the Learning Vector Quantization (LVQ) algorithm, achieving an accuracy rate of 88%. This demonstrates the effectiveness of CNN-based classification techniques in detecting and categorizing leaf diseases. The study classified the data into a total of five categories, including four disease classes and one healthy class. Notably, their work highlighted the potential of integrating traditional algorithms like LVQ with CNN models to improve classification performance, particularly in datasets with moderate complexity.

Gensheng and his team [30] applied a trained CIFAR10 fast CNN model for the identification of tea leaf diseases. The experimental results showed an identification accuracy of 92.5%, which underscores the applicability of lightweight deep learning models in agricultural diagnostics. Their approach also emphasized the importance of efficient feature extraction in achieving reliable results with minimal computational costs.

Militante and his team [31] proposed a CNN for the detection of plant leaf diseases. The study was completed in three stages: image acquisition, image preprocessing, and classification. Using just a simple CNN model, they achieved an accuracy rate of 96.5%. Their findings demonstrated that even straightforward CNN architectures can yield high classification accuracy when properly optimized for specific agricultural datasets, suggesting a viable solution for scalable and practical applications in precision agriculture.

In a similar vein, Saradhambal and his team proposed an alternative classification method aimed at improving diagnostic efficiency for early plant leaf disease detection [32]. The dataset employed in their study comprised five categories: one healthy class and four disease classes. However, it is important to note that the dataset was constrained by a relatively small sample size, with only 75 distinct plant leaf images included. This limited sample size posed a challenge to the generalizability of the findings, thereby affecting the robustness of the model's performance. Despite applying a range of detection techniques, the researchers faced several limitations. Specifically, the restricted feature set within the dataset resulted in suboptimal training rates for the algorithm, and the absence of detailed methodological implementation further hindered the potential for future improvements. These limitations highlight the critical need for dataset expansion to facilitate the generation of more accurate and reliable results in plant disease detection.

Similarly, K. Elangovan and his team observed limited success in their plant disease classification efforts using Support Vector Machine (SVM) techniques [33]. They attempted to mitigate noise issues by transforming images into an alternative color space and employing image preprocessing methods, such as grayscale conversion. However, their approach also faced challenges, resulting in comparatively low accuracy rates. This emphasizes the importance of using larger and more diverse datasets to improve the accuracy and applicability of classification algorithms, particularly in the context of plant disease detection. The study indicates that advancements in data quality and preprocessing techniques are crucial for enhancing model performance and reliability.

In the past decade, significant advancements in deep learning (DL) have contributed to substantial progress in classification tasks, with the field of pepper leaf disease recognition being no exception. Initially focused on basic binary classification, the field has since transitioned to more complex multi-class classification tasks, facilitating more comprehensive and precise diagnostic approaches. In recent studies, models capable of performing five-class classification, distinguishing between healthy leaves and four distinct disease categories, have been developed and reported [34]. These advancements have enhanced the potential for early detection and monitoring of plant diseases in agricultural practices. A thorough review of the literature reveals that, among the various algorithms employed, the VGGNet image processing algorithm stands out for its exceptional effectiveness in handling large and complex datasets, consistently achieving superior accuracy rates. The success of VGGNet in diagnosing a wide range of plant diseases has established it as a leading benchmark in this area of research. Building on this, the present study aims to assess the performance of the VGG16 model using a dataset comprising seven distinct classes. The primary objective is to optimize model performance and reduce processing time by leveraging the pre-trained VGG16 architecture, thus contributing to more efficient and accurate early detection of plant diseases.

3. METHODOLOGY

This section provides a detailed explanation of the training and testing processes of the proposed CNN model, supported by the schematic diagram presented in Figure 15. The raw dataset, consisting of 1,600 original digital images, underwent a comprehensive preprocessing pipeline prior to model training. This preprocessing aimed to enhance image quality, improve overall model performance, and optimize the training duration. The training process was conducted using a pre-trained VGG16 architecture, where the convolutional and max-pooling layers were frozen to preserve the pre-learned features. Additionally, dense and dropout layers were incorporated into the model to accelerate the training process and achieve better generalization performance. Further details regarding the dataset characteristics, preprocessing stages, and classification procedures are elaborated below.

The classifier was developed to identify and categorize pepper leaves. In this process, the photos were divided into two main groups after the preprocessing stage: training images and test images. Training images were used to train the model based on characteristic features to classify diseased or unhealthy leaves. Test images, on the other hand, were utilized to evaluate the accuracy and generalization capability of the model. These images were processed through an enhanced CNN-based characterization framework to determine whether the leaves were contaminated or healthy.

The methodological steps followed in the implementation of this system can be summarized as follows:

- **Image acquisition:** Collecting digital photographs of pepper leaves and storing them in a processable format.

- **Preprocessing:** Enhancing the quality of the images and preparing them for the model through operations such as filtering, resizing, and color adjustments.

- **Feature extraction:** Analyzing the images to extract meaningful features that can be used for modeling.

- **Classification:** Processing the extracted features with a CNN-based model to determine whether the leaf is healthy or contaminated.

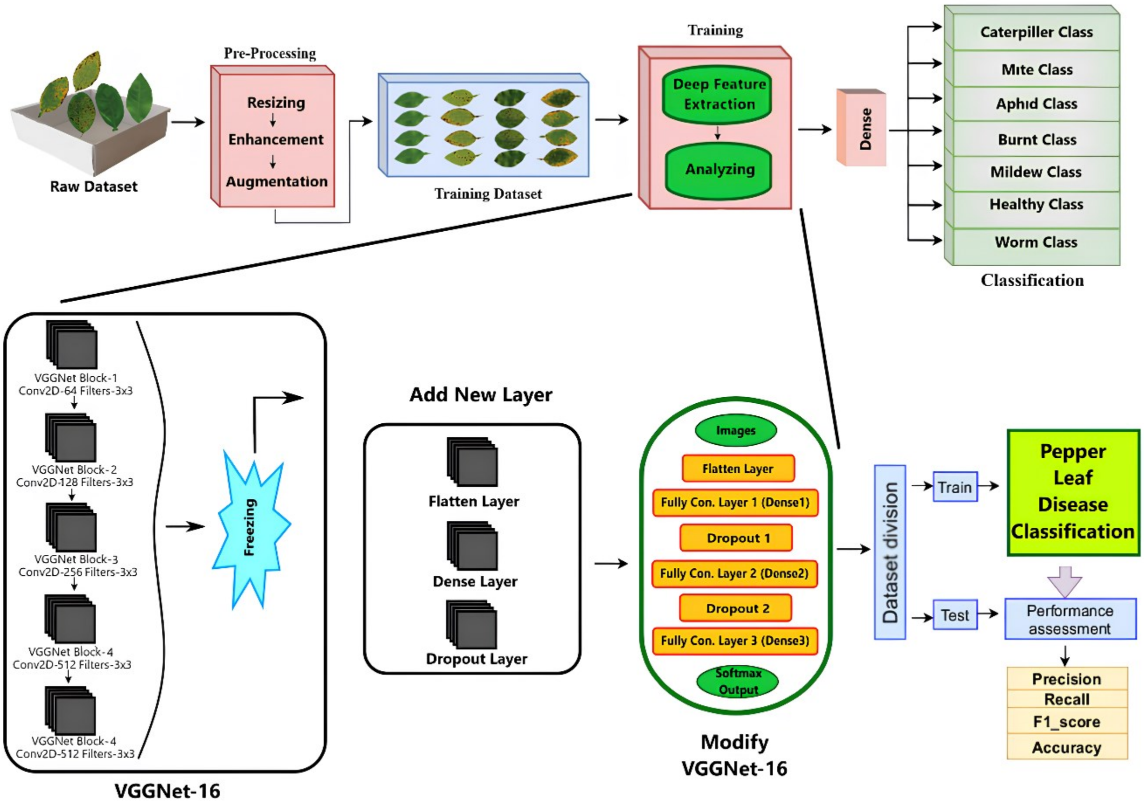


Figure 15.A Block Diagram of Proposed Model

Subsequently, the Deep Neural Network (DNN) is tested and implemented within the training model [35]. The validation process is carried out by running the model with the test dataset. If the obtained results do not correspond to the expected output, which should represent a specific feature of the leaf, the process is restarted from the DNN to ensure the accuracy of the classification. This iterative process continues until the model outputs the correct classification of the pepper leaf type. This approach is employed to enhance classification accuracy and ensure the model achieves the desired performance level.

3.1. DataSet

In this study, a custom dataset was carefully compiled, consisting of a total of 1,679 leaf images captured using an iPhone digital camera with original dimensions of 1200x1600 pixels. To ensure high consistency in visual analysis and accurate classification, each image was isolated against a uniform background. This preparation minimizes distractions and enhances the visibility of leaf diseases, facilitating the model's ability to identify key features. The dataset is divided into two primary groups: 300 healthy leaves and 1,379 infected leaves. The infected leaves are categorized into six distinct disease groups: 123 aphid-infected leaves, 300 burnt leaves, 145 caterpillar-infested leaves, 300 mildew-infected leaves, 211 mite-infested leaves, and 300 leafworm-infected leaves. Each of these groups represents a specific plant disease, with the diversity in the dataset ensuring that the model is trained to recognize a broad range of leaf conditions.

The dataset is structured into seven labeled classes: one for healthy leaves and six for various types of leaf diseases. This classification is fundamental to the model's training, as it enables the network to learn the distinguishing features of each class. For the training process, 70% of the images are used, while the remaining 30% is allocated for testing and validation in a 2:1 ratio. This partitioning of the data is critical for assessing the model's generalization capabilities. The testing and validation phases are designed to evaluate how well the model performs on unseen data, which is crucial for ensuring that it can make accurate predictions on new, real-world examples. By structuring the dataset in this manner, the study aims to develop a model that not only classifies leaf diseases accurately but also demonstrates robust performance across a variety of conditions.

3.2. Pre-Processing Strategy

Data pre-processing is a critical set of steps aimed at improving the quality of images in the raw dataset to enable more efficient analysis. This process enhances the performance of image recognition and facilitates better generalization across different variations of images. Specifically, in deep learning models, pre-processing operations are essential for enabling the model to learn faster and more effectively, thereby improving the accuracy of the results derived from visual data. The pre-processing model proposed in this study consists of four main steps: resizing, cropping, color enhancement, and data augmentation. Each of these steps plays a crucial role in preparing the images for modeling, ensuring they are appropriately formatted and optimized for training. Further details regarding the implementation and importance of each pre-processing step are provided below.

3.3. Cropping and Resizing

The leaf images utilized in this study were initially captured at a resolution of 1200×1600 pixels, offering a high level of detail. However, to enhance the efficiency of analysis and model training, it was necessary to refine the images. The non-leaf regions of the images were manually cropped, ensuring that only the leaf itself remained, thus eliminating any irrelevant background that could potentially interfere with the model's learning process. Subsequently, the resolution of the images was reduced to 500×1000 pixels, preserving essential details while making the images more manageable for processing [36].

Despite these adjustments, the VGG16 model, chosen for this study, is designed to work most effectively with images of a standard dimension of 224×224 pixels. This image size is ideal for computational efficiency, as it allows for faster processing and reduces training time, ultimately improving the overall effectiveness of the model [37]. To ensure compatibility with the VGG16 architecture, all images were resized to 224×224 pixels using a program with the parameters Img Width and Img Height set to 224×224. This resizing process is vital for standardizing the input data and aligning it with the model's specifications. Figure 16 provides before-and-after examples of the cropping and resizing procedures, showcasing the transformation of the images and the adjustments made to optimize the dataset for model training.

Figure 16. Cropping and Resizing

3.4. Augmentation

Data augmentation is a set of techniques designed to artificially generate new data samples, thereby increasing the diversity of the training dataset to enhance model generalization [38]. This process plays a crucial role in addressing issues such as overfitting, which often arises from working with limited original data, and helps ensure that the model produces robust results across various conditions. In this study, data augmentation was carried out using eight different techniques, the details of which are outlined in Table 1. For each original image, nine new samples were created, increasing the total number of samples in the training dataset to 10,080. During this augmentation process, it is possible for empty pixels to appear around the area subjected to the transformation. These empty pixels were addressed using the 'fil mode=’nearest’ function, which filled the gaps by assigning values from neighboring pixels, thereby maintaining visual consistency. Following this, color enhancement was applied to emphasize disease features, utilizing the 'enhance colors' preprocessing function, which streamlined the training process by making the disease characteristics more pronounced. Figure 17 illustrates the various techniques employed during the data augmentation process, along with corresponding visual examples. These techniques significantly contributed to increasing the diversity of the data, ultimately enhancing the training performance of the model.

Technique	**Range**	**Task**
Random Rotation	40	Randomly rotate the images between 0 and 40.
Zoom	0.2	Randomly zooms the images in or out by 20%.
Width Shift	0.2	Shift the images horizontally by 20%.
Height Shift	0.2	Shift the images vertically by 20%.
Shear	0.2	Apply a 20% shear to the images.
Horizontal Flip	True	Randomly flips the images horizontally.
Brightness Range	[0.2,1.0]	Randomly adjusts image brightness between 20% and 100%.
Fill Mode	Nearest	Fills empty pixels with the nearest neighbor values.

Table 1. The Augmentation Techniques Performed in The Study

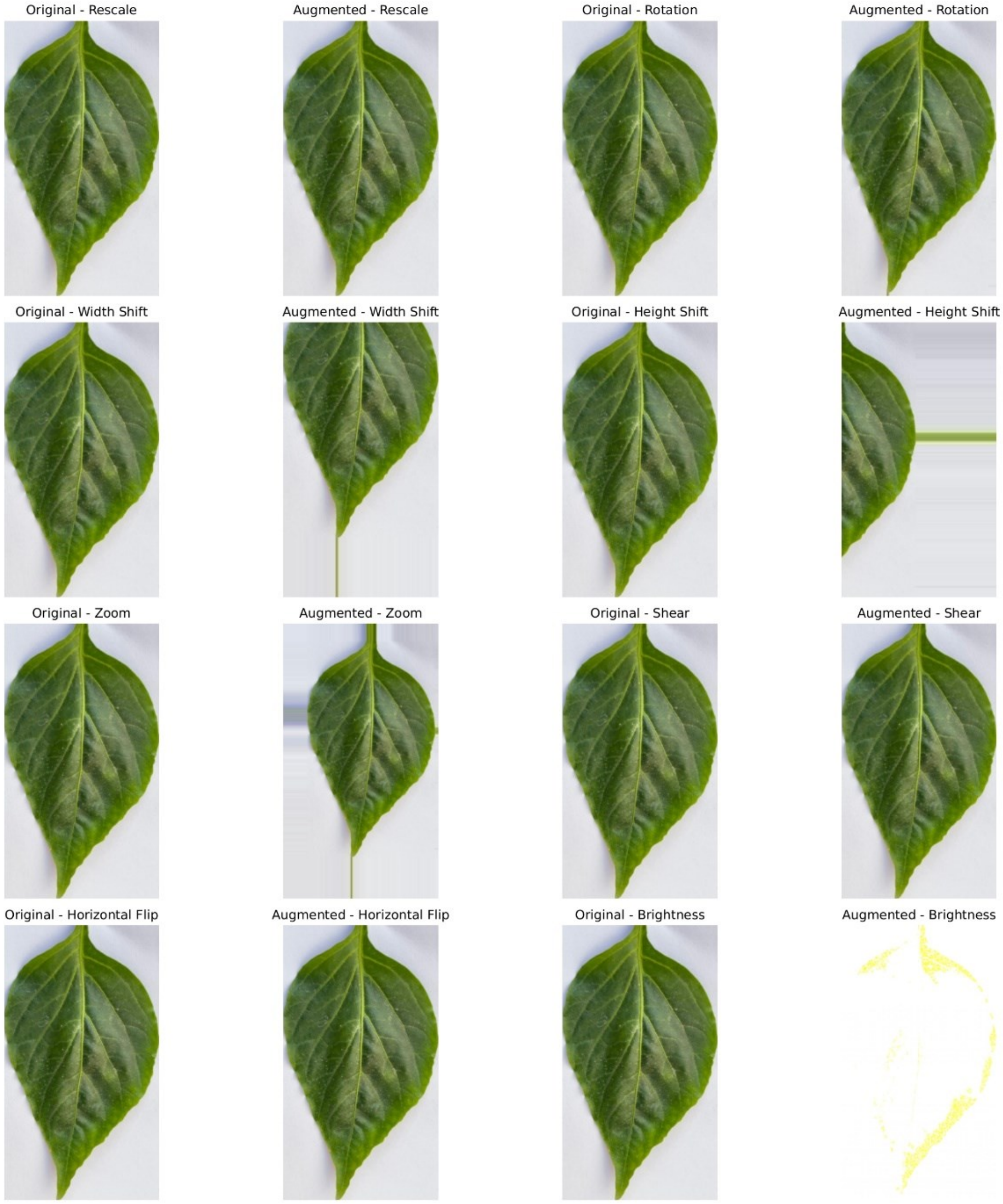


Figure 17. Data Augmentation

3.5. Color Enhancement

The algorithm, illustrated with the flowchart diagram in Figure 19, is specifically designed to enhance and intensify the green and yellow colors present in an image. This enhancement process aims not only to improve the visual vibrancy and contrast of the images but also to optimize their color properties for analytical purposes. By making the color tones more prominent, the algorithm facilitates the artificial intelligence model's ability to accurately recognize and differentiate between various color variations.

The operation of the algorithm begins with the conversion of the images from the RGB (Red, Green, Blue) color space to the HSV (Hue, Saturation, Value) color space. This transformation allows for more precise identification and processing of colors, as the HSV space provides a more intuitive representation of color attributes. Within the HSV color space, specific lower and upper boundary values are defined for the green and yellow tones, which are critical for accurately isolating the target colors within the images.

Subsequently, the cv2.inRange() function is employed to detect and mask the areas within the images that fall within the defined green and yellow color ranges. These masked regions are then subjected to additional processing to enhance the intensity and clarity of the detected colors. In the final step, the identified green and yellow areas are converted back to the RGB color space to preserve the natural appearance of the images while maintaining the enhancements. As a result, the algorithm effectively amplifies the saturation and prominence of the green and yellow tones, enabling the model to more accurately distinguish subtle color variations.

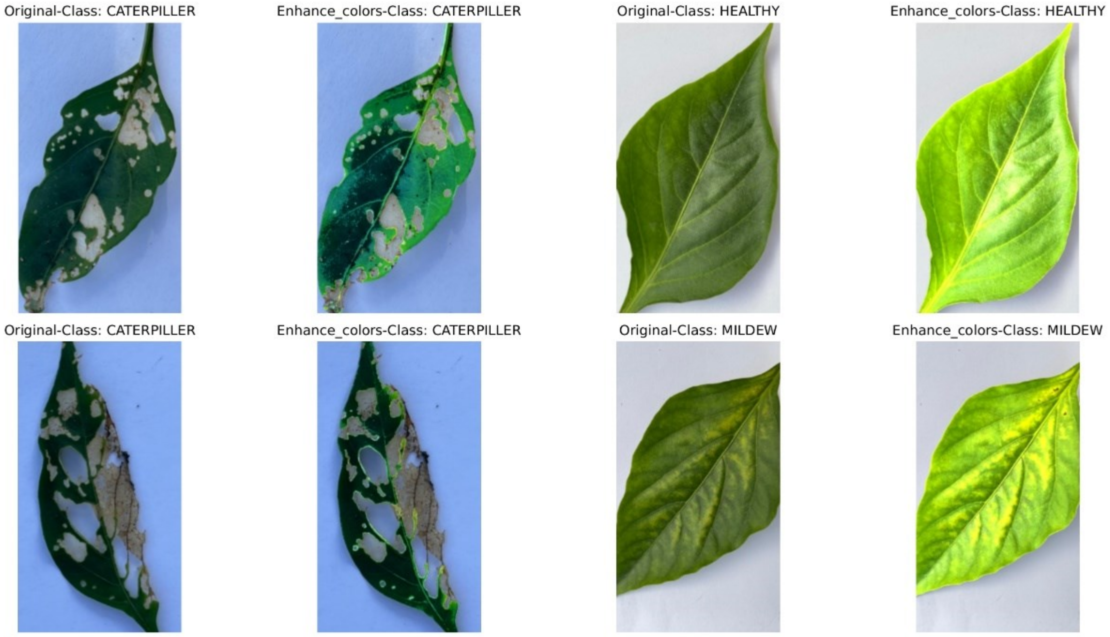


Figure 18. Color Enhancement

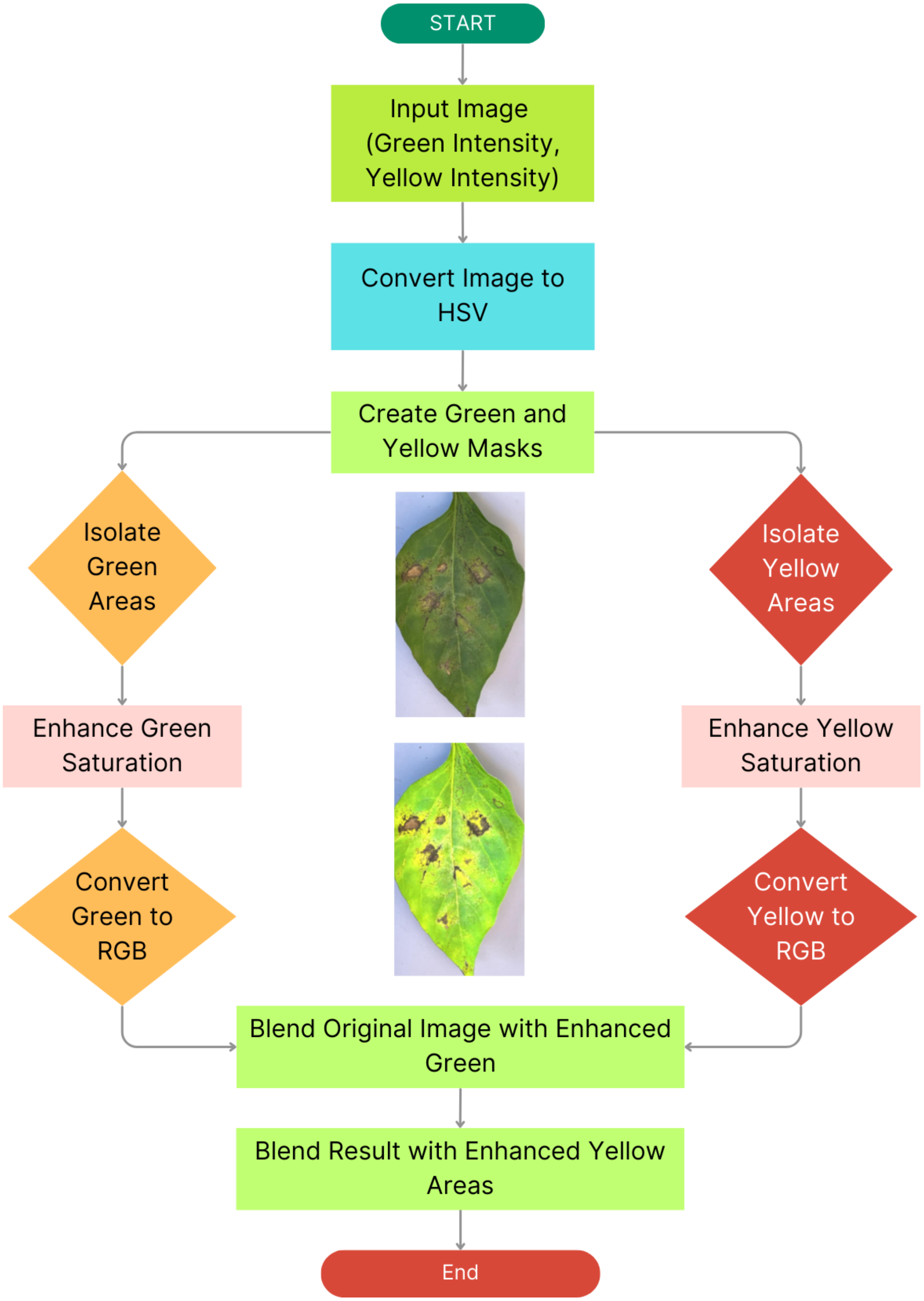


Figure 19. Flowchart Diagram of Color Enhancement

This process is commonly employed for images featuring green and yellow tones, such as those of plants, to enhance the vividness and prominence of these colors. Specifically, since green and yellow tones are indicative of critical elements in plants, such as healthy leaves and fruits, enhancing these colors visually aids the model in better recognizing and distinguishing these features. The increase in the saturation of green and yellow hues aims to improve the model's ability to accurately identify these colors, which is crucial not only for the detection of plant diseases but also for assessing the overall health of the plants. By emphasizing these key color regions, the process contributes significantly to more precise analysis and interpretation.

Various masking techniques were utilized in the present study. These techniques are essential for isolating and processing specific color regions in the images, ensuring that the targeted features are accentuated effectively. Given that this research focuses on plant diseases, masking operations were specifically applied to the green and yellow color regions. This operation plays a critical role, not only in highlighting disease symptoms but also in allowing the model to detect abnormalities within these areas more efficiently and accurately. Figure 18 presents some sample images where the saturation of green and yellow colors has been enhanced, demonstrating the effectiveness of the color enhancement process and helping to visualize its impact on the model's performance.

3.6. Classification with Modified VGGNet-16

The pre-trained VGG16 model was utilized as the deep learning architecture for classifying pepper leaf diseases. VGG16 was selected due to its robust feature extraction capabilities and strong generalization performance among deep neural networks. The model was initialized with pre-trained weights on the large and diverse ImageNet dataset, and these weights were preserved and frozen. This ensured that the fundamental visual features previously learned by the model were retained and provided a solid foundation for classification. Specifically, the core convolutional and MaxPooling layers of the model were set to be non-trainable, preventing any updates during the training process.

For this specific task, customized additional layers were integrated into the VGG16 model. These newly added layers included a flatten layer, dense (fully connected) layers, and dropout layers. The flatten layer facilitated the transfer of outputs from the convolutional layers to the fully connected layers, while the dense layer acted as a fully connected classification layer. In the dense layer, the ReLU activation function was employed to capture non-linear relationships more effectively. Furthermore, this dense layer was configured according to the number of classes corresponding to pepper leaf diseases. To mitigate overfitting, a dropout layer was introduced, playing a critical role in enhancing the overall performance of the model [39].

This customized approach provides a powerful solution for detecting pepper leaf diseases by combining the advantages of the large and complex VGG16 model with the flexibility of the newly added layers. The method not only ensures faster training but also delivers effective classification performance tailored to this specific application.

Additionally, the importance of computational hardware in supporting model performance was emphasized. To process high-resolution images and execute complex algorithms efficiently, powerful hardware specifications are essential. A high-performance NVIDIA graphics card is necessary for CUDA software, enabling efficient training and testing. However, a good graphics card alone is insufficient; a fast processor and ample RAM capacity are also critical to achieving optimal operational performance. Particularly for deep learning models trained on large datasets, the impact of hardware on overall efficiency becomes increasingly significant.

4. ANALYSIS AND DISCUSSION

4.1. Theoretical Background Behind the Performance Analysis of the DL Model

The confusion matrix is one of the most effective methods for summarizing the performance of a classifier, as it displays the absolute number of correct and incorrect predictions on a given test dataset. This matrix provides critical insights into the model's strengths and weaknesses by identifying the classes where it performs well and those where it fails, thereby playing a pivotal role in the evaluation process. To ensure a comprehensive assessment of their performance, the deep learning (DL) models were further evaluated using key metrics, including accuracy, loss, precision, recall, and F1 score. These evaluation metrics not only measure the overall effectiveness of the models but also provide a deeper understanding of their behavior under various conditions. For clarity and improved organization of this section, all symbols utilized in the calculations are presented in Table 2.

Symbol	Definition
a_{ik}	Number of predictions of the ith class predicted as k^{th} class
TP_k	True predictions for the k^{th} class
FP_k	False positive predictions of the k^{th} class
TN_k	True negative predictions for the k^{th} class
FN_k	False negative predictions of the k^{th} class
i	Expected class
k	Predicted class
M	Number of classes
Class 1	APHID
Class 2	BURNT
Class 3	CATERPILLAR
Class 4	HEALTHY
Class 5	MILDEW
Class 6	MITE
Class 7	WORM

Table 2. Classification Results of Test Pepper Images.

Accuracy: The same data set was tested with different models. As seen at the end of this research, different accuracies were achieved. Accuracy is calculated from the confusion matrix. Moreover, accuracy is the natural performance calculation [40]. Accuracy is calculated by the formula stated below. Information in the dataset is used to obtain clearer accuracy. Equal false positive and false negative values in the data set increases the accuracy of the result.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{3}$$

Loss Calculation: The loss function measures the error between the model's predictions and the actual values.

$$Loss = -\frac{1}{M}\sum_{k=1}^{M}\sum_{i=1}^{C} \mathrm{a}_{ik} \cdot \log(a_{ik}) \tag{4}$$

Precision: Briefly, sensitivity calculation is obtained with the following formula. It is obtained by dividing the sum of correctly predicted positive values to the sum of all predicted positive values. The ratio formula used to calculate sensitivity is shown below. [41].

$$Precision_k = \frac{TP}{TP+FP} \tag{5}$$

Recall: It means the ratio of correctly predicted positive values to the total positive values in the confusion matrix. Also known as sensitivity, is the ratio of true positive predictions (TP) to the total actual positive samples (TP + FN). This metric shows how well the model identifies the positive class; in other words, it indicates how effectively the model captures the actual positives.

$$Recall_k = \frac{TP}{TP+FN} \tag{6}$$

F1 Score : F1 Score is the precision and recall frequency of the model. A good F1 Score means that the model predicts fewer false positives and false negatives. The formula used to calculate F1 is given below. İt is the harmonic mean of precision and recall. The metric balances both metrics are used to better reflect the overall model performance, especially in case of class imbalance. The higher the F1 score the better model performance [42].

$$F1\ Score = 2x\frac{Precision\ x\ Recall}{Precision+Recall} \tag{7}$$

4.2.Numerical Results

Prior to training, the dataset is automatically partitioned, with randomly selected images allocated to the training, validation, and testing folders according to the specified ratios. Following the training and validation processes, the model is evaluated using the images in the testing folder. The performance metrics of the model are assessed through the relevant equations, and the results are presented in Table 3.

The model achieved a rather high overall accuracy of 0.92, which suggests that the model is generally performing well across all classes. The macro averages indicate a precision of 0.94, recall of 0.87, and F1-score of 0.87. These figures suggest that while precision is generally high, the model may struggle with recall across certain classes. **The weighted average,** which accounts for the support of each class according to its incidence, indicates that the model's overall performance is slightly better in terms of precision (0.93), recall (0.92), and F1-score (0.90).

The classification model performs well overall, especially in identifying the "CATERPILLER," "HEALTHY," "MITE," and "WORM" classes. The "APHID" class shows significant room for improvement, as indicated by its low precision, recall, and F1 score. While the model's high accuracy is promising, specific classes need targeted adjustments or more training data to enhance their detection capabilities.

Class	**Precision**	**Recall**	**F1-Score**	**Support**
APHID	1.00	0.23	0.38	13
BURNT	0.86	0.97	0.91	33
CATERPILLER	1.00	1.00	1.00	15
HEALTHY	0.94	1.00	0.97	31
MILDEW	0.83	1.00	0.91	29
MITE	0.96	1.00	0.98	22
WORM	1.00	0.89	0.94	27
Accuracy			0.92	170
Macro Avg	0.94	0.87	0.87	170
Weighted Avg	0.93	0.92	0.90	170

Table 3. The Values of The Report (f1-score, recall, precision)

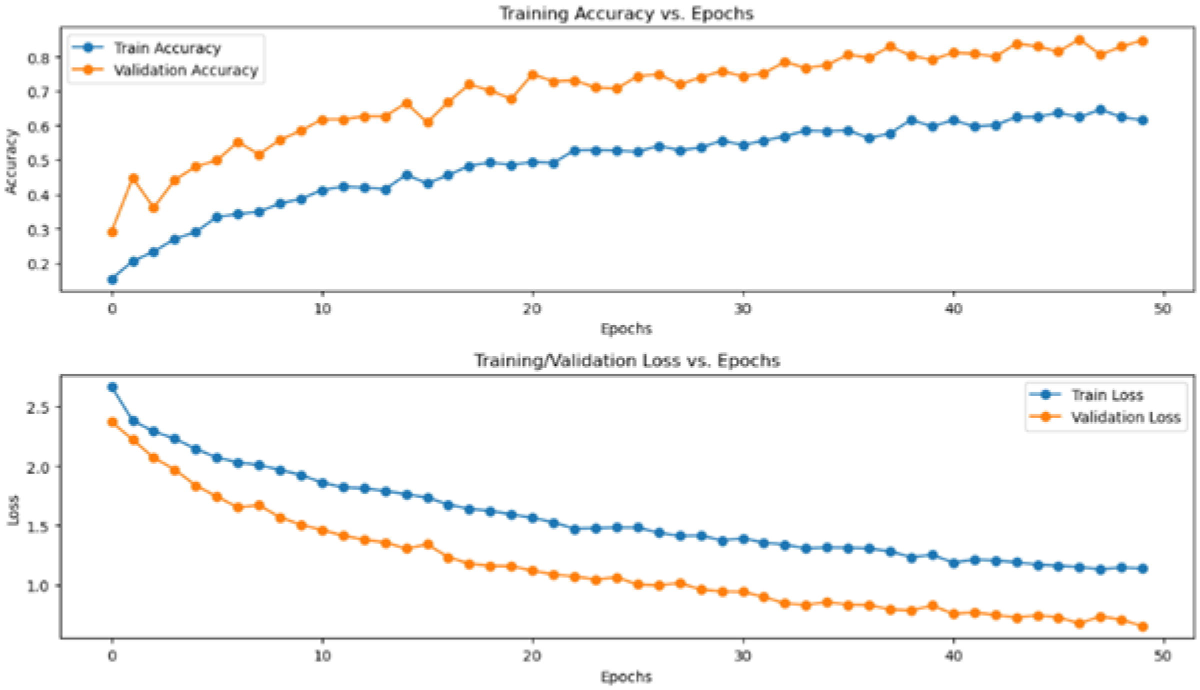
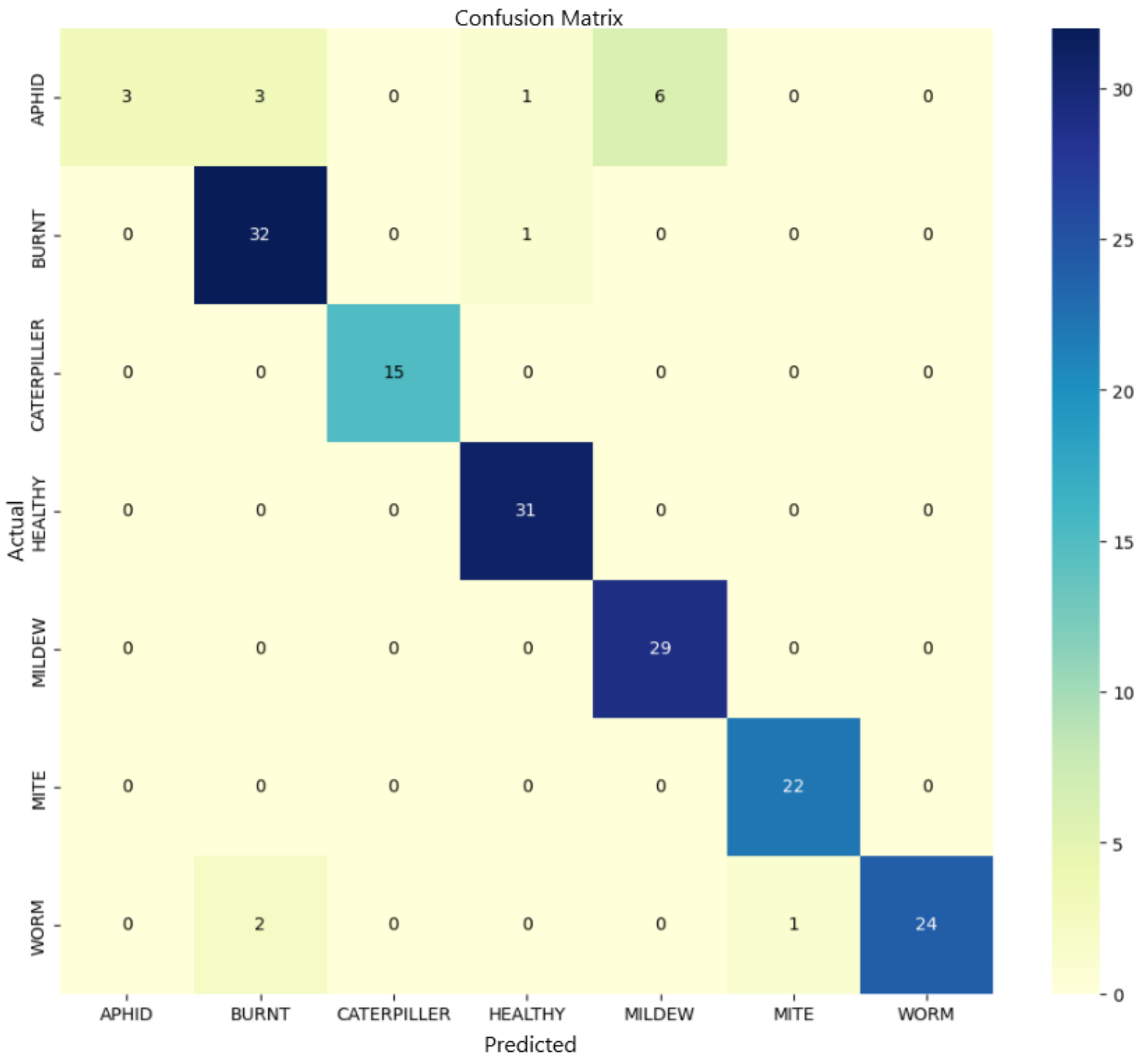


Figure 20. (a) Accuracy and Loss (b) Functions During the Training Process.

Figures 20 (a) and (b) illustrate the progression of the model’s training and validation performance over 50 epochs. The model’s training process is depicted in terms of (a) accuracy and (b) loss. In the first graph, the blue dotted line represents the training accuracy, while the orange dotted line indicates the validation accuracy. The training accuracy begins at 0.2 and increases steadily as the epochs progress, eventually reaching approximately 0.6. Notably, the validation accuracy surpasses the training accuracy, which highlights the model’s strong generalization capability. However, after the 20th epoch, no further improvement in validation accuracy is observed, suggesting that the model has reached its maximum learning capacity from the dataset and is unlikely to exhibit additional progress.

In the second graph, the dotted blue line corresponds to the training loss, whereas the dotted orange line represents the validation loss. The training loss decreases consistently with each epoch, while the validation loss stabilizes around the 20th epoch. Beyond this point, additional training may lead to overfitting, a phenomenon where the model becomes overly specialized to the training data, thereby compromising its generalization ability. To mitigate the risk of overfitting, it is recommended to halt training around the 20th epoch or implement early stopping techniques to prevent the model from undergoing excessive training.



Actual \ Predicted	APHID	BURNT	CATERPILLER	HEALTHY	MILDEW	MITE	WORM
APHID	3	3	0	1	6	0	0
BURNT	0	32	0	1	0	0	0
CATERPILLER	0	0	15	0	0	0	0
HEALTHY	0	0	0	31	0	0	0
MILDEW	0	0	0	0	29	0	0
MITE	0	0	0	0	0	22	0
WORM	0	2	0	0	0	1	24

Figure 21. Confusion Matrix for The Classifier

The obtained test results are also summarised in the confusion matrix shown in Figure 21. Analysis of images in the test folder revealed that the learning process yielded foolproof outcomes for some specific classes, such as HEALTHY, CATERPILLAR, MILDEW, and MITE. Minor classification errors are obtained for BURNT and WORM classes, while misclassifications in the APHID class are more challenging. Although APID predictions are mostly confused with MILDEW, there is some minor confusion with the BURNT and HEALTHY classes.

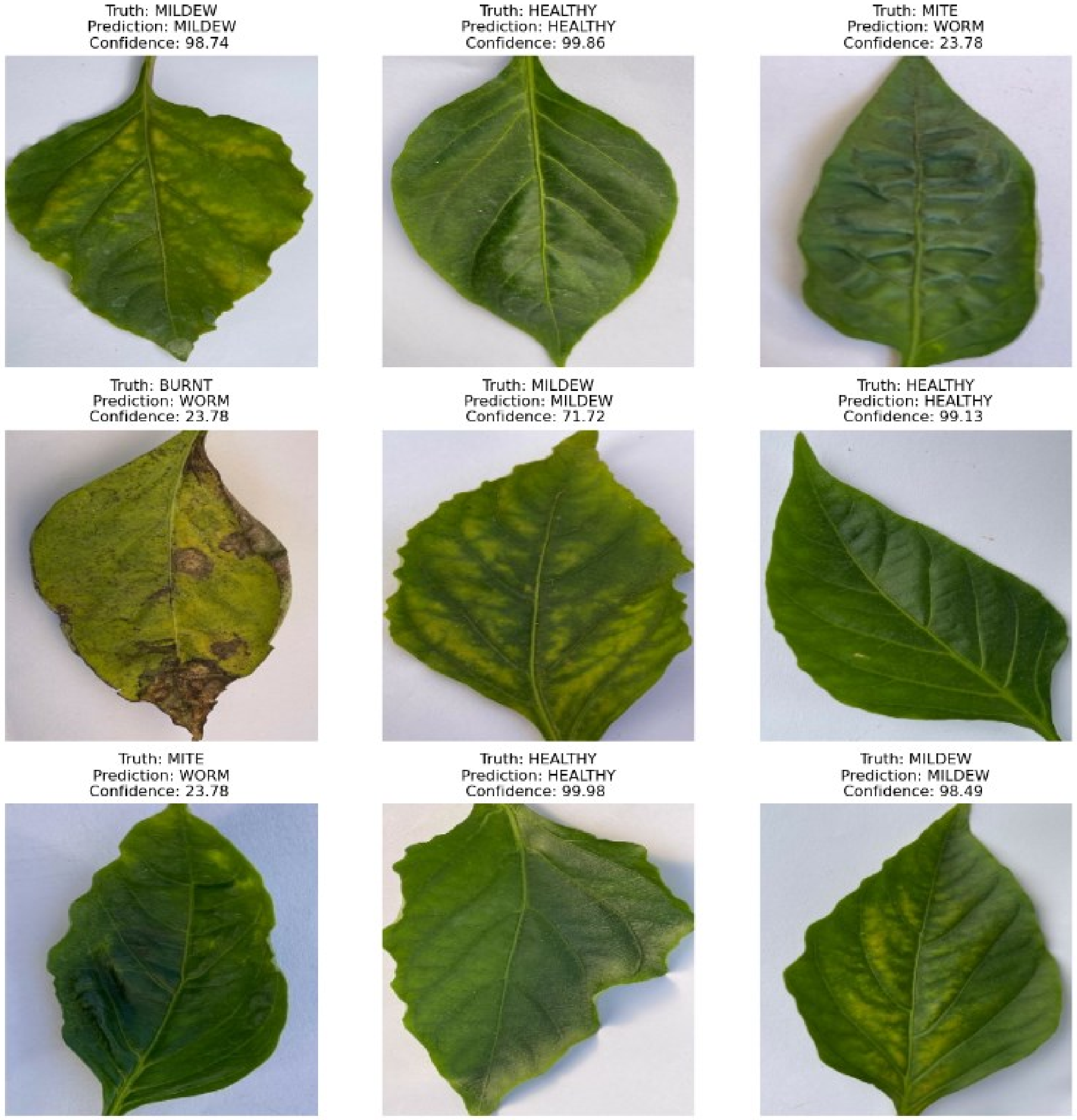


Figure 22. The Values of the Accuracy and Loss Functions

4.3. Performance Analysis and Discussion of the DL Model

The classification of various diseases affecting pepper leaves is bas ed on their distinct visual and color differences, which provides important guidance for accurate differentiation. Healthy pepper leaves are predominantly uniform, bright green in color, without any discoloration or structural damage. In contrast to the healthy ones, burnt-affected leaves are characterized by brown, ring-shaped spots, which generally spread from the edges of the leaf toward the center. The aphid class is defined by black dots, which are predominantly located on the underside of the leaf. The caterpillar class is distinguished by characteristic leaf deficiencies, irregular edges, and deep perforations; these are among its most prominent features and typically result from the feeding habits of caterpillars. The mite class exhibits wrinkled leaves, folds, and dark green layers, which indicate the intense stress imposed by mites on the plant. In the worm class, the formation of white lines on the leaf surface, creating irregular patterns, is identified as a key distinguishing detail. Finally, the mildew class is primarily characterized by yellow-colored decay, lesions, and irregular-sized spots, which severely affect the overall health of the leaf.

Comprehensive preprocessing, guided by these distinguishing characteristics, along with high-quality photographic lighting and optimized color contrast, has made these differences more pronounced, thereby improving the model's ability to recognize and classify these diseases. Moreover, this process not only improved the model’s accuracy but also enhanced human observers' ability to identify and differentiate between these classes more effectively.

To the best of our knowledge, this study is the first to focus on the aphid and caterpillar classes. The specialized dataset collected for this study, though slightly imbalanced, contains more than 220 images per class, which is a reasonable average compared to other datasets in this field, and it is sufficient for making meaningful inferences. However, the results show that the relatively small number of aphid-infected leaves negatively affected the accuracy. Predictions for the aphid class were often confused with images from the burnt and mildew classes. Further investigation revealed that this issue stemmed from aphid class images being captured under varying lighting conditions and at different times during dataset preparation. Additionally, the varying angles and perspectives of the aphid images also led to misclassification by the model. Furthermore, the model’s sensitivity to cool and warm color tones contributed to the classification errors during the training process. This underscores the importance of using a more balanced dataset and standardized imaging conditions. In Figure 22 above, the predicted values made by the algorithm for randomly selected pepper disease images, the class assigned based on this prediction, and the actual class are shown.

To provide a more comprehensive assessment of the accuracy, the model was compared with others presented recently in the literature. Several studies were reviewed according to the criteria listed in Table 4. As shown in Table 4, higher accuracy rates are achieved for a smaller number of tasks, where the number of patterns the model needs to learn is significantly lower.

This results in higher accuracy scores, and when considering the scope and complexity of the dataset used in the present study, it adds value to the interpretation of the results.

Author(Year)/ Cite	Dataset/Availability	Num.of Samples	Classifier	Classes	Method
Begum et al. (2024)[43]	Plant Village/ Public	1855	2-class	Healthy and Bacterial Spots	93.50
Bezabh et al. (2023) [44]	North Macha Worenda/Public	1596	4-class, 2-class	Healthy and Bacterial Spots	97.34
Yin et al. (2020) [45]	National Institute of Horticultural And Herbal Science /Public	1977, 2621	15-class, 19-class	15 class pepper diseases and 19 class pepper pest	70.45 and 88.72
Fu et al. (2024) [46]	Nanchang Academy of Agricultural Sciences /Public	1262	4-class	healthy, brown spot, leaf mold, and viral diseases	76.89 and 87.38
Cetinkaya	Private	1600	7-class	Healthy, caterpillar, mildew, mite, burnt, worm, and aphid	92.00

Table 4.A List of Similar Studies That Have Been Presented Recently.

5. CONCLUSIONS

The study introduces a modified deep learning (DL)-based model designed to classify six pepper leaf diseases alongside healthy leaves. Notably, this is the first instance where two classes, aphids and caterpillars, are included in the diagnostic process. A private dataset, specifically collected for this research, consists of 90 aphid-infected leaves and 140 caterpillar-infested leaves. Unlike prior studies that primarily focused on enhancing contrast settings, this research emphasizes a novel preprocessing approach aimed at enhancing distinguishing characteristic colors among classes. To achieve this, the color settings were specifically optimized for sensitivity to yellows and browns. A more comprehensive preprocessing pipeline has successfully highlighted distinguishing features for each class, thereby positively influencing the model’s learning performance.

The VGG16-based model, selected for its superior feature extraction capabilities, was initialized with pre-trained weights from the ImageNet dataset. These weights were preserved and frozen, ensuring that all convolutional and MaxPooling layers of the VGG16 architecture remained non-trainable. By adding and training only newly introduced layers, the model not only achieves faster training but also enhances pattern recognition, which is critical for diagnostic performance.

While the results demonstrate high accuracy across most classes, minor classification errors are observed in the BURNT and WORM classes, with more pronounced misclassifications in the APHID class. The relatively small number of aphid-infected leaves is believed to have limited the model’s accuracy. Additionally, variability in lighting conditions during data collection and the model’s sensitivity to cool and warm color tones contributed to classification errors during training. Consequently, future work will focus on expanding the dataset for underrepresented classes and enhancing the model's robustness to variations in color tones, particularly sensitivity to cool and warm hues.

Building upon our research, we anticipate achieving significant progress in the next phase. Our plan involves expanding the current dataset by incorporating additional disease classes and increasing the number of images within each class to enhance the training process of the algorithm. Furthermore, we aim to develop a more distinguishable dataset by integrating diverse lighting variations and employing advanced masking techniques during data preparation. Additionally, one of our primary objectives is to optimize efficiency by leveraging alternative algorithms to reduce analysis time and deliver results more rapidly, thereby improving overall performance.

REFERENCES

[1] Abade, A., Ferreira, P.A., de Barros Vidal, F.: Plant diseases recognition on images using convolutional neural networks: A systematic review. Computers and Electronics in Agriculture 185, 106125 (2021) https://doi.org/10.1016/j.compag. 2021.106125.

[2] V. Singh, N. Sharma, and S. Singh, “Arti fi cial Intelligence in Agriculture A review of imaging techniques for plant disease detection,” Artif. Intell. Agric, pp. 229–242, 2020.

[3] V. Singh and A.K. Misra, “Detection of Plant Leaf Diseases using Image Segmentation and Soft Computing Techniques,” Information Processing in Agriculture, pp. 41-49, 2017.

[4] R. K. Sarkar, “Segmentation of Plant Disease Spots Using Automatic SRG Algorithm : A Look Up Table Approach,” 2015 Int. Conf. Adv. Comput. Eng. Appl, pp. 129–133, 2015.

[5] M. N. Saranya, “Detection of Banana Leaf and Fruit Diseases Using Neural Networks,” Proc. Second Int. Conf. Inven. Res. Comput. Appl. IEEE Xplore Part, pp. 493–499, 2020.

[6] Van den Oord, Aaron; Dieleman, Sander; Schrauwen, Benjamin “Deep content-based music recommendation.” pp. 2643–2651. 2022

[7] Kaya, Y., Gürsoy, E.: A novel multi-head cnn design to identify plant diseases using the fusion of rgb images. Ecological Informatics 75, 101998 (2023) https: //doi.org/10.1016/j.ecoinf.2023.101998

[8] Lee, S.H., GoAau, H., Bonnet, P., Joly, A.: New perspectives on plant disease char- ˜ acterization based on deep learning. Computers and Electronics in Agriculture 170, 105220 (2020) https://doi.org/10.1016/j.compag.2020.105220

[9] S. Liu and W. Deng, “Very deep convolutional neural network based image classification using small training sample size,” pp.10-15. 2016.

[10] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards Realtime Object Detection with Region Proposal Networks,” IEEE Trans. Pattern Anal. Mach. Intell., pp. 1137-1149, 2017.

[11] E. Elhariri, N. El-Bendary, A. M. M. Hussein, A. E. Hassanien, and A. Badr, “Bell pepper ripeness classification based on support vector machine,” ICET 2014 - 2nd Int. Conf. Eng. Technol. pp. 5-10, 2015.

[12] V. Singh and A.K. Misra, “Detection of Plant Leaf Diseases using Image Segmentation and Soft Computing Techniques,” Information Processing in Agriculture, pp. 41-49, 2017.

[13] K. Golhani, S.K. Balasundram, G. Vadamalai, and B. Pradhan, “A Review of Neural Networks in Plant Disease Detection using Hyperspectral Data,” Information Processing in Agriculture, pp. 354-371, 2018.

[14] Y.Toda and F. Okura, "How Convolutional Neural Networks Diagnose Plant Disease", Plant Phenomics, pp. 1-14, 2019.

[15] Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., Fei-Fei, L., Imagenet: A large-scale hierarchical image database, IEEE conference on computer vision and pattern recognition, IEEE, pp. 248-255, 2009.

[16] Krizhevsky, Alex, Sutskever, I., Geoffrey, E.H., Imagenet classification with deep convolutional Neural Networks, Communications of the ACM, pp.84-90, 2017.

[17] Wagle, S. A. and Harikrishnan, R. (2021) Comparison of Plant Leaf Classification Using Modified AlexNet and Support Vector Machine, Traitement Du Signal, pp 79-87. 2021

[18] Atlas, Homma, and Marks. "An Artificial Neural Network for Spatio-Temporal Bipolar Patterns: Application to Phoneme Classification." pp. 50-55. 2021

[19] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” 2015.

[20] Chadha, S., Sharma, M. and Sayyed, A. (2021) Advances in sensing plant diseases by imaging and machine learning methods for precision crop protection, Microbial Management of Plant Stresses: Current Trends, Application and Challenges, pp.157–183, 2021.

[21] G. F. Antonious, “Capsaicinoids and vitamins in hot pepper and their role in disease therapy,” in Capsaicin and its human therapeutic development, IntechOpen, 2018.

[22] Souid, A., Sakli, N., Sakli, H., "Classification and Predictions of Lung Diseases from Chest X-rays Using MobileNet V2", Applied Sciences, pp. 2751, 2021.

[23] H. Ajra, “Disease Detection of Plant Leaf using Image Processing and CNN with Preventive Measures,” pp. 5–7, 2020

[24] B. K. Saleh, A. Omer, and B. Teweldemedhin, “Medicinal uses and health benefits of chili pepper (Capsicum spp.): a review,” MOJ Food Process Technol, pp. 325–328, 2018.

[25] T. Nitte, T. Nitte, and T. Nitte, “A Machine Learning Technique for Identification of Plant Diseases in Leaves,” Conf. Inven. Comput. Technol. [ICICT 2021] IEEE, pp. 481–484, 2021.

[26] S. V. Militante, “Fruit grading of Garcinia Binucao (Batuan) using image processing,” International Journal of Recent Technology and Engineering, pp. 1828–1832, 2019.

[27] A. Kaya, A. Seydi, C. Catal, H. Yalin, and H. Temucin, “Analysis of transfer learning for deep neural network based plant classification models.” pp. 20–29. 2019

[28] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards Realtime Object Detection with Region Proposal Networks,” IEEE Trans. Pattern Anal. Mach. Intell., pp. 1137-1149, 2017.

[29] Sardogan, M., Tuncer, A., & Ozen, Y. “Plant leaf disease detection and classification based on CNN with LVQ algorithm” 3rd International Conference on Computer Science and Engineering. pp. 382-385, 2018.

[30] N. Hossain, M. M. Ahamad, S. Aktar, and M. A. Moni, “Movie genre classification with deep neural network using poster images,” International Conference on Information and Communication Technology for Sustainable Development pp. 82-85, 2021.

[31] N. Jayaram, A. M. Rao, S. Ramesh, B. Manjunath, N. Mangalagowri, and M. Ashwini, “Tagging ssr markers to genomic regions associated with anthracnose disease resistance and productivity per se traits in hot pepper (Capsicum annuum L.),” Env. Ecol, pp. 1440–1446, 2016.

[32] K.P. Ferentinos, “Deep Learning Models for Plant Disease Detection and Diagnosis,” Computers and Electronics in Agriculture, pp. 311-318, 2018.

[33] G. Hu, X. Yang, Y. Zhang, and M. Wan, “Identification of Tea Leaf Diseases by using an Improved Deep Convolutional Neural Network,” Sustainable Computing: Informatics and Systems, 100353, 2019.

[34] G. Geetharamani, and A. Pandian, “Identification of Plant Leaf Diseases using a Nine-layer Deep Convolutional Neural Network,” Computers & Electrical Engineering, pp. 323-338, 2019.

[35] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” Nature, vol. 521, no. 7553, pp. 436–444, 2015.

[36] Chadha, S., Sharma, M. and Sayyed, A. (2021) Advances in sensing plant diseases by imaging and machine learning methods for precision crop protection, Microbial Management of Plant Stresses: Current Trends, Application and Challenges, pp.157–183, 2021.

[37] M. Islam, A. Dinh, and K. Wahid, “Detection of Potato Diseases Using Image Segmentation and Multiclass Support Vector Machine,” pp. 8–11, 2017.

[38] A.K. Rangarajan, R. Purushothaman, and A. Ramesh, “Tomato Crop Disease Classification using Pre-trained Deep Learning Algorithm,” Procedia Computer Science, pp. 1040-1047, 2018.

[39] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.-C., "MobileNetV2: Inverted Residuals and Linear Bottlenecks", IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4510–4520, 2018.

[40] S. Liu and W. Deng, “Very deep convolutional neural network based image classification using small training sample size,” pp.10-15. 2016.

[41] K. Li, J. Lin, J. Liu, and Y. Zhao, “Using deep learning for Image-Based different degrees of ginkgo leaf disease classification,” Information, vol. 11, no. 2, pp. 95. 2020

[42] Fu, Y., Guo, L., Huang, F.: A lightweight cnn model for pepper leaf disease recognition in a human palm background. Heliyon 10(12), 33447 (2024) https: //doi.org/10.1016/j.heliyon.2024.e33447

[43] Begum, S.S.A., Syed, H.: Gsatt-cmnetv3: Pepper leaf disease classification using osprey optimization. IEEE Access 12, 32493–32506 (2024) https://doi.org/10. 1109/ACCESS.2024.3358833

[44] Bezabh, Y.A., Salau, A.O., Abuhayi, B.M., Mussa, A.A., Ayalew, A.M.: Cpdccnn: classification of pepper disease using a concatenation of convolutional neural network models. Scientific Reports 13(1), 15581 (2023) https://doi.org/10.1038/ s41598-023-42843-2

[45] Yin, H., Gu, Y.H., Park, C.-J., Park, J.-H., Yoo, S.J.: Transfer learning-based search model for hot pepper diseases and pests. Agriculture 10(10) (2020) https: //doi.org/10.3390/agriculture10100439

[46] Fu, Y., Guo, L., Huang, F.: A lightweight cnn model for pepper leaf disease recognition in a human palm background. Heliyon 10(12), 33447 (2024) https: //doi.org/10.1016/j.heliyon.2024.e33447