WOLDIA UNIVERSITY

INSTITUTE OF TECHNOLOGY

SCHOOL OF COMPUTING

DEPARTMENT OF INFORMATION TECHNOLOGY

DEVELOPMENT OF MAIZE LEAF DISEASES AND PESTS IDENTIFICATION MODEL USING DEEP LEARNING

By

PAULOS MEKONNEN SISAY

A Thesis Submitted to the Department of Information Technology Presented in Partial Fulfillment of the Requirements for the Degree of Master of Science in Information Technology

JUNE, 2024 G.C.

WOLDIA, ETHIOPIA

Declaration

I, Paulos Mekonnen, hereby declare that the thesis entitled "Development of a Maize Leaf Disease and Pest Identification model using deep learning" is my original work. I have conducted this research independently under the guidance and support of my research advisor. This study has not been previously submitted for any degree or diploma program in this or any other institution. I acknowledge and give proper credit to all sources of materials used in this thesis.

I affirm that:

- ✓ The research work presented in this thesis is the result of my own efforts and contributions.
- ✓ Any ideas, concepts, or information obtained from external sources have been properly cited and referenced.
- ✓ I have not engaged in any form of plagiarism or academic dishonesty during the course of this research.
- ✓ This thesis has not been submitted in whole or in part for any other academic qualification.
- ✓ I understand that any violation of academic integrity or misrepresentation of the work presented in this thesis may have serious consequences, including the revocation of my degree.

Signature: ____________

Candidate Name: Paulos Mekonnen

Date: 17/06/2024

Signature: ____________

Advisor Name: Dr. T. PRIN

Date: 17/06/2024

Certification

This is to certify that I, Paulos Mekonnen Sisay, have prepared the thesis titled "Development of a Maize Leaf Disease and Pest Identification model using deep learning" as a partial fulfillment of the requirements for the Degree of Master in the Institute of Technology. This thesis adheres to the University's regulations and meets the prescribed standards in terms of originality and quality.

Name of Candidate: Paulos Signature: ________ Date: 17/06/2024

Name of Advisor: Dr. T. PRINCE Signature: ________ Date: 17/06/24

Signature of Board of Examiners:

External examiner: Dr. Alemu Jorgi Signature: ________ Date: July 4, 2024

Internal examiner: ________________ Signature: ____________ Date: _____________

Name of chair Person: ______________ Signature :_____________ Date :____________

Acknowlgement

I would like to express my sincere gratitude to the Almighty God for his blessings and guidance throughout this thesis work.

I would also like to extend my heartfelt appreciation to my advisor, Dr. Prince, for his invaluable support and constructive guidance from the inception to the completion of this research. His motivation and insightful discussions have continually inspired me and instilled the belief that all challenges can be overcome.

I am thankful to all those who have contributed to this research in various ways, including providing resources, annotated datasets, assistance, and encouragement. Your contributions have been instrumental in shaping the outcome of this thesis work.

Finally, I would like to acknowledge the support of my family and friends for their unwavering encouragement and understanding during this undertaking. Their belief in my abilities has been a constant source of motivation.

To all those who have played a part in this study, directly or indirectly, I offer my heartfelt appreciation and gratitude.

Table of Contents

List of Table

List of Figure

List of Abbreviation

ANN	Artificial Neural Network
CLR	Common Leaf Rust
CNNS	Convolutional Neural Networks
DL	Deep Learning
DT	Decision Tree
FAW	Fall Armyworm
FN	False negative
FP	False Positive
GF	Gaussian Filter
GLS	Gray Leaf Spot
KDD	Knowledge Discovery from Data
KNN	K-Nearest Neighbor
MF	Median Filter
MLB	Maydis Leaf Bight
MSV	Maize Streak Virus
NB	Naive Bayes
NLB	Northern Leaf Blight
RELU	Rectified Linear Unit
RF	Random Forest
RNN	Recurrent Neural Network
SSA	Sub-Saharan Africa
SVM	Support Vector Machine
TLB	Turcicum Leaf Blight
TN	True Negative
TP	True Positive

Abstract

Ethiopia possesses tremendous potential for the development of diverse crop varieties, including those used for food, shelter, medicine, and daily necessities. Maize, a crucial crop worldwide, plays a significant role in providing sustenance and income for both rural and urban communities. However, the productivity of maize is often hindered by diseases and pests. Early detection and effective control of these issues are essential for improving crop yield. Unfortunately, many farmers lack access to agricultural experts capable of accurately identifying and treating these problems. This research paper developed a deep-learning model for the identification of common rust diseases and Fall Armyworm pests in maize leaves. To develop and evaluate the proposed method, a comprehensive dataset comprising 3899 images was collected, serving the purposes of training, validation, and testing. The methodology involves several key steps, including image preprocessing, feature extraction, and the training of both a Sequential model and an EfficientNetB0 model. To enhance the quality of maize leaf images and remove noise, various filtering techniques such as Gaussian filtering, median filtering, and a hybrid approach combining the two were compared. The comparative analysis revealed that the hybrid approach, which combines Gaussian and median filtering, yielded superior results. Irfanview64 software was applied for denoising, resizing, and augmenting images, while a convolutional neural network (CNN) was used for feature extraction. In the evaluation of the proposed model, the performance of the Sequential model was compared with the EfficientNetB0 model. The EfficientNetB0 model demonstrated significantly superior performance across all evaluation metrics. Specifically, it achieved perfect accuracy of 100% in both training and validation, as well as 100% accuracy in testing. Furthermore, the model achieved an exceptional overall classification accuracy of 100% based on the f1 score. This deep learning model offers a cost-effective and accurate solution for diagnosing maize leaf diseases and pests, addressing the lack of agricultural experts. Early detection and treatment can significantly improve maize crop yield, contributing to food security.

Keywords: deep learning, convolutional neural network, Sequential model, EfficientNetB0, maize leaf disease and pest classification

CHAPTER ONE

1. Introduction

1.1. Background

Agriculture is the backbone of the world economy. Maize, scientifically known as Zea mays, is a highly important cereal crop in the world, serving as a crucial source of sustenance and income for rural and urban communities. Maize (Zea mays) is believed to have originated from central Mexico 7000 years ago from wild grass, and Native Americans transformed maize into a better source of food and income[1].

Sub-Saharan Africa (SSA) people especially rural communities dependent on maize, with over 40 million hectares dedicated to its cultivation. Smallholder farmers in SSA produce more than 70 million metric tons of maize grain, making it the most essential cereal crop in the region. In Ethiopia, maize plays a critical role in ensuring food security which mostly used for indigenous food like bread, injera, qolo, nifiro, genfo, qita and soup. Approximately 9 million smallholder farmers cultivate maize on around 2 million hectares of land, covering 14% of Ethiopia's total land area [2].

Maize is widely grown in Ethiopia in diverse agro-climate conditions. It is one of the most important cereal crops selected for food security mainly due to its high productivity, and wider adaptability. Maize is the second most important cereal crop after Teff in area coverage. In 2007, it was produced on 2.1 million ha of land which covers about 21.7% of all the land allowed for cereals production [2].

Maize has emerged as a critical staple food crop in Ethiopia, providing sustenance for a large portion of the population. The crop's importance has significantly increased in the wake of the major drought and famine event in 1984, which highlighted the need to strengthen the country's food security. Today, maize is cultivated by more smallholder farmers in Ethiopia than any other agricultural commodity. This widespread adoption underscores the crop's growing significance in the livelihoods and food supplies of rural communities across the nation [3].

However, maize cultivation in Ethiopia faces numerous challenges, with diseases and pests being major constraints. There are more than 72 reported cases of diseases caused by fungi, bacteria, nematodes, viruses, and pests that affect maize. Among these, common rust disease and fall armyworm pests are prevalent on maize leaves [3].

In Africa; Ethiopia is the largest producer of maize. However, the crop is often affected by various diseases caused by fungi, bacteria, viruses, and pests, which can significantly reduce productivity. In Ethiopia, foliar diseases affect maize like Common leaf rust (CLR) and turcicum leaf blight (TLB). Notably, the common rust disease caused by the fungus Puccinia sorghi and fall poses a substantial risk to maize cultivation in Ethiopia [4].

Habru Woreda Agricultural Office plant pathologist Mr. Yimer described and other studies showed that Ethiopia especially as Habru Woreda maize cultivated tropical areas has been severely affected by the FAW within a short period.

Developing an accurate and reliable maize leaf disease and pest identification model using deep learning is crucial for addressing the challenges faced by smallholder farmers, like Ethiopia particularly in low- and middle-income countries. These farmers often struggle to combat pest and disease outbreaks, which have significant negative impacts on agricultural productivity, sustainability, and resilience. Globally, plant diseases and pests result in staggering economic losses of approximately US $220 billion every year. Notably, regions with food deficits and rapidly growing populations experience the highest agricultural losses due to these issues [5].

Early and accurate maize leaf diseases and pest identification models are essential for implementing timely control measures and reducing yield losses. However, in Ethiopia currently, disease identification and classification rely on traditional methods such as visual inspection and chemical analysis which are time-consuming and require expertise.

Visual inspection by agricultural experts is the primary approach, but it is subjective and prone to errors. Moreover, not all agricultural regions have access to experts, leading to delayed or inadequate responses to disease outbreaks. The timely detection and management of diseases and pests are crucial for optimizing maize crop yield and quality.

To tackle this problem, a recent analysis emphasized the importance of leveraging advanced technologies like deep learning to develop an effective plant health management solution. By training a deep learning model specifically tailored for leaf disease and pest identification, farmers can quickly and accurately diagnose and address plant health issues.

This study was developed using deep learning algorithms and highly accurate and efficient models for identifying common rust disease and fall armyworm pests in maize leaves. This model utilized a vast dataset of annotated images representing various maize leaf diseases and pests to learn and identify specific patterns and characteristics associated with each condition.

By applying noise filtering techniques Gaussian, median, and the combination of the Gaussian and median filtering techniques then train a sequential and the EfficientNetB0 model, accurate identification of these diseases and pests was achieved, enabling farmers to implement timely disease management strategies, ensure effective disease control, minimize crop losses, and secure food production.

1.2. Motivation

The economic development of many countries, especially those dependent on agriculture and industry, relies heavily on the success of their agricultural sector. Ethiopia is one such country where agriculture plays a crucial role in the economy, with maize being one of the most important cereal crops. Maize not only provides a vital source of food but also serves as a significant income generator for rural communities. However, the production of maize is hindered by various challenges, and one of the major contributors to yield losses is the damage caused by diseases and pests.

Traditional methods of disease and pest diagnosis, such as visual inspection by agricultural experts, present several limitations. These methods are subjective, time-consuming, and often require specialized expertise that may not be readily available in certain regions. To overcome these challenges, automating the disease identification process using advanced deep learning algorithms and approaches has been explored in previous studies, particularly in the domain of plant disease recognition and identification. Deep learning techniques, specifically Convolutional Neural Networks (CNNs), have shown great promise in various image classification tasks.

This research proposed an accurate and efficient model for identifying two significant threats to maize crops: common rust disease and fall armyworm pests by utilizing the hybrid of Gaussian and Median noise filtering techniques from maize leaf images and to improve the quality of maize leaf datasets and employing deep learning techniques, including both sequential and EfficientNetB0 models, the proposed research seeks to enhance the quality of the dataset using noise filtering techniques. This approach enabled the model to achieve timely disease identification, empowering farmers to implement appropriate measures for disease and pest management, as well as crop protection.

1.3. Statement of the Problem

Agriculture is the most important economic activity in Ethiopia and one of the most cereal crops next to Teff are maize crop products. But in Ethiopia, maize production faces significant challenges due to the prevalence of diseases such as common leaf rust (CLR) and pests like fall armyworm. The methods of maize diseases and pests identification in Ethiopia is through a traditional method of disease diagnosis and classification, relying on visual inspection by agricultural experts, pose several problems. These include subjectivity, time-consuming processes, and limited access to specialized expertise in certain regions.

In Ethiopian maize diseases recognition and classification using different algorithms and approaches has been explored in previous studies, but there is a need for robust and accurate identification systems that overcome the limitations of traditional methods and existing machine learning approaches. The advanced solution is development of maize diseases and pest identification and classification model using deep learning algorithms and to enhance the dataset quality by using hybrid of GF-MF noise removal techniques.

There are previous studies on plant diseases recognition and identification technique using imaging processing and machine learning including maize plants. The paper studied by [4] Ethiopian maize diseases recognition and classification using support vector machine, used a traditional feature extraction methods combined (texture, colour and morphology) and the authors consider both the median filtering and Gaussian filter techniques. But the Authors proposed that Gaussian filter is more accurate than median filter to reduce noises.

In this study the proposed model to address the problem by developing a robust and accurate Sequential and EfficientNetB0 model for identifying common rust disease and fall armyworm pests in maize leaves and to enhance the dataset by using Gaussian, Median and the combination of Gausian and median filtering techniques to remove noises from maize leaves. But the hybrid of Gf-MF techniques more accurate than the two individual filtering techniques and train the proposed model to enable timely disease detection, empowering farmers to take appropriate measures for disease management and crop protection. Accurate and timely identification of these diseases and pests is crucial for effective disease and pest management, minimizing their negative impacts on crop yield and quality.

Furthermore, existing approaches for plant disease identification often rely on limited datasets or lack comprehensive data from specific regions. There is a need to enhance the quality of the dataset by applying the hybrid noise-filtering techniques and incorporating region-specific data to ensure the model's effectiveness in diverse agricultural contexts.

1.4. Research Questions

- ✓ What are suitable methods and techniques to apply and prepare quality image datasets for experimentation?
- ✓ Which deep learning algorithm is suitable for detecting common rust maize leaf disease and fall armyworm pests?
- ✓ To what extent does the proposed model work in classifying common rust maize leaf disease and fall armyworm pests?

1.5. Objective of the Study

1.5.1. General Objective

To develop a deep learning model for accurate identification of common rust disease and fall armyworm pest in maize leaves.

1.5.2. Specific Objectives

- ✓ Collecting a diverse dataset of maize leaf images infected with Common Rust Maize Leaf Disease, Fall Armyworm Pest, and Healthy for model training and evaluation.
- ✓ To enhance the quality of the dataset by employing noise filtering techniques and incorporating region-specific data to improve the model's accuracy and generalization capabilities.
- ✓ Developing and Training the deep learning model using the annotated dataset and optimizing hyperparameters for accurate identification.
- ✓ To evaluate the performance of the developed model using appropriate metrics such as accuracy, precision, recall, and F1 score.
- ✓ Compare the performance of the developed Sequential and EfficientNetB0 model a state-of-the-art deep learning architecture known for its efficiency and accuracy.

1.6. Scope of the Study

There are five major cereals crops (Teff, Wheat, Barley, Maize and Sorghum) which are produced in Ethiopia. The main focus of this research is designing, modeling and development of an automatic identification of maize leaves Common rust disease and fall armyworm pest by using deep learning. While common rust diseases and Fall Armyworm pests were chosen as the primary targets in this research, it is important to note that there may be other diseases and pests that also significantly affect maize crop productivity. These could include diseases like northern corn leaf blight, southern corn leaf blight, maize dwarf mosaic virus, or pests like corn earworm, corn borers, or aphids.

The decision to focus on specific diseases and pests is often driven by the urgency of the problem, the availability of data and expertise, and the potential impact on farmers' livelihoods. In this study, common rust diseases and Fall Armyworm pests were likely chosen due to their widespread occurrence, severe impact on maize crops in Ethiopia especially tropical areas like Habru Woreda, and the need for developed accurate maize leaf disease and pest identification model using deep learning.

1.7. Limitation of the Study

Due to time constraints, budget limitations, and data availability, this research focuses solely on maize leaves infected with common rust disease and fall armyworm pests. Other, diseases like northern corn leaf blight, southern corn leaf blight, maize dwarf mosaic virus, or pests like corn earworm, corn borers, or aphids and disease severity classification were not included within the scope of this thesis work.

1.8. Significance of the study

The successful development of an accurate and robust model for identifying common rust disease and fall armyworm pests in maize leaves has several significant implications:

- ✓ Improved disease management: Early and accurate detection of common rust disease and fall armyworm pests enables farmers to implement timely control measures, reducing crop losses and improving disease management strategies.
- ✓ Enhanced crop yield and quality: Timely disease detection and appropriate interventions can help protect maize crops from the negative impacts of diseases and pests, leading to improved crop yield and quality.
- ✓ Cost-effective solutions: Automated disease identification using deep learning algorithms provides a cost-effective alternative to traditional methods, which often require specialized expertise and extensive time commitments.
- ✓ Empowerment of farmers: The proposed model empowers farmers by providing them with a tool for accurate disease identification, enabling them to make informed decisions regarding disease management and crop protection.
- ✓ Food security: By minimizing crop losses due to diseases and pests, the model contributes to food security by ensuring a stable and reliable supply of maize, a staple crop in many regions.
- ✓ Expansion to other crops and regions: The developed model can serve as a foundation for similar disease identification systems for other crops and in different regions, addressing

the challenges of disease management and crop protection across diverse agricultural contexts.

1.9. Organization of the Thesis

This thesis work consists of five chapters. The introductory chapter provides an overview of the research topic and its significance. The second chapter encompasses a comprehensive literature review and analysis of related works in the field. In the third chapter, the research methodology is described, covering aspects such as the dataset, system architecture, image preprocessing techniques, and details of the CNN layers used for model training. The fourth chapter presents the experimental setup, results, and evaluation of the model's performance. It also discusses the practical implementation of the proposed system and its effectiveness based on the obtained results. Finally, the fifth chapter concludes the thesis, summarizing key findings, and providing recommendations for future research or potential enhancements to the proposed system.

CHAPTER TWO

2. Literature Review

2.1. Overview

In this chapter, an overview of Maize, Maize diseases, and pests, focusing on relevant studies conducted on maize diseases and pests, as well as previous works related to disease detection, image processing, pattern recognition, and deep learning models and image processing machine and deep learning, convolutional neural network, Sequential model and CNN Model architectures also discussed image processing steps followed, such as image preprocessing, image segmentation and feature extraction. A literature review is made concerning machine learning algorithms with specific consideration of deep learning models. Finally, a related works review is provided to define the research gap this study attempts to fill.

2.2. Maize

Maize (Zea mays) is one of the most important staple crops worldwide, playing a crucial role in food security and rural livelihoods. Maize is one of the most important cereal crops globally, providing a vital source of food and income for rural communities [5].

Maize was introduced to Ethiopia from Kenya around the 17th century, according to estimations. Currently, it holds the second-largest land cover in the country, following Teff, but it takes the lead as the most important crop in terms of production. In the 2012/13 production season, maize covered 2,526,212 hectares (19.46%) out of a total of 12,979,460 hectares of land dedicated to main crops nationwide. Among the major crops produced in the country, maize accounted for a significant share of 105,570,936 cubic meters (30.88%) out of the total production of 341,828,693 cubic meters. In the maize crop sector, the Amhara region held a substantial share of 23.56% in terms of land area and 24.08% in production, as reported by the Central Statistics Agency's 2012/13 Post-Production Forecast.[1]

[1] “ግብርና ቢሮ የስልጠናው ይዘት 2014”.

Maize holds a crucial position in Ethiopia as a staple food crop and serves as a primary source of calories in major maize-producing regions. It is cultivated across approximately 2.135 million hectares of land in the country, making it one of the most significant cereal crops. The extensive genetic diversity of maize and its versatile applications contribute to its cultivation across various environments in Ethiopia. Maize is Ethiopia’s leading cereal crop in terms of production with 6.2 million tons produced in 2013 by 9.3 million farmers across 2 million hectares of land [6].

In Ethiopia, it is one of the most important strategic crops and ranks second to Teff in area coverage and first in total production. It is primarily grown by Ethiopian farmers for subsistence purposes[7]. Approximately 75% of the total maize output is consumed by farming households, making it a crucial crop for ensuring food security and contributing to the country's economic development. Furthermore, a significant portion, approximately three-fourths, of the maize produced is consumed directly by small-scale producers within their households. Additionally, maize holds the status of being the most widely cultivated staple food crop in sub-Saharan Africa (SSA), with an annual cultivation area exceeding 33 million hectares [1].

Figure 2.1: Maize leaf captured on Kokono Agricultural farm

Ethiopia is the only country in Sub-Saharan Africa to have made significant progress in maize production and input usage. However, due to biotic, abiotic, and socioeconomic restrictions, maize yields have remained low. Diseases are the main biotic factors limiting maize production and productivity. Diseases like common rust, maize lethal necrosis, gray leaf spot, turcicum leaf blight diseases, and pests are becoming very important diseases and pests due to agronomic improvement of maize crops [8].

2.3. Major Maize Diseases and Pests in Ethiopia

Maize, also known as corn, is a crucial staple crop with global significance, providing essential food, feed, and raw materials. However, maize plants are vulnerable to various diseases that can have a significant impact on crop yield and quality. The productivity of maize (Zea mays L.) is predominantly constrained by several factors, including diseases, weeds, and insect pests [9]. It is essential to understand the different types of maize leaf diseases and their potential consequences to effectively manage diseases and ensure optimal productivity. Here are some prevalent maize leaf diseases and their associated impacts:

2.3.1. Northern Leaf Blight (NLB)

One of the significant diseases that negatively impacts maize yields in the tropics and humid tropics, including Ethiopia, is the northern leaf blight (NLB) disease. This disease is caused by the fungus Exserohilum turcicum, and it can inflict substantial losses in maize yields in these regions [9]. NLB is primarily affects maize leaves and it manifests as long, elliptical lesions with tan centers and brown borders. NLB can reduce photosynthetic activity in affected leaves, leading to decreased plant growth and diminished grain yield. Severe infections can cause premature leaf death, further exacerbating yield loss [10].

2.3.2. Gray Leaf Spot (GLS)

The Gray leaf spots caused by Cercospora zeae-maydis fungal pathogen is one of the most significant yield-limiting foliar diseases found in the maize plant. Since its initial report in the 1970s, this pest has emerged as a grave menace to global maize cultivation, causing substantial repercussions in extensive regions of Africa and the Corn Belt in the United States [3]. GLS is caused by the fungus Cercospora zeae-maydis and is wide spread in maize-growing regions. It presents as gray to tan rectangular lesions with dark borders on maize leaves. GLS can reduce photosynthetic capacity and leaf area, resulting in decreased plant vigor and lower grain yield. Severe infections can lead to premature defoliation, impairing the plant's ability to fill grain [11].

2.3.3. Common Rust

Common rust, caused by the fungus Puccinia sorghi, is a widespread disease in maize. It manifests as small, reddish-brown to black pustules on leaves, stems, and husks. While common rust does not typically cause severe yield losses, heavy infections can result in premature leaf senescence and reduced photosynthetic activity, potentially impacting grain fill and overall yield [12].

Common rust disease caused by the fungus Puccinia sorghi, is a significant threat to maize production worldwide. Common rust is the most destructive foliar disease of the crop in the country. Currently, common rust is the most serious danger to maize production in Ethiopia's maize growing areas, causing major yield losses [13].

Common rust disease affects maize leaves, leading to characteristic symptoms such as small reddish-brown pustules on the leaf surface. Common leaf rust (CLR) is a significant disease in Ethiopia and is prevalent in the major maize-growing regions of the country. However, its impact varies across different areas [14].

The researchers states that the "report confirms the widespread occurrence of common rust in Ethiopia, with the fungus Puccinia sorghi being the causal agent. This provides clear evidence of the presence and prevalence of the common rust disease affecting maize crops in Ethiopia. Common rust becomes highly noticeable as maize plants near the tasseling stage. It can be identified by the presence of small, elongated, and powdery pustules on both sides of the leaves, as illustrated in Figure 2.2, initially; the pustules appear dark brown during the early stages of infection. As the plant matures, the epidermis ruptures, and the lesions transition to a black color.

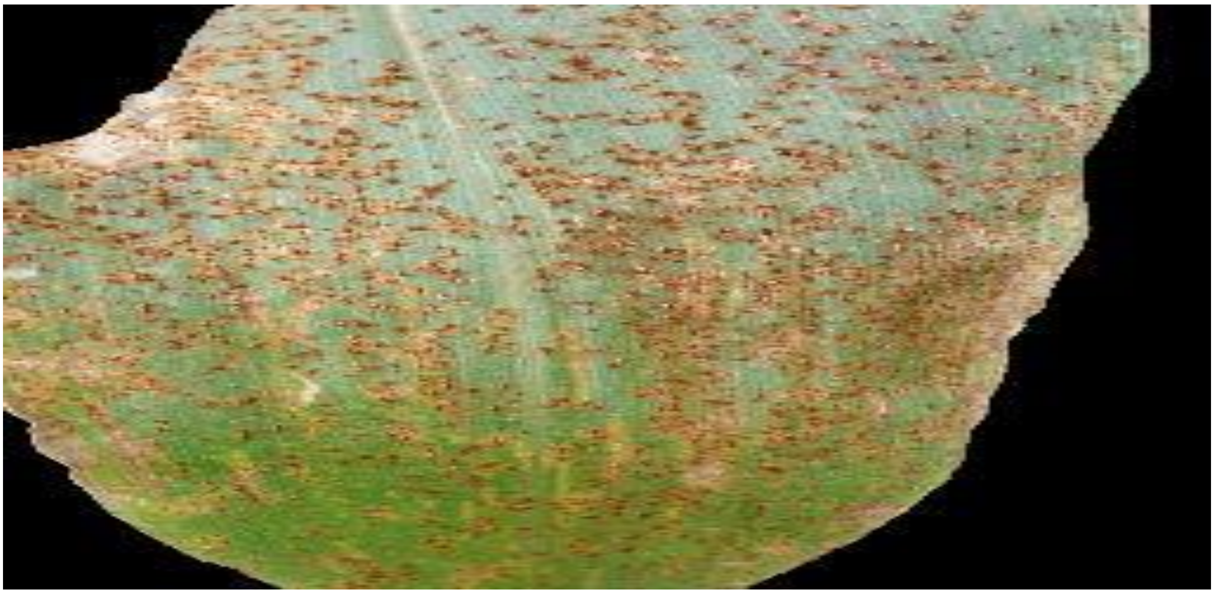


Figure 2.2 : Common Rust Disease

2.3.4. Maize Fall Armyworm

Fall armyworm (FAW) (Spodoptera frugiperda) is another newborn challenge and pandemic to Africa’s crop production .The Fall Armyworm is a migratory insect pest known to cause massive destruction of maize crops under warm and humid conditions in America. The pest was first detected in Africa in 2016 in Nigeria and subsequently in southern Africa. In just one year, the insect moved all the way to East Africa and reached Ethiopia in March 2017; and is now confirmed in more than 30 countries on the continent[15].

The fall armyworm (FAW), Spodoptera frugiperda is known to feed on more than 350 plant species, including those that are important sources of food (e.g., sorghum, rice, and maize), fber (e.g., cotton), and fodder (e.g., grasses). Originating in the Americas, this pest has recently spread to various regions in Africa, Asia, and Australia, and is now on the verge of reaching Europe. It possesses several distinctive characteristics, such as its ability to move swiftly, its ability to feed on a wide range of plants, a highly efficient detoxification system that allows it to develop resistance to insecticides and plant toxins, as well as its adaptable behavior and physiology [16].

Ethiopia is one of the African countries that recently confirmed the presence of the fall armyworm, with the pest being detected in the country in 2017. Since its introduction, Ethiopia has been severely affected by the FAW within a short period of time. As of early

June 2017, the FAW was confirmed in six major maize-producing regions of the country, including the remote rural areas of the Gamo-gofa zone in southwestern Ethiopia, where the international agriculture project Nuru Ethiopia (NE) operates [17].

"The FAW is a tropical species adapted to the warmer climates with average temperatures between 10.9–30°C. Stages of larvae are killed at lower temperatures, while the wings of the adult FAW tend to be deformed at temperatures above 30°C (Simmons, 1993)." This indicates that the climate type of this study area that is best suited for the FAW is a tropical climate, with average temperatures generally falling within the 10.9-30°C range. Regions that experience this type of warmer, tropical climate would provide the most favorable conditions for the FAW to thrive and cause significant damage to maize and other crops. The studies highlight the significant yield losses caused by fall armyworm infestations in Ethiopia, emphasizing the need for "coordinated monitoring and management efforts[17].

Given the widespread nature of these issues, the development of deep learning models for the identification and of both the maize leaf common rust disease and the fall armyworm pest could be highly beneficial for Ethiopia's agricultural sector. The evidence presented in the studies, regarding the prevalence of maize leaf common rust and the fall armyworm pests in Ethiopia, provides a strong justification for the development of deep learning-based identification models to support the country's efforts in managing these significant agricultural challenges.

Figure 2.3: Maize leaf infected by fall armyworm pest

2.4. Overview of Digital Image Processing

Digital image processing involves the manipulation and analysis of 2D functions, referred to as digital images. A digital image, denoted as f(x, y), represents the intensity or gray level at each coordinate (x, y) in a discrete and finite manner. This field utilizes computers to process and analyze digital images [18].

The adoption of digital images began early in industries such as newspapers. In the 1920s, the Bartlane cable picture transmission system revolutionized the process of transmitting images between London and New York. This system significantly reduced the time required to transport images across the Atlantic, from over a week to less than three hours. Specialized printing equipment encoded the pictures to facilitate cable transmission, and they were reconstructed at the receiving end [19].Digital image processing applications typically follow a set of basic steps. These steps include:

2.4.1. Image Pre-processing

Image preprocessing is a technique used to convert unrefined image data into a refined form by eliminating noise and addressing issues such as missing or incomplete values, inconsistent values, and erroneous information commonly present in raw image data [20]. Image pre-processing encompasses low-level techniques aimed at reducing noise, enhancing contrast, and sharpening images. These techniques include image resampling, which involves transforming an image's pixel dimensions, either by down sampling (losing some information) or up sampling (adding new pixels based on existing color values). Image enhancement focuses on bringing out obscured details or highlighting specific features of interest, often through contrast adjustments.

2.4.2. Feature Extraction

Feature extraction is a crucial step in image classification. It involves identifying and extracting the most relevant and informative characteristics [21].Feature extraction is a dimensionality reduction technique that identifies and represents interesting image parts as compact feature vectors. In the context of maize leaf disease and pest identification, features such as color, intensity, and leaf dimensions play a crucial role in classification. Various image preprocessing techniques, such as binarization, Thresholding, and standardization, are applied to obtain the region of interest. Feature extraction techniques, such as color, texture, shape, and Gabor filters, are then applied to capture useful information for image classification and recognition.

Color features are significant for easy visual identification, representing attributes using color moments, fuzzy color moments, or color histograms. Texture features focus on analyzing patterns and repetitions in the image, describing textons that vary in size, shape, color, and orientation. Shape features extract corner information invariant to rotation, translation, and scaling, addressing important problems in computer vision and object recognition.

2.4.3. Classification

Classification involves categorizing images into predefined classes using supervised or unsupervised approaches. Softmax classifiers are commonly used for image classification tasks in the case of maize leaf disease and pest identification, a Sequential model can be utilized to perform supervised classification by training it to classify and distinguish diseases and pests affecting maize leaves from healthy ones.

In summary, for maize leaf disease and pest identification using a Sequential and EfficientNetB0 model, the process involves pre-processing, feature extraction and classification using Softmax classifiers.

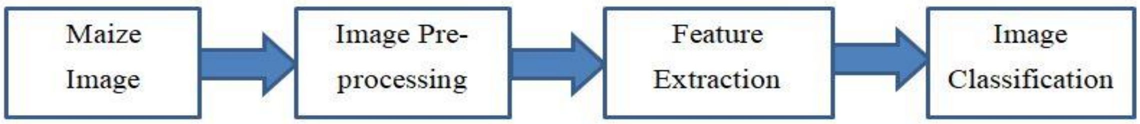


Figure 2.4: Image preprocessing step in digital image processing

2.5. Machine Learning

Arthur Samuel (1959), credited with the early development of machine learning (ML), commonly defined it as a field of study that grants computers the capability to learn. Machine learning primarily pertains to the modifications in systems engaged in tasks related to artificial intelligence (AI) Such tasks involve recognition, diagnosis, planning, robot control, prediction, etc. [22].

Machine learning is a subpart of artificial intelligence (AI) that studies on the development of algorithms and models that enable computers to learn and make predictions or decisions without explicit programming. It involves the construction of mathematical models and algorithms that can automatically learn and improve from experience or data.

Machine learning is a subfield of artificial intelligence that focuses on enabling machines to perform tasks proficiently using intelligent software. The statistical learning methods form the foundation of this intelligent software, which is leveraged to develop machine intelligence. Since machine learning algorithms rely on data to learn and improve, the field has a strong connection with database systems. Closely related concepts include:

- ✓ Knowledge Discovery from Data (KDD): The process of extracting valuable insights and patterns from large datasets.
- ✓ Data Mining: The application of specific algorithms and techniques to discover and extract meaningful information from data.
- ✓ Pattern Recognition: The identification of regularities and structures within data, which can be used to classify, cluster, or make predictions [23].

2.5.1. Types of Machine Learning

Machine learning algorithms can be categorized into three primary groups: supervised learning, unsupervised learning, and reinforcement learning. The machine learning algorithm then learns to recognize the distinct visual patterns and features associated with each disease category. Once trained, the model can be used to classify new, unseen images of maize leaves and predict the type of disease present.

Supervised learning entails training a model using labeled examples, where the desired output is already known. This approach is commonly used in agriculture, where models are trained on labeled data to establish the relationship between input features and target variables. For instance, in crop disease detection, machine learning algorithms can be trained on labeled images of healthy and diseased plants. This enables the algorithms to classify new images and identify potential diseases based on what they have learned.

This supervised learning approach enables the development of intelligent systems that can assist farmers, agronomists, and researchers in the early detection and diagnosis of maize leaf diseases. By accurately classifying the disease, appropriate intervention and management strategies can be implemented to mitigate the impact on crop yields and quality. In the field of machine learning, the two most prevalent supervised learning tasks are classification and regression “classification” that separates the data and “regression” that fits the data. For instance, predicting the class label or sentiment of a piece of text, like a tweet or a product review, i.e., text classification, is an example of supervised learning [24].

Unsupervised learning, on the other hand, aims to uncover patterns or structures in data without the presence of explicit labels. Instead of relying on labeled data, unsupervised learning techniques enable the identification of inherent patterns and structures in unlabeled data. This is particularly useful for tasks like identifying distinct soil types or grouping similar weather patterns for localized forecasting. Unsupervised learning allows for data exploration and clustering analysis, providing insights into the underlying structure of the data [25].

Reinforcement learning involves training an agent to interact with an environment and learn optimal actions through a process of trial and error, guided by rewards and punishments. The agent receives feedback in the form of rewards or penalties based on its actions, which helps it learn and improve over time. Reinforcement learning is applicable in scenarios where an agent needs to make sequential decisions, such as game playing or autonomous robot control. Reinforcement learning is a type of machine learning algorithm that enables software agents and machines to automatically evaluate the optimal behavior in a particular context or environment to improve its efficiency[26].

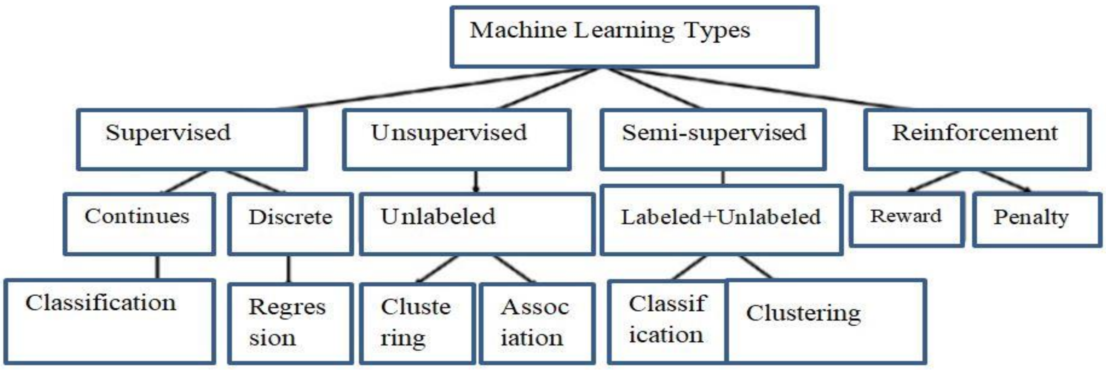


Figure 2.5: Machine Learning Types

In the context of agriculture, machine learning techniques have been employed for a wide range of applications, including crop yield prediction, disease detection, pest monitoring, and soil quality assessment. These techniques utilize large datasets collected from diverse sources, such as remote sensing, sensor networks, and historical records, to train models that can extract meaningful patterns and relationships. It has gained significant attention and revolutionized various fields, including agriculture, by providing powerful tools for data analysis, pattern recognition, and prediction.

One of the key advantages of machine learning is its ability to handle complex and high-dimensional data, enabling the discovery of hidden patterns that may not be apparent through traditional statistical methods. By analyzing vast amounts of data, machine learning algorithms can identify non-linear relationships and make accurate predictions or classifications.

2.6. Deep Learning

Deep learning is a subfield of machine learning that has gained significant attention and achieved remarkable success in various domains, including agriculture. It entails the process of training artificial neural networks with multiple layers to acquire hierarchical representations of data and make precise predictions. In recent years, deep learning has revolutionized agricultural applications, particularly in image analysis and computer vision tasks [27].

Deep learning (DL) is increasingly gaining importance in our daily lives. It has already had a significant impact in fields including disease detection, precision medicine, self-driving cars, predictive forecasting, and speech recognition, among others. Traditional learning, classification, and pattern recognition algorithms require carefully built feature extractors that are not scalable for big data sets [28].

Deep learning addresses the core challenge of representation learning by introducing representations that are expressed in terms of other, simpler representations. Deep learning allows the computer to build complex concepts out of simpler concepts [29]. One of the key advantages of deep learning is its ability to automatically learn complex features from raw data, eliminating the need for manual feature engineering.

Table 2.1: Comparing the differences between machine learning and deep learning

Characteristic	Machine Learning	Deep Learning
Feature Extraction	Requires manual feature engineering by domain experts to extract relevant characteristics from leaf images (e.g. color, texture, shape)	Automatically learns discriminative features from the raw leaf image data through the neural network layers
Model Complexity	Uses relatively simple models like Support Vector Machines, Random Forests, etc.	Employs complex, multi-layer convolutional neural network architectures
Training Data Requirement	Works reasonably well with smaller datasets (e.g. hundreds to thousands of labeled leaf images)	Requires large datasets (e.g. thousands to millions of labeled leaf images) to achieve high performance
Training Time	Faster training times, especially for small datasets	Much longer training times, especially for large datasets and complex models
Inference Speed	Can provide real-time inference on leaf images	May require specialized hardware (e.g. GPUs) for efficient real-time inference
Generalization Ability	Model performance may degrade when applied to new leaf varieties or environmental conditions not seen in the training data	Can better generalize to new leaf varieties and environmental conditions if trained on sufficiently diverse datasets
Interpretability	Models are more interpretable, with the ability to explain which features are important for disease classification	Models are often treated as "black boxes", making it difficult to explain the reasoning behind predictions
Applications	Suitable for small-scale, well-defined plant leaf disease classification tasks	complex plant leaf disease classification tasks

Traditional machine learning approaches can be effective for plant leaf disease classification with smaller, well-curated datasets, deep learning models have the ability to automatically learn powerful visual features from raw leaf image data and achieve superior performance on large-scale, real-world plant disease classification tasks. However, deep learning models require significantly more training data and computational resources compared to classical machine learning methods.

2.6.1. Convolutional Neural Networks

Convolutional neural networks (CNNs) are a popular class of deep learning models that excel in image analysis tasks. CNNs utilize convolutional layers to extract spatial features from images and pool them to capture increasingly higher-level representations. This hierarchical feature extraction enables CNNs to detect intricate patterns and make accurate predictions, making them highly effective in tasks such as plant disease detection, pest identification, and weed classification [30].

Convolutional neural networks (CNNs) are a widely used class of deep learning models that have revolutionized image analysis and computer vision tasks in agriculture and other domains. CNNs are specifically designed to process grid-like data, such as images, by exploiting the spatial relationships between pixels.

Convolutional Neural Network (CNN), also known as ConvNet, is a type of Artificial Neural Network (ANN) with a deep feed-forward architecture and amazing generalizing ability when compared to other networks with FC layers. It can learn highly abstracted features of objects, especially spatial data, and can identify them more efficiently than other networks with FC layers [31].

A CNN is made up of several convolution and sub-sampling layers, as well as a fully linked layer and a normalization layer. Moving from input to output layers, the series of successive convolution layers performs progressively more sophisticated feature extraction at each layer. Following the convolution layers are fully connected layers that do categorization. Between each convolution layer, sub-sampling or pooling layers are frequently used. A 2D n x n pixelated image is fed into a CNN. Filters or kernels are groupings of 2D neurons that make up each layer [31].

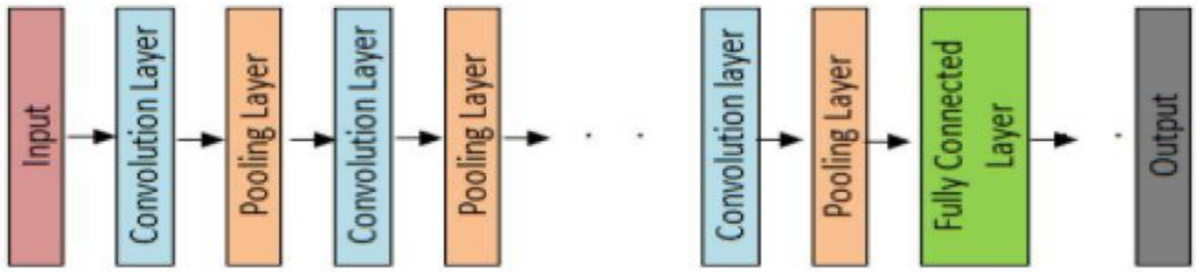


Figure 2.6: Conceptual model of CNN[31]

A convolutional neural network is a specific type of neural network with many layers. It processes data that has a grid-like arrangement then extract features. One huge advantage of using CNNs is that you don't need to do a lot of pre-processing on images[32]. The basic functionality of CNN can be broken down into four key areas.

- ✓ The input layer should be configured to accommodate the size and dimensions of the maize leaf images in the dataset. Resizing the images to a consistent size is often necessary to ensure uniformity.
- ✓ The convolutional layer: Convolutional layers are the core building blocks of a CNN. They extract features from the input images. The number of convolutional layers and their depth depend on the complexity of the disease identification task and the size of the dataset. Deeper networks can capture more intricate patterns but may be prone to overfitting if the dataset is small.
- ✓ The pooling layer: Pooling layers help reduce the spatial dimensions of the feature maps while retaining important features. Pooling operations commonly employed in image processing include max pooling and average pooling. These operations involve taking the maximum or average value within specific regions of an image, respectively. Pooling can be performed after each convolutional layer to down sample the feature maps.
- ✓ The fully-connected layer: After the convolutional and pooling layers, fully connected layers can be added to perform classification tasks. These layers take the extracted features and map them to the desired number of output classes, corresponding to different maize leaf diseases or healthy states [33].

2.6.2. Major CNN architectures

Convolutional Neural Networks, often called CNNs, are a specific type of neural network structure created to handle data organized in a grid-like format. This unique design makes them highly effective in processing spatial and temporal data, such as images and videos, where neighboring elements exhibit strong correlations. While CNNs share similarities with other neural networks, they possess an additional layer of intricacy due to their utilization of multiple convolutional layers [34].

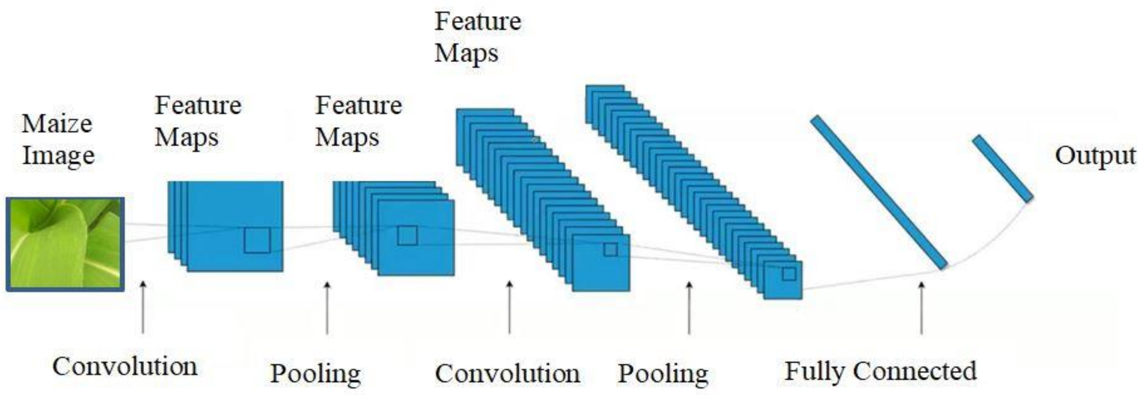


Figure 2.7: Architecture of CNN Model

The choice of the best CNN architecture for plant leaf disease identification depends on various factors, including the size and diversity of the dataset, available computational resources, and desired trade-offs between accuracy and computational efficiency. Here are a few CNN architectures that have been successfully applied to plant leaf disease identification:

- ✓ AlexNet: AlexNet was one of the pioneering CNN architectures that gained prominence after winning the ILSVRC 2012 competition. It introduced the concept of deep convolutional neural networks and popularized the use of rectified linear units (ReLU) as activation functions.
- ✓ VGGNet: VGGNet, including VGG-16 and VGG-19, was proposed by the Visual Geometry Group at the University of Oxford. It is known for its simplicity and uniform architecture, with small 3x3 convolutional filters stacked on top of each other.
- ✓ GoogLeNet (Inception-v1): GoogLeNet, also known as Inception-v1, introduced the inception module concept, which consists of parallel convolutional operations with different filter sizes. It aimed to reduce the number of parameters and computational complexity.
- ✓ ResNet: ResNet, short for Residual Network, introduced residual connections to address the vanishing gradient problem. It allowed the network to be much deeper by using skip connections that bypassed one or more layers.
- ✓ DenseNet: DenseNet proposed a densely connected architecture in which each layer was connected to every other layer in a feed-forward fashion. It promoted feature reuse and alleviated the vanishing gradient problem.

- ✓ MobileNet: MobileNet focused on designing lightweight CNN architectures suitable for mobile and embedded devices. It utilized depth-wise separable convolutions to reduce computational complexity while maintaining high accuracy.
- ✓ EfficientNet: EfficientNet aimed to achieve better trade-offs between accuracy and computational efficiency by scaling the network in terms of depth, width, and resolution. It used a compound scaling method to balance these factors.
- ✓ SqueezeNet: SqueezeNet aimed to reduce the number of parameters in the network while maintaining high accuracy.
- ✓ Xception: Xception, derived from "Extreme Inception," used depth-wise separable convolutions to replace the standard convolutional layers in the Inception module. It aimed to capture both spatial and channel-wise dependencies.

The aforementioned examples are instances of convolutional neural network (CNN) architectures. Many more architectures have been proposed, each with its own specific contributions and variations to address different challenges in computer vision tasks [35].

2.6.3. Sequential model

The Sequential model is a popular deep learning architecture commonly used for various tasks, including image classification. In the context of maize leaf disease and pest identification, the Sequential model can be employed to develop an effective classification model. The Sequential model is characterized by a linear stack of layers, where each layer performs specific operations on the input data. In the context of image classification, convolutional layers are commonly used for feature extraction, followed by fully connected layers for classification. Here is an example of how a Sequential model can be utilized for maize leaf disease and pest identification:

- ✓ Input Layer: The first layer in the Sequential model is the input layer, which receives the input data, i.e., the maize leaf images. The shape of the input layer is determined by the dimensions of the input images.
- ✓ Convolutional Layers: Convolutional layers are used for extracting important features from the input images. These layers consist of multiple filters that convolve over the input, detecting patterns and extracting features at different spatial scales. Each

convolutional layer is typically followed by an activation function, such as ReLU, to introduce non-linearity.

- ✓ Pooling Layers: Pooling layers play a crucial role in extracting feature maps produced by convolutional layers. Their purpose is to decrease the spatial dimensions of the feature maps while preserving the essential features. Max pooling and average pooling are prevalent techniques employed for this purpose.
- ✓ Flatten Layer: After the convolutional and pooling layers, a flatten layer is used to convert the 2D feature maps into a 1D vector, preparing them for input to the fully connected layers.
- ✓ Fully Connected Layers: Fully connected layers receive the flattened feature vectors and perform the classification task. These layers consist of nodes or neurons that are connected to all the nodes in the previous layer. Activation functions, such as ReLU, are applied to introduce non-linearity in these layers.
- ✓ Output Layer: The final layer in the Sequential model is the output layer. For maize leaf disease and pest identification, the output layer typically contains nodes corresponding to different disease or pest classes. The activation function used in this layer depends on the specific task, such as softmax for multi-class classification.

2.6.4. EfficientNetB0 Model

The EfficientNetB0 model is a well-suited deep learning architecture for the task of maize leaf disease and pest classification. EfficientNetB0 is a compact and efficient convolutional neural network (CNN) model that has demonstrated strong performance in various image classification tasks. It is part of the EfficientNet family of models, which are designed to achieve high accuracy while maintaining a small model size and low computational complexity. The EfficientNetB0 model has around 5.3 million parameters, making it a lightweight and efficient choice for deployment on edge devices or resource-constrained environments EfficientNet B0, which serves as the foundational model within the EfficientNet family, consists of a diverse set of layers. Each of these layers plays a vital role in ensuring the model's efficiency and effectiveness in accomplishing image classification tasks.

Figure 2.8: EfficientNet Model Architecture

2.6.5. Model Evaluation Metrics

Accuracy alone cannot identify whether a model is good or terrible, but accuracy combined with precision, recall, and F1 Score can provide a good understanding of the model's performance [36]. Various evaluation metrics such as accuracy, precision, recall, and F1-score can be calculated to measure the model's effectiveness in disease and pest identification.

Precision: precision is the ability of classifier not to label an instance positive that is actually negative. The precision of each class is determined by dividing the number of true positives by the sum of true positives and false positives.

$precision = \frac{True\ Positive}{Actucal\ result}$ Or $precision \ = \frac{True\ positive}{True\ positive\ +Falsepostive}$

Recall: - is the ability of classifier to find all positive instances. For each class it is defined as the ratio of true positive to the sum of true positives and false negative

$Recall = \frac{True\ positive}{Predicted\ result}$ Or $Recall = \frac{True\ positive}{True\ positive+\ False\ negative}$

F1 score: - The F1 score is a metric that captures the weighted harmonic mean of precision and recall, with a perfect score of 1.0 and the worst score of 0.0. F1 scores tend to be lower than accuracy measures as they incorporate both precision and recall in their calculation. When comparing different classifier models, it is advised to use the weighted average of the F1 score instead of relying solely on overall accuracy as a more reliable guideline. $f1\ score = \frac{2*(Recall*percisison\ )}{(Recall+percisison\ )}$

Confusion matrix: The confusion matrix provides insights into the predictions made by a model in comparison to the actual classes of the maize leaf samples. It is a matrix of size N x N, where N represents the number of target classes, and it serves as a valuable tool for evaluating the performance of a classification model. By visualizing the confusion matrix, an individual could determine the accuracy of the model by observing the diagonal values for measuring the number of accurate classification [37].

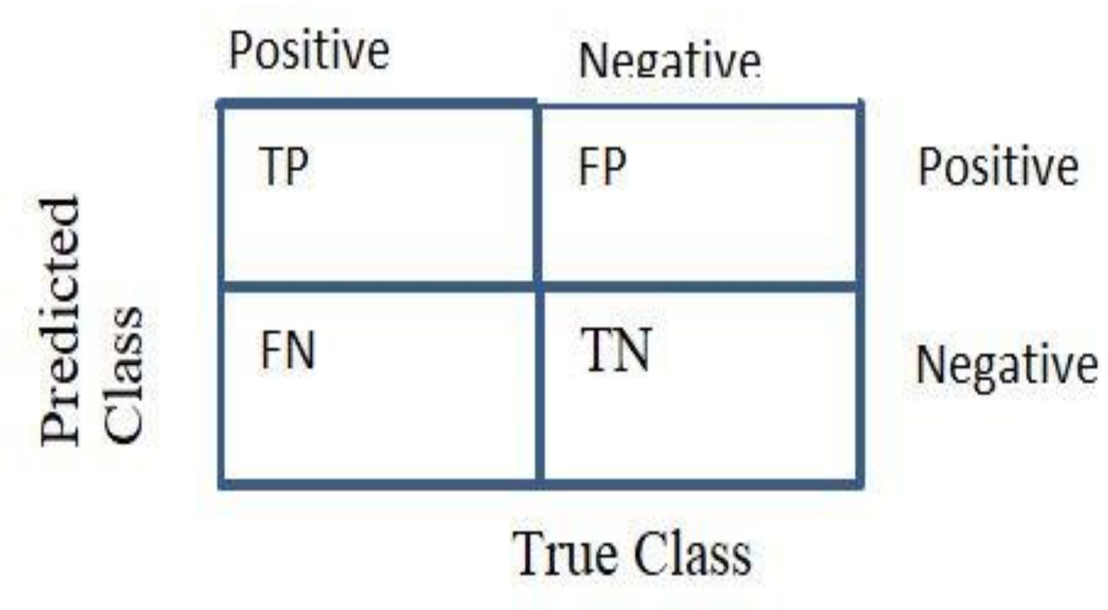


Figure 2.9: confusion matrix is in the form of a square matrix

TP (True Positive): The model correctly identified the presence of a disease or pest on the maize leaf.

FP (False Positive): The model incorrectly identified the presence of a disease or pest on the maize leaf when it was actually healthy.

FN (False Negative): The model failed to identify the presence of a disease or pest on the maize leaf when it was actually present.

TN (True Negative): The model correctly identified the absence of a disease or pest on the maize leaf [38].

2.7. Related Works

Several international studies have focused on leveraging deep learning and machine learning models for disease identification in various crops, including maize and coffee. These studies utilize techniques such as Convolutional Neural Networks (CNNs) to accurately detect and classify diseases based on visual symptoms and patterns in leaf images.

In Ethiopian contexts, researchers focus on disease recognition and classification for different crops including maize. For maize leaf diseases, they utilize support vector machine models and image processing techniques, achieving an average accuracy of 95.5% by combining texture, color, and morphology features [4].

In another research paper, the authors propose a maize leaf disease detection and classification approach using supervised machine learning algorithms. They compare techniques like Naive Bayes, Decision Tree, K-Nearest Neighbor, Support Vector Machine, and Random Forest. The Random Forest algorithm achieves the highest accuracy of 79.23%, although it falls short compared to other deep learning models [39].

In another study, the authors focus on maize leaf disease classification using CNNs and hyperparameter optimization. They demonstrate that the accuracy for maize leaf disease classification reaches an impressive 97% across all evaluated CNN models [40].

For the identification of Maydis leaf blight (MLB) disease in maize crops, researchers develop a deep CNN model using the architecture of GoogleNet. This model achieves an exceptional overall accuracy of 99.14% on a separate test dataset comprising 596 healthy and 951 infected maize leaf images [41].

In the context of maize common rust disease severity prediction, authors propose an approach that combines threshold segmentation, fuzzy decision rules, and VGG-16 network classification. The VGG-16 network achieves a validation accuracy of 95.63% and a testing accuracy of 89% when classifying images into severity classes [42].

In coffee leaf disease identification, the authors compare filtering techniques and apply KMeans clustering for segmentation. Their proposed model with an SVM classifier achieves an overall classification accuracy of 96.5% [43].

Another research paper introduces, in chickpea disease identification, researchers combine CNN and LSTM models for feature extraction and Softmax for classification. The proposed CNN-LSTM model achieves an accuracy of 92.55% in identifying chickpea diseases. The study highlights the superior performance of the CNN-LSTM model compared to existing methods [44].

A novel classification model for maize leaf diseases, including blight, common rust, gray leaf spot, and healthy. The approach utilizes DenseNet201; a deep-learning architecture specialized for image classification tasks. The model achieves an outstanding classification accuracy of 94.6% by combining deep features with fine-tuned Support Vector Machine (SVM) using Bayesian optimization techniques [45].

Lastly, Researchers have introduced an enhanced model based on ResNet50 to improve the identification of pests and diseases affecting maize. By incorporating additional effective channels into the residual network module, the model has achieved an impressive recognition accuracy of 96.02%. This study has successfully identified a range of maize pests and diseases, including but not limited to maize leaf blight, gray leaf spot, rust disease, stem borer, and corn armyworm [46].

In summary, these related works demonstrate significant advancements in disease identification for maize leafs and other crops using deep learning models. These approaches leverage the power of CNNs, optimization techniques, and innovative methodologies to achieve high accuracy in disease detection and classification.

Our proposed the development of a deep learning model using the sequential and EfficientNetB0 architecture for maize leaf disease and pest identification, this research aims to bridge the existing research gaps in the field. By focusing on the local context, incorporating a representative dataset, and considering practical implementation, the research strives to provide a valuable tool for farmers and agricultural experts in Ethiopia. The expected outcome is an accurate and efficient system for early detection and management of common rust disease and fall armyworm pests in maize crops, contributing to the advancement of agricultural practices in the region.

Table 2.2: summary related works

Author	Title	Methodology	Accuracy
Enquhone Alehegn 2019	Ethiopian maize diseases recognition and classification using support vector machine	Using both SVM and image processing. Based on the experiment result using combined (texture, colour and morphology)	95.5 %
Himansu Das(2020)	developed Maize Leaf Disease Detection and Classification Using Machine Learning Algorithms	supervised machine learning techniques such as NB,DT,KNN, SVM, and RF for maize plant disease detection with the help of the images of the plant	79.23%
Erik Lucas da Rocha(2020)	Maize leaf disease classification using convolutional neural networks and hyperparameter optimization	We apply enhancement methods such as Bayesian hyperparameter optimization, data augmentation, and fine-tuning strategies. We evaluate these CNNs on the maize leaf images from Plant Village dataset, and all experiments were validated using a five-fold cross-validation procedure over the training and test sets.	97%
Md Ashraful(2021)	Image-based identification of maydis leaf blight disease using deep learning	GoogleNet has been used to build the deep CNN model.	99.14%
Malusi Sibiya 2021	Automatic Fuzzy Logic-Based Maize Common Rust	applying threshold-segmentation on images of diseased maize leaves a VGG-16	Vaccuracy 95.63% Taccuracy of

	Disease Severity Predictions with Thresholding and Deep Learning		89%
Linigerew Mengstie Shita(2021)	Ethiopian coffee leaf diseases identification using deep learning features	GF, MF and hybrid of GF&MF CNN-SVM and CNN-softmax classifier	96.5 %
Abebech Jenber Belay(2022)	developed deep learning chickpea disease identification by combining CNN and LSTM	combining Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) Hybrid of GF&MF noise filtering technique	92.55%
Arabinda Dash(2023)	Maize disease identification based on optimized support vector machine using deep feature of DenseNet201	optimized support vector machine using deep feature of DenseNet201	94.6%
Wenqing (2023)	Enhancing Corn Pest and Disease Recognition through Deep Learning: A Comprehensive Analysis	introduced additional effective channels (environment–cognition–action) into the residual network ResNet 50 Model.	96.02 %

CHAPTER THREE

3. Research Methodology

3.1. Materials and Methods

In this study, we used experimental research to develop a deep-learning model to identify and classify Maize leaf diseases and Pests. The proposed research approach used a combination of Gaussian and median noise filtering techniques to enhance the dataset quality and train EfficientNetB0 and sequential models. We used a Samsung A12 smartphone camera to capture images from the maize crop. When images were taken, the camera was fixed on a stand which reduced the movement of the hand and captured uniform images of maize leaf. To obtain uniform lightning or balanced illumination we used full mobile brightness. Whenever we capture images of maize we turn on the brightness of mobile resolution to get minimal noises of maize leaf images. The photos were taken at a resolution of 4000x3000 pixels and finally reduced to 224 X 224 pixels these are the standard images that can be used in image processing for EfficientNetB0 and sequential models. This allows deep learning models to achieve better performance and accuracy, even with large datasets or limited datasets.

3.2. Dataset Collection

To gather the necessary images (including maize diseases, pests, and healthy samples) for experimentation, an initial attempt was made to obtain data from the Sirinka Agriculture Research Center Bureau. However, the data obtained from the center proved to be insufficient as they only possessed a limited number of disease images primarily intended for demonstration purposes. Since there is no well-organized database of maize diseases and pest in public image repositories, images were captured for both diseased/infected and healthy images directly from some selected Sirinka agriculture research center sites, Maize growing coverage areas in Habru Zuria kebeles, Woldia university Mersa Agricultural campus and in public image repositories.

Maize growing coverage areas in Habru Zuria kebeles were selected, such as: Kokono, Mersa Meda and Anto from three different categories. These categories are; healthy images, images with Common Rust, and images with infected by fall armyworm pest. The images dataset were taken using mobile-phone, specifically Samsung A12 camera full HD and 4000 x3000 pixel resolution was used to acquire the images in the JPG (Joint Photographic Group) file format. After the diseased and healthy images were captured, the Sirinka agriculture research center bureau experts and Habru woreda Agricultural office experts helped to classify the images correctly as Common Rust, infected by fall armyworm pest and Healthy class.

Only 1082 images were capturing the Maize leaf images, we settle the smart-phone on a stand to diminish hand development and make a difference to capture uniform. All images were captured in the identical lighting conditions were used for all images. Figure 3.1 presents the sample Maize leaf image collected from on site for healthy, common rust and infected by fall armyworm pest images.

Figure 3.1: Sample images for Healthy, Fall Armyworm and Common leaf rust

3.3. Proposed Model Architecture

The proposed model architecture for identifying Maize diseases and Pests comprises the following phases: preprocessing, feature extraction, classification (identification), model design, training, validation, and testing phase. The system's complete architecture is depicted in Figure 11.

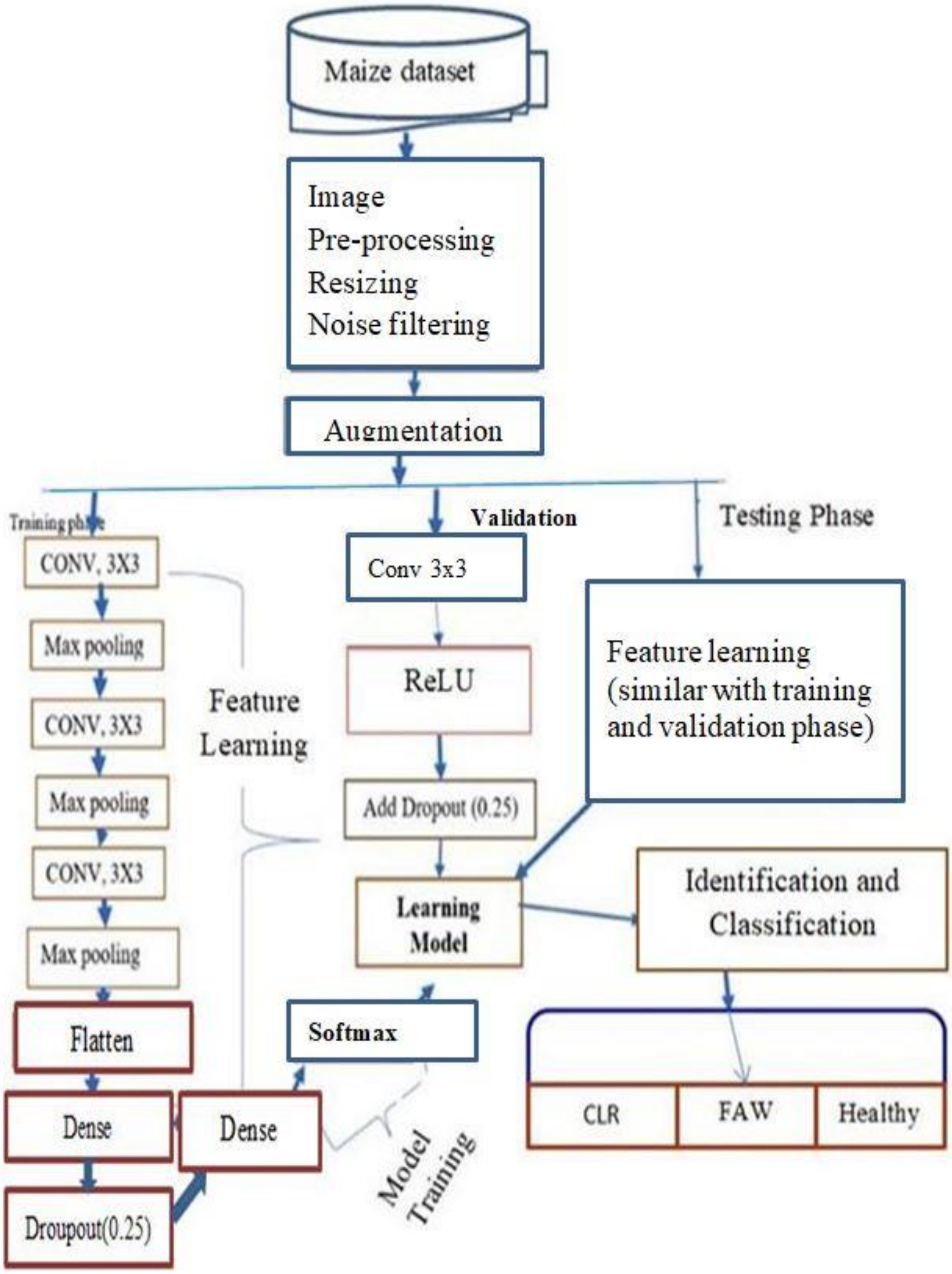


Figure 3.2: Proposed model Architecture (Diagram)

3.3.1. Image Preprocessing

Since the maize diseased and pest images are collected from the harvesting areas at different conditions, some noises may be found in the image that degrade the image quality as well as the recognition ability of the model. To minimize the effect of noise, image preprocessing tasks such as image resizing, Pixel normalization, noise removal, and augmentation involved the images and adjusting their color channels to enhance the quality and comparability of the dataset before feeding it to the CNN model to improve the prediction ability of the model.

3.3.1.1. Image Resizing

The images dataset was taken using a mobile phone, specifically, a Samsung A12 camera full HD and 3000x4000 pixel resolution was used to acquire the images captured from onsite and online after the diseased and healthy images were captured, In the preprocessing phase, the images of maize leaves are normalized to a standard size of 224x224 pixels. This step ensures consistency in the input data and facilitates further processing.

3.3.1.2. Pixel normalization

In many cases, image data consists of pixel values represented as integers ranging from 0 to 255. However, the presence of large integer values in the input image can impede the learning process of deep learning models. Consequently, it becomes necessary to normalize the pixel values of the image, scaling them down to a range between 0 and 1.

3.3.1.3. Noise Removal

The process of Removed noises from the maize leaf images to enhance the dataset quality. Different kinds of noises exist in an image such as camera flash, change in brightness, and noisy background. There are varieties of noise reduction techniques that are available to reduce the noise. Different types of noises are found in images some are impulse noise, additive noise, Gaussian noise, and multiplicative noise. Various techniques for reducing noise can be employed, including median filtering, mean filtering, Gaussian filtering, and more. In this study, we focus on Gaussian filtering, Median filtering, and a hybrid approach combining Gaussian and Median filtering. We utilize the irfanview64 offline application to train a sequential model and compare the performance of these three noise removal techniques by adjusting different hyperparameters.

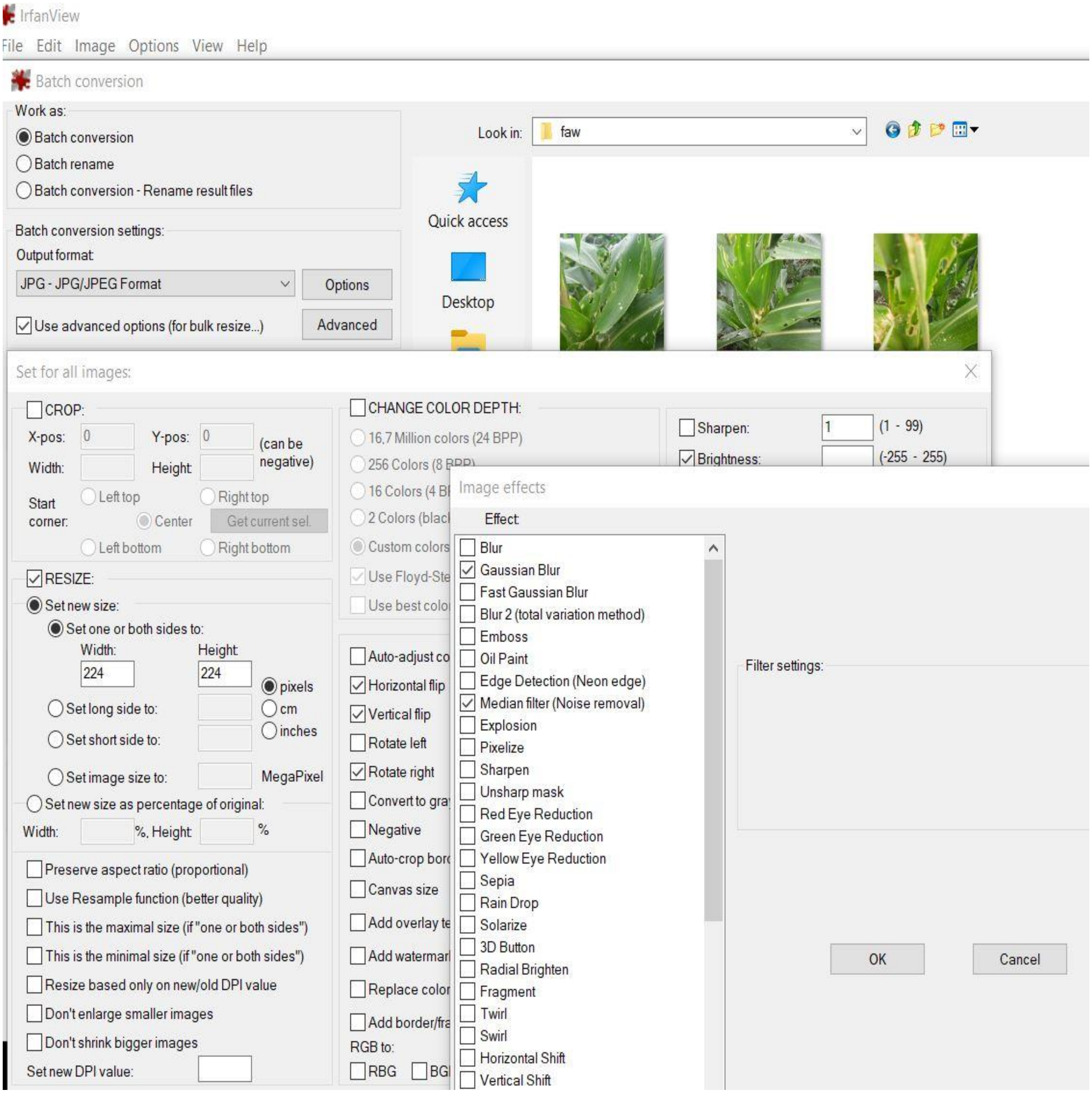


Figure 3.3: Noise filtering and image preprocessing techniques using irfanview 64

3.3.1.4. Augmentation

A large number of images were required for each class to train the deep learning model aimed at increasing the performance of the model. The main goal of augmentation is to expand the image/dataset quantity and introduce subtle distortions. This technique effectively addresses the issue of overfitting during the training phase. This image transformation is also used when testing the model on various rotated images in different degrees, horizontal flip, vertical flip, rotate left, and rotate right which improves the model's testing performance.

Table 3.1: Image dataset construction acquired by online, onsite and augmentation

Class	Image format	Image data source			
		Online	On site	Augmentation	After Augmentation
Healthy	JPG	0	530	757	1287
Common Rust	JPG	530	80	696	1306
fall armyworm	JPG	0	653	653	1306
Total	JPG	530	1263	2106	3899

As depicted in Table 3.1, the distribution of image datasets across three distinct categories. A total of 3899 image datasets were included in the analysis. Among these, 530 images originated from the Kaggle and plant image dataset, 1263 were captured on-site using an A12 Samsung Galaxy device, and the remaining 2106 datasets were obtained through data augmentation using irfanview64 software. Following the data augmentation process, the dataset size was ultimately expanded to 3,899 images.

3.3.2. Feature Extraction

The third phase of the proposed model architecture is feature extraction. We used feature extraction to extract high-level features from the maize images and stored these features as a feature set in the database.

3.3.3. Model Description

The model consists of various layers, including convolutional layers, pooling layers, flattening layer, dense layers, and a dropout layer.

Convolution Layer: The convolution layer applies a filter to the input image, inspecting small windows of pixels at a time. It extracts features from the image and helps the model learn specific image characteristics. In this research, four convolution layers, four max pooling layers and two dense layers were used in the training phase.

In the research, the pooling layer was used to decrease the spatial size of the representation, resulting in a reduction in parameters and computational requirements. Max pooling, which preserves crucial features, was chosen for this purpose. The researchers employed four max pooling layers with a window size of (3x3) and applied the "same" padding technique to maintain the spatial dimensions of the input.

- ✓ Conv2D: The first convolutional layer with 32 filters, a filter size of (3, 3), and ReLU activation. It takes input with the specified input shape.
- ✓ MaxPooling2D: The first max pooling layer with a pool size of (2, 2), reducing spatial dimensions.
- ✓ Conv2D: The second convolutional layer with 64 filters and a filter size of (3, 3), followed by ReLU activation.
- ✓ MaxPooling2D: The second max pooling layer with a pool size of (2, 2).
- ✓ Conv2D: The third convolutional layer with 64 filters and a filter size of (3, 3), followed by ReLU activation.
- ✓ MaxPooling2D: The third max pooling layer with a pool size of (2, 2).
- ✓ Conv2D: The fourth convolutional layer with 64 filters and a filter size of (3, 3), followed by ReLU activation.
- ✓ MaxPooling2D: The fourth max pooling layer with a pool size of (2, 2).
- ✓ Flattening: The flattening layer transforms the output of the convolution layers into a vector, preparing it for input into the fully connected layers.
- ✓ Dense Layer: The dense layer, also known as the fully connected layer, receives the output from the convolution layers and generates predictions. Softmax activation was used for classifying the output classes.
- ✓ Dropout Layer: The dropout layer randomly sets input units to zero during training to prevent overfitting. A dropout rate of 0.25 was applied in this research.
- ✓ Epoch and Batch Size: An epoch represents one pass through the entire training dataset, while the batch size refers to the number of samples considered before updating the model weights. These parameters control the training process.
- ✓ Class Model: The model used in this research was a categorical sequential model. Functions and code snippets were provided for partitioning the dataset, preprocessing the images, and building the CNN model for image classification.
- ✓ get_tf_dataset_partitions: This function partitions a Tensorflow dataset into training, validation, and test sets based on specified split ratios.

3.3.4. Model Building

The model architecture consists of the following layers:

resize_and_scale: Resizes and scales the input images.

data_augmentation: Applies data augmentation to the images.

Convolutional layers: layers.Conv2D: Four convolutional layers with increasing numbers of filters (32, 64, 64, 64) and a filter size of (3, 3). They use ReLU activation.

layers.MaxPooling2D: Four max pooling layers with a pool size of (2, 2) to reduce spatial dimensions.

layers.Flatten: Flattens the output from the previous layer.

Fully connected layers: layers.Dense: A dense layer with 64 units and ReLU activation.

layers.Dense: The output layer with n_classes units and softmax activation, producing class probabilities.

Table 3.2: Building a sequential model for maize leaf disease and pest identification

Model: "sequential_2"		
Layer (type)	Output Shape	Param #
sequential (Sequential)	(32, 224, 224, 3)	0
sequential_1(Sequential)	(32, 224, 224, 3)	0
conv2d (Conv2D)	(32, 222, 222, 32)	896
max_pooling2d(MaxPooling2D)	(32, 111, 111, 32)	0
conv2d_1 (Conv2D)	(32, 109, 109, 64)	18,496
max_pooling2d_1(MaxPooling2D)	(32, 54, 54, 64)	0
conv2d_2 (Conv2D)	(32, 52, 52, 64)	36,928
max_pooling2d_2(MaxPooling2D)	(32, 26, 26, 64)	0
conv2d_3 (Conv2D)	(32, 24, 24, 64)	36,928
max_pooling2d_3(MaxPooling2D)	(32, 12, 12, 64)	0
flatten (Flatten)	(32, 9216)	0
dense (Dense)	(32, 64)	589,888
dropout (Dropout)	(32, 64)	0
dense_1 (Dense)	(32, 3)	195
Total params: 683,331 (2.61 MB)		
Trainable params: 683,331 (2.61 MB)		
Non-trainable params: 0 (0.00 B)		

Table 3.3: Building an EfficientNetB0 model for maize leaf disease and pest identification

Layer (type)	Output Shape	Param #
sequential (Sequential)	(32, 224, 224, 3)	0
sequential_1(Sequential)	(32, 224, 224, 3)	0
efficientnetb0(Functional)	(32, 7, 7, 1280)	4,049,564
global_average_pooling2d (GlobalAveragePooling2D)	(32, 1280)	0
dense (Dense)	(32, 64)	81,984
dropout (Dropout)	(32, 64)	0
dense_1 (Dense)	(32, 3)	195
Total params: 4,131,743 (15.76 MB)		
Trainable params: 4,089,727 (15.60 MB)		
Non-trainable params: 42,016 (164.12 KB)		

3.3.5. Training the model

The dataset was split into training, validation, and testing sets using a predefined ratio. The training set was used to optimize the model's parameters through backpropagation and gradient descent. During training, the model's performance was evaluated on the validation set to monitor its generalization ability and prevent overfitting. The training process was repeated for a predefined number of epochs, and the model with the best validation performance was saved. Finally, the model's accuracy and loss were evaluated on the independent testing set to assess its performance on unseen data.

The provided code is a Python script that sets up a deep learning model for classifying maize leaf diseases and pests. Here's a breakdown of the code and its purpose:

Dataset Partitioning: The get_tf_dataset_partitions () function takes a Tensorflow dataset ds and splits it into training, validation, and test sets based on the specified train_split, val_split, and test_split ratios. The dataset is optionally shuffled before the split, with a shuffle_size parameter controlling the shuffle buffer size.

Image Preprocessing: The code defines an IMAGE_SIZE of (224, 224) and creates a resize_and_scale sequential model that resizes the input images to the specified size and scales the pixel values between 0 and 1.

Data Augmentation: The data_augmentation sequential model is defined, which applies random horizontal and vertical flipping, as well as random rotation of the input images. This helps to improve the model's generalization by artificially expanding the training dataset.

3.3.6. Classification

After feature extraction, the classification phase was built using softmax classifier. The experimental results showed that the proposed CNN feature extraction model. A deep learning model based on convolutional neural networks (CNNs) was designed for disease and pest identification. The architecture consisted of four convolutional and pooling layers, followed by fully connected layers for classification. Dropout and batch normalization techniques were incorporated to improve generalization and alleviate overfitting.

For the classification phase, a convolutional neural network (CNN) is utilized. CNNs are well-suited for image-based classification tasks due to their ability to learn hierarchical features directly from the raw image data. The architecture of the CNN may comprise multiple convolutional layers for feature extraction, followed by pooling layers for spatial down-sampling. Fully connected layers are employed for classification, and activation functions such as ReLU (Rectified Linear Unit) were used to introduce non-linearity. Dropout regularization technique 0.25 also applied to prevent overfitting.

Confusion matrix was used to classify images into their class by using the test set image (unseen or untrained images). Categorizing the images into three classes such as common rust, fall armyworm and healthy. During data collection process the experts grouped each image into their disease class. Based on this data the researcher tries to classify the disease class using sequential model.

CHAPTER FOUR

4. Experiment and Results of Discussion

As part of the experiment result and discussion, the specification used in the development of the Maize leaf disease and pest identification Followed by the detail of experimental result and discussion under below in details of the experiment including the outcomes of each experiment and the discussions of these results are described in this section. The experimental results are visually presented using figures and tables.

4.1. Experimental setup and setting

The experiment was conducted on a PC with an Intel Core i5-6300U CPU, 8GB RAM, and Windows 10 Pro operating system. Various tools were utilized for the development of classification models, including Python, TensorFlow, and Keras. TensorFlow, an open-source machine learning platform, provided the necessary tools, libraries, and community resources for the experiment. Keras, which integrates with TensorFlow, offers a user-friendly API for model development. Python, a high-level programming language, was used with the Anaconda Jupyter Notebook as the coding environment.

4.1.1. Hyper-Parameter Setting

In this section, before the training process, several experiments were conducted with different hyperparameter settings to select the optimal configuration for the model. The hyperparameters selected for the model are as follows:

- ✓ Dataset Ratio: The best results were obtained using 80% training, 10% validation, and 10% testing split.
- ✓ Learning Rate: A learning rate of 0.001 was found to be the most effective.
- ✓ Activation Function: The ReLU activation function in the output layer performed better than alternatives like Tanh and Softmax for the classification problem.
- ✓ Batch Size: A batch size of 32 was used, as it provided a good balance between computational time and performance.
- ✓ Epochs: The sequential model performed optimally at 25 epochs, while the EfficientNetB0 model achieved the best results at 10 and 50 epochs.

- ✓ Optimization Algorithm: The Adaptive Moment Estimation (Adam) optimizer was used to update the model's weights and tune the parameters.

The hyperparameters used during the training of the CNN model are presented in Table 6 and the results presented in Appendeix3.

Table 4.1: Different Hyperparameters

Hyper parameters	First	Second	Third
Batch size	32	64	32
Learning rate	0.01	0.0001	0.001
Number of epoch	10	15	50
Optimizer	Adam	Adam	Adam

4.2. Experimental Result

In this section, the experimental results were analyzed and evaluated to determine the best-performing approach for the maize leaf disease and pest identification dataset. This study compared training methods involving noise filtering techniques and varying numbers of epochs, using both sequential and EfficientNet models.

4.2.1. Comparison between noise filtering techniques in sequential model

In this study, three experiments were conducted to compare the performance of different noise filtering techniques, including Gaussian blur, median filter, and a combination of Gaussian and median filters. The maize leaf image dataset, consisting of 3,899 images (1,306 for common rust and fall armyworm, and 1,287 for the healthy class), was partitioned into 80% for training, 10% for validation, and 10% for testing.

The experimental results showed that the combination of Gaussian and median filtering (GF-MF) achieved the highest testing accuracy of 99.76%, outperforming the individual Gaussian filtering (99.28%) and median filtering (99.04%) techniques. The Gaussian and median results depicts in Appendix3. The GF-MF method effectively removed both Gaussian and salt-and-pepper noises, leading to better performance. Consequently, the GF-MF filtering technique was selected for further experiments.

The results and performance evaluation of the different approaches are presented below the table.

Table 4.2: Comparison between noises filtering technique in sequential model

Experiment number	filtering techniques used	accuracy in percentage		
		Training	Validation	Testing
Experiment 1	GF	99.48%	98.96%	99.28%
Experiment 2	MF	99.18%	98.44%	99.04%
Experiment 3	GF-MF	99.11%	99.74%	99.76%

4.2.2. Maize leaf disease and pest Classification Using Sequential Model

This section focuses on providing a detailed explanation of the experiment and model evaluation conducted for the identification and classification of maize leaf disease and pest using sequential model. All the training, validation and testing images underwent the necessary preprocessing steps. After preprocessing, the data was fed into a sequential model and trained accordingly. The experiment aimed to identify and classify diseases and pests affecting maize leaves.

The classification process involves three main steps: training, validation, and testing. The training set, which consists of labeled images of healthy, common rust and infected by fall armyworm pest maize leaves were used to optimize the parameters of the CNN through back propagation and gradient descent. The validation set is used for optimizing hyperparameters and selecting the best model. The performance of the trained model is evaluated using the testing set, which contains unseen images.

To train the model, the dataset is split into training, validation, and testing subsets. Data augmentation techniques, such as rotation, scaling, and flipping, are applied to increase the diversity of the training data and improve the model's generalization capability. The model is trained using the training subset, and the validation subset is used for hyperparameter tuning and early stopping criteria. During the training process, the model iteratively learns to recognize and classify the different classes of maize leaves—healthy, common rust affected, and Fall Armyworm infested. The training process involves forward and backward propagation, optimizing the model's parameters using gradient descent or its variants. The model achieved high accuracy rates of 99.11% for training, 99.74% for validation, and 99.76% for testing, indicating its effectiveness in disease identification and classification in sequential model.

The following table explains about the initial training process using Adam optimizer with learning rate of 0.001 with dropout 0.25 and batch normalization 32 and also by using three hidden layers (convolution, pooling, fully connected).

Table 4.3: Train the Sequential model on the training and validation dataset

Epoch	Duration	Accuracy	Loss	Validation Accuracy	Validation Loss	Test Accuracy
1/50	349s	0.7536	0.5586	0.9141	0.2370	99.76%
2/50	119s	0.9498	0.1486	0.9766	0.0805	
3/50	116s	0.9744	0.0729	0.9766	0.0683	
4/50	118s	0.9806	0.0758	0.9635	0.1000	
5/50	114s	0.9736	0.0984	0.9635	0.1413	
6/50	115s	0.9578	0.1379	0.9375	0.1850	
7/50	114s	0.9673	0.1052	0.9844	0.0434	
8/50	113s	0.9713	0.0941	0.9844	0.0521	
9/50	113s	0.9819	0.0600	0.9714	0.0707	
10/50	116s	0.9788	0.0765	0.9870	0.0358	
11/50	114s	0.9821	0.0761	0.9818	0.0358	
12/50	115s	0.9845	0.0559	0.9844	0.0426	
13/50	121s	0.9810	0.0552	0.9818	0.0747	
14/50	117s	0.9859	0.0542	0.9870	0.0564	
15/50	124s	0.9846	0.0641	0.9792	0.0389	
16/50	121s	0.9828	0.0573	0.9922	0.0391	
17/50	120s	0.9813	0.0660	0.9922	0.0236	
18/50	121s	0.9822	0.0541	0.9870	0.0352	
19/50	120s	0.9842	0.0579	0.9896	0.0345	
20/50	118s	0.9889	0.0388	0.9870	0.0327	
21/50	125s	0.9882	0.0396	0.9896	0.0367	
22/50	127s	0.9840	0.0470	0.9714	0.0968	
23/50	116s	0.9862	0.0518	0.9844	0.0348	
24/50	116s	0.9762	0.0818	0.9766	0.0636	
25/50	113s	0.9833	0.0533	0.9922	0.0326	

Epoch	Duration	Accuracy	Loss	Validation Accuracy	Validation Loss	Test Accuracy
26/50	114s	0.9919	0.0305	0.9844	0.0408	
27/50	116s	0.9918	0.0387	0.9948	0.0208	
28/50	113s	0.9841	0.0499	0.9922	0.0191	
29/50	116s	0.9903	0.0451	0.9896	0.0297	
30/50	114s	0.9955	0.0212	0.9922	0.0272	
31/50	113s	0.9924	0.0274	0.9896	0.0414	
32/50	115s	0.9914	0.0334	0.9766	0.0592	
33/50	113s	0.9907	0.0304	0.9922	0.0257	
34/50	114s	0.9942	0.0231	0.9896	0.0250	
35/50	113s	0.9868	0.0344	0.9870	0.0381	
36/50	113s	0.9915	0.0316	0.9948	0.0219	
37/50	119s	0.9922	0.0326	0.9922	0.0366	
38/50	114s	0.9921	0.0345	0.9948	0.0274	
39/50	112s	0.9856	0.0436	0.9896	0.0201	
40/50	114s	0.9960	0.0149	0.9948	0.0109	
41/50	113s	0.9887	0.0351	0.9974	0.0074	
42/50	120s	0.9925	0.0354	0.9870	0.0495	
43/50	124s	0.9888	0.0470	0.9922	0.0315	
44/50	121s	0.9937	0.0227	0.9974	0.0096	
45/50	119s	0.9949	0.0187	0.9818	0.0370	
46/50	113s	0.9900	0.0360	0.9844	0.0349	
47/50	114s	0.9962	0.0171	0.9922	0.0257	
48/50	115s	0.9938	0.0212	1.0000	0.0039	
49/50	113s	0.9937	0.0147	0.9896	0.0160	
50/50	113s	0.9911	0.0516	0.9974	0.0069	

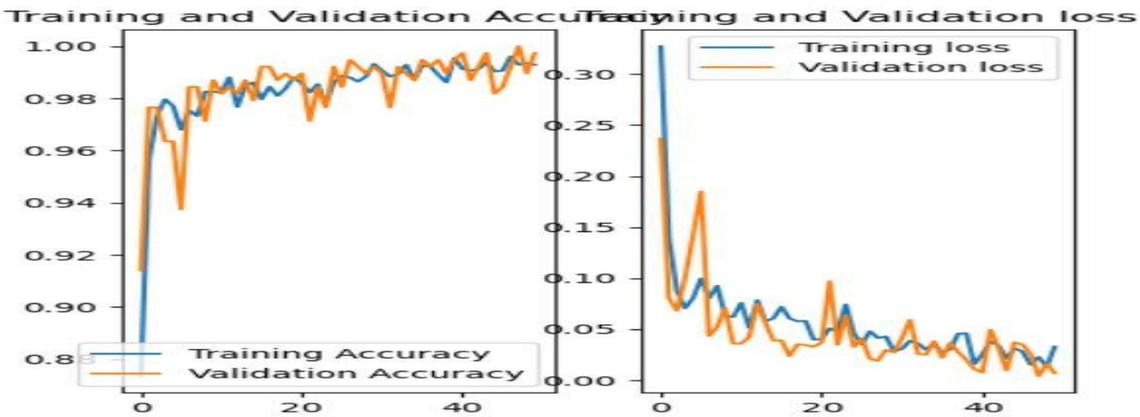


Figure 4.1: Sequential Model Training, Validation Accuracy and Loss

After training, the model's performance is evaluated using the testing subset, which contains unseen data.

The provided result for maize leaf disease and pest identification using a sequential model demonstrates exceptional performance, as indicated by the confusion matrix and classification report.

Confusion matrix: The confusion matrix reveals the model's predictions compared to the actual classes of the maize leaf samples. It is a 3x3 matrix, where the rows represent the true classes, and the columns represent the predicted classes. In this case, the matrix indicates that the model made perfect predictions for all classes:

- ✓ Common Rust: The model correctly predicted all 131 samples as Common Rust, without any false positives or false negatives.
- ✓ Fall Armyworm: The model accurately identified all 152 samples as Fall Armyworm, with no misclassifications.
- ✓ Healthy: The model predicted 132 samples correctly as healthy, while wrongly predicting 1 sample as Fall Armyworm.
- ✓ Classification report: The classification report provides evaluation metrics such as precision, recall, and F1-score for each class, along with the overall accuracy of the model. Here's how the metrics are described:
- ✓ Precision: The precision metric measures the proportion of correctly predicted samples out of the total predicted samples for a specific class. In this case, the precision for all classes is 1.00, indicating that the model made no false positive errors.
- ✓ Recall: The recall metric represents the proportion of correctly predicted samples out of the total actual samples for a particular class. Again, the recall for all classes is 1.00, indicating that the model made no false negative errors.
- ✓ F1-score: The F1-score is the harmonic mean of precision and recall, providing a single metric to assess the model's performance. In this scenario, the F1-scores for all classes are 1.00, indicating perfect predictions.
- ✓ Support: The support denotes the number of samples in each class.

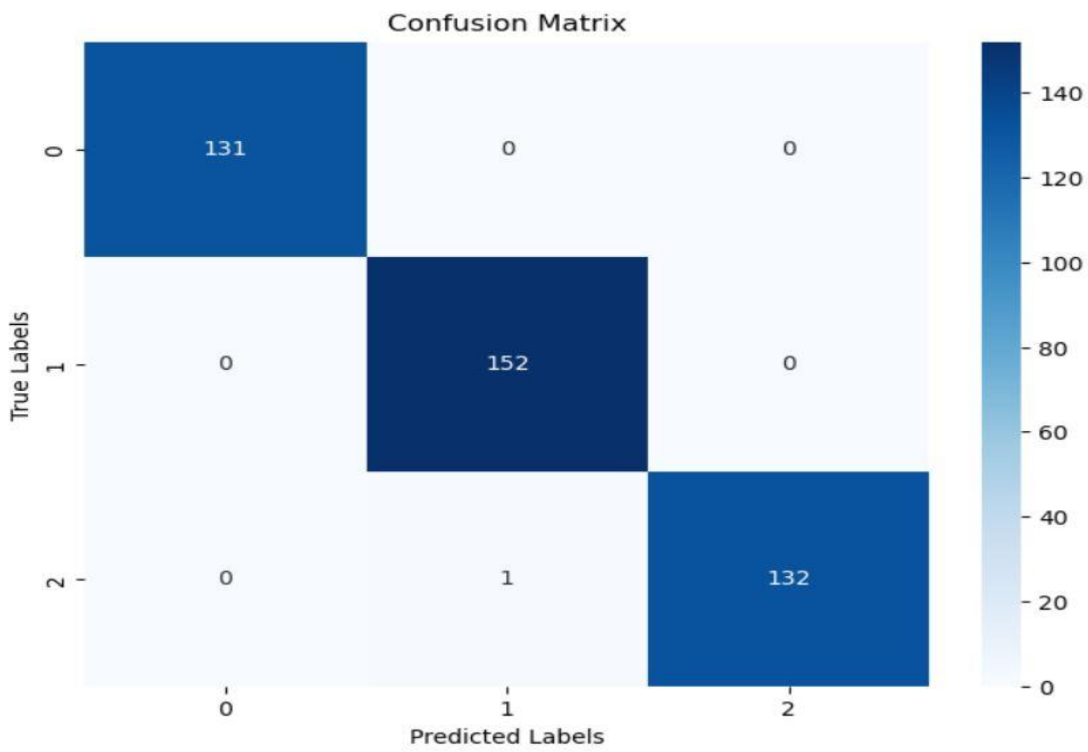


Figure 4.2: Confusion matrix of Sequential model

```
Confusion Matrix:
[[131   0   0]
 [  0 152   0]
 [  0   1 132]]
Classification Report:
              precision    recall  f1-score   support

 Common_rust       1.00      1.00      1.00       131
Fall_armyworm      0.99      1.00      1.00       152
     Healthy       1.00      0.99      1.00       133

    accuracy                           1.00       416
   macro avg       1.00      1.00      1.00       416
weighted avg       1.00      1.00      1.00       416
```

Figure 4.3 : Classification report for sequential model

4.2.3. Maize Leaf Disease and Pest Classification using the EfficientNetB0 Model

During the training and validation phase, it appears that an EfficientNet model was trained and evaluated for maize diseases, and pest identification and classification. The model achieved remarkable results, with high accuracy rates of 100% for training, validation, and testing.

During the training phase, the model initially started with an accuracy of 0.9204 and loss of 3.2448, indicating that it was already performing reasonably well. However, within a few epochs, the accuracy quickly improved and surpassed 0.99, suggesting that the model rapidly learned to make accurate predictions on the training data. The training loss consistently decreased over the initial epochs, indicating that the model was improving its predictions and reducing errors. This decreasing trend suggests that the model was effectively learning from the training data and becoming more accurate.

The validation accuracy and validation loss values fluctuated during the training process. While the model achieved high accuracy, the variations in validation accuracy and loss suggest that the model's performance on the validation dataset varied. Monitoring these values is crucial to ensure that the model generalizes well and performs consistently on unseen data.

The test accuracy consistently reached a perfect score of 1.00, indicating that the model performed flawlessly on the test dataset. This exceptional performance suggests that the model successfully learned to generalize well and make accurate predictions on unseen data, which is a desirable outcome.

In later epochs, the model achieved high accuracy and low loss values, indicating that it converged and stabilized its performance. This stability in performance further reinforces the model's effectiveness in maize leaf disease and pest identification and classification.

Overall, based on the training and validation accuracy and loss-provided in table 9, the EfficientNet model demonstrated exceptional performance, quickly learning to make accurate predictions, reducing training loss, and achieving perfect accuracy on both validation and test datasets. These results indicate the model's effectiveness in accurately identifying and classifying diseases.

Table 4.4: Train the EfficientNetB0 model on the training and validation dataset

Epoch	Duration	Accuracy	Loss	Validation Accuracy	Validation Loss	Test Accuracy
1	1263s	0.9204	0.2067	0.4349	3.2448	1.00
2	943s	0.9893	0.0356	0.7292	1.2153	
3	904s	0.9899	0.0321	0.5599	2.6790	
4	882s	0.9895	0.0423	0.9922	0.0359	
5	779s	0.9943	0.0193	1.0000	9.2663e-04	
6	813s	0.9993	0.0032	1.0000	1.7317e-04	
7	771s	0.9993	0.0030	0.9844	0.0406	
8	789s	0.9779	0.0655	0.8411	2.4111	
9	811s	0.9920	0.0307	0.9974	0.0028	
10	734s	0.9942	0.0127	0.9661	0.1201	
11	762s	0.9995	0.0048	1.0000	0.0018	
12	740s	0.9999	0.0019	1.0000	9.9836e-04	
13	800s	0.9970	0.0074	0.9974	0.0044	
14	762s	0.9987	0.0055	1.0000	1.0983e-04	
15	772s	0.9993	0.0037	0.9896	0.0338	
16	770s	0.9990	0.0040	0.9974	0.0222	
17	752s	0.9961	0.0111	0.9688	0.0882	
18	768s	0.9926	0.0258	0.9974	0.0067	
19	747s	0.9978	0.0115	0.9922	0.0250	
20	759s	0.9985	0.0058	0.9948	0.0152	
21	751s	0.9978	0.0074	0.9974	0.0175	
22	762s	0.9965	0.0100	1.0000	6.9565e-04	
23	750s	1.0000	6.2983e-04	1.0000	5.2261e-05	
24	763s	1.0000	3.5128e-04	0.9974	0.0065	
25	782s	0.9965	0.0150	0.9974	0.0022	
26	777s	0.9992	0.0036	1.0000	0.0011	
27	798s	0.9994	0.0016	0.9922	0.0465	
28	830s	0.9955	0.0143	0.9818	0.0632	
29	820s	0.9942	0.0242	0.9922	0.0478	
30	775s	0.9927	0.0234	0.9870	0.0373	

Epoch	Duration	Accuracy	Loss	Validation Accuracy	Validation Loss	Test Accuracy
31	773s	0.9965	0.0101	0.9896	0.0272	
32	750s	0.9985	0.0044	1.0000	2.5419e-04	
33	719s	1.0000	3.9426e-04	1.0000	3.9850e-05	
34	742s	1.0000	1.9361e-04	1.0000	3.8192e-05	
35	763s	1.0000	1.6938e-04	1.0000	3.0308e-05	
36	749s	1.0000	2.8268e-05	1.0000	1.3590e-05	
37	763s	1.0000	4.2323e-05	1.0000	1.1718e-05	
38	734s	1.0000	1.1192e-04	0.9974	0.0032	
39	752s	1.0000	1.4498e-04	1.0000	1.2285e-04	
40	754s	1.0000	7.4286e-05	1.0000	4.8650e-05	
41	765s	1.0000	1.4040e-05	1.0000	2.0457e-05	
42	793s	0.9971	0.0160	0.9401	0.2492	
43	905s	0.9904	0.0295	0.9557	0.2280	
44	812s	0.9926	0.0174	0.9948	0.0235	
45	806s	0.9969	0.0075	0.9974	0.0043	
46	834s	0.9967	0.0202	0.9792	0.0704	
47	852s	0.9979	0.0066	1.0000	4.3101e-04	
48	1878s	0.9986	0.0018	1.0000	1.1189e-04	
49	930s	0.9988	0.0062	0.9974	0.0043	
50	885s	1.0000	0.0012	1.0000	1.7241e-04	

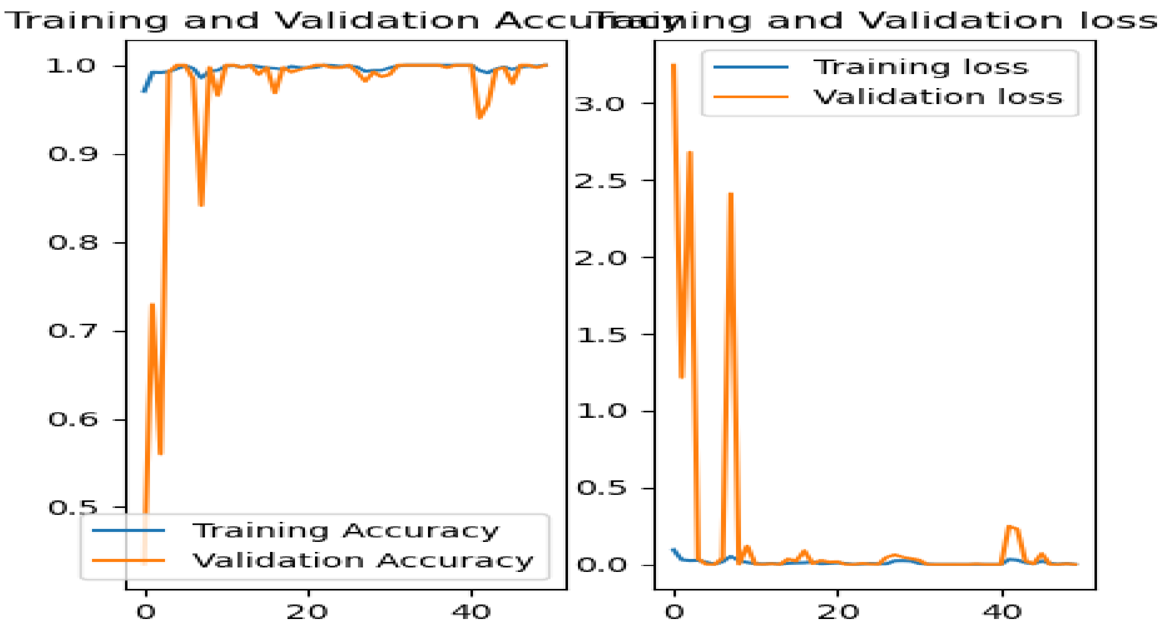


Figure 4.4: Training, Validation Accuracy and Loss of EfficientNetB0 Model

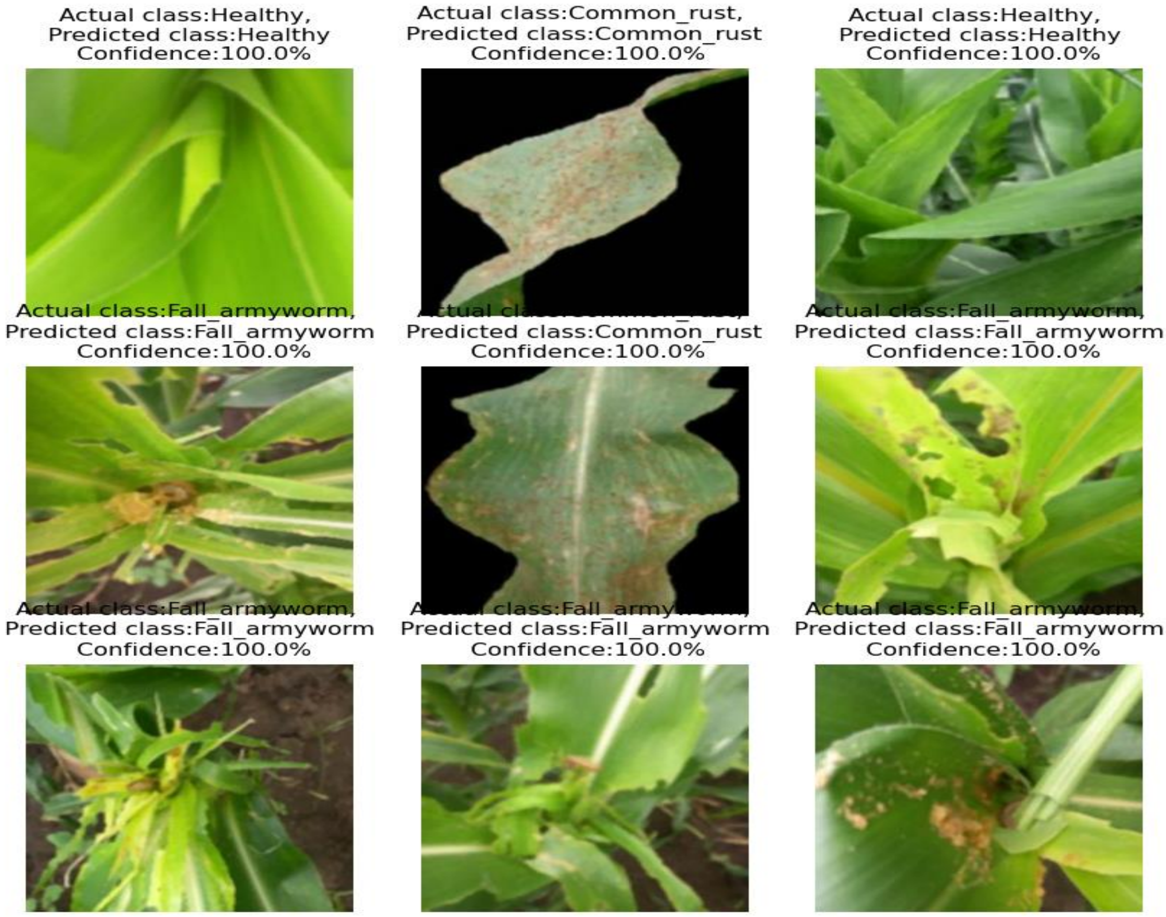


Figure 4.5: Actual Class, Prediction Class and confidence

The confusion matrix, precision, recall, and f1-score for this experiment are presented Classifying the testing data into their corresponding class was done using confusion matrix as follow:

The confusion matrix reveals the model's predictions compared to the actual classes of the maize leaf samples. It is a 3x3 matrix, where the rows represent the true classes, and the columns represent the predicted classes. In this case, the matrix indicates that the model made perfect predictions for all classes:

- ✓ Common Rust: The model correctly predicted all 131 samples as Common Rust, without any false positives or false negatives.
- ✓ Fall Armyworm: The model accurately identified all 148 samples as Fall Armyworm, with no misclassifications.
- ✓ Healthy: The model correctly classified all 137 samples as Healthy, without any errors.

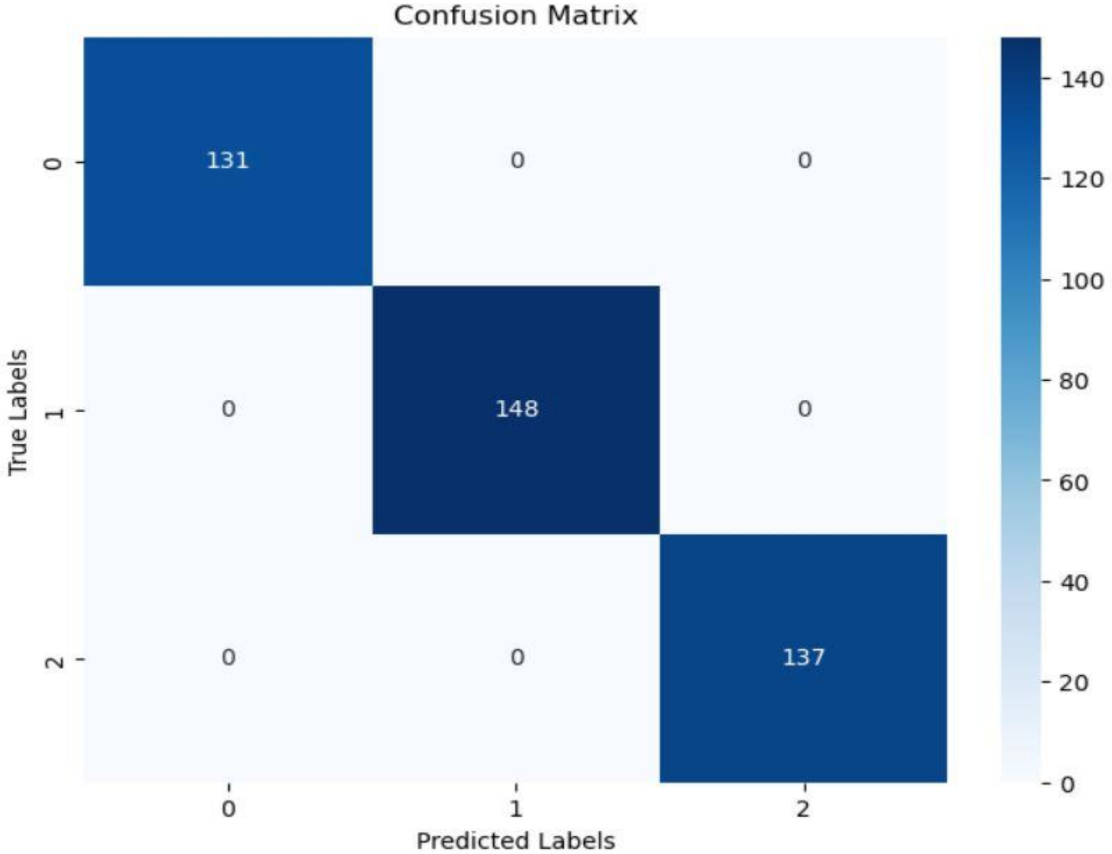


Figure 4.6: The result of Confusion Matrix of EfficientNetB0 model

The classification report shows precision, recall, and F1-score per class (in terms of metrics) as well as the macro average precision, recall, and F1-score (in terms of overall accuracy). Here is an explanation of the metrics:

- ✓ Precision: indicates the percentage of samples predicted correctly out of the total predicted samples in that class, for one class. The values for all classes are 1.00, indicating that there were no false positive errors made by the model.
- ✓ Recall: The recall metric is the percentage of correctly predicted samples out of all actual samples for a class. For the third time, there were no recalls less than 1.00, demonstrating that there were no false negative errors in any of the classes.
- ✓ F1-score: Our final metric – the F1-score– is defined as the harmonic mean of precision and recall, and again provides a single automatic measure of model performance. Again, all classes here have perfect F1 scores at 1.00.
- ✓ Support: The support denotes the number of samples in each class.

```
Confusion Matrix:
[[131   0   0]
 [  0 148   0]
 [  0   0 137]]
Classification Report:
              precision    recall  f1-score   support

 Common_rust       1.00      1.00      1.00       131
Fall_armyworm      1.00      1.00      1.00       148
     Healthy       1.00      1.00      1.00       137

    accuracy                           1.00       416
   macro avg       1.00      1.00      1.00       416
weighted avg       1.00      1.00      1.00       416
```

Figure 4.7: The Classification Report of EfficientNetB0 Model

Considering the results from the classification report, the model achieved outstanding performance in identifying the maize leaf diseases and pests. The precision, recall, and F1-scores for all classes are 1.00, demonstrating accurate predictions with no errors. The overall accuracy of the model is also 1.00, indicating that all 416 maize leaf samples were classified correctly.

4.3. Experimental Discussion

This study proposed the use of sequential and EfficientNetB0 models for maize leaf disease and pest identification. The models were trained with the preprocessed images, using a batch size of 32 and 64, a learning rate of 0.001 and 0.01 and 0.0001, and different numbers of epochs 10, 15, and 50. The results demonstrated that the EfficientNetB0 model achieved perfect training, validation, and testing accuracy utilizing 32 batch-sized 0.001 learning rates and 50 epochs and the sequential model achieved better validation and testing accuracy.

Overall, the study showcased the effectiveness of using deep learning architectures, such as sequential and EfficientNetB0, combined with noise filtering techniques for accurate identification of maize leaf diseases and pests. The findings provide valuable insights for developing robust and reliable agricultural monitoring systems.

The total time taken to train the proposed EfficienNetB0 model using Softmax classifier is around 10:42 hours approximately 750 seconds per epoch, and the Sequential model using the Softmax classifier is around 1:40 hours approximately 120 seconds per epoch. So, using the EfficienNetB0 model for the classification takes huge computational time.

In this experiment, we have used the proposed EfficienNetB0 model for feature extraction and softmax for classification and we have achieved 100 % for training, validations, and testing accuracy. When using the proposed Sequential model for feature extraction and softmax for classification we have achieved 99.11 training, 99.74 validations, and 99.76 % testing accuracy.

The proposed EfficienNetB0 model accuracy is greater than the proposed Sequential model with Softmax classifier by 0.89 training 0.26 validations and 0.24 % testing accuracy. The Sequential model takes less processing time compared with the EfficienNetB0 model.

4.3.1. Comparison between Proposed and Existing Methods

Table 4.5: Comparison of Proposed Method with Existing Methods

Reference	Plant type	Method	Better Accuracy
Himansu Das[39]	maize plant	NB,DT,KNN, SVM, and RF	79.23%
Erik Lucas da Rocha[40]	Maize	Bayesian hyperparameter optimization	97%
Md Ashraful[41]	maydis leaf blight maize disease	GoogleNet	99.14%
Arabinda Dash[45]	Maize	optimized support vector machine using deep feature of DenseNet201	94.6%
Malusi Sibiya 2021[42]	maize	VGG-16	95.63% of 89%
Wenqing (2023) [46]	Enhancing Corn Pest	ResNet 50 Model.	96.06 % 97.07 using CM
Enquhone Alehegn 2019[4]	Ethiopian maize diseases	Using both SVM and image processing.	95.5 %
Linigerew Mengstie Shita(2021) [43]	Ethiopian coffee leaf diseases	GF, MF and hybrid of GF&MF CNN-SVM and CNN-softmax classifier	96.02 %
Abebech Jenber Belay(2022) [44]	chickpea disease	(CNN) and (LSTM) Hybrid of GF&MF	92.55%
Proposed Sequential model	Maize disease and pest	GF, MF and hybrid of GF&MF Sequential model	99.76%
Proposed EfficientNetB0 model	Maize disease and pest	GF, MF and hybrid of GF&MF EfficientNetB0 model	100% over all accuracy

Table 4.6 : Comparison of proposed models with the existing models in evalluation metrics

Model Name	Class	Precision	Recall	F1-Score	Accuracy
DenseNet 201	Blight	0.99	0.99	0.99	99%
	Common Rust	1.00	1.00	1.00	
	Gray Leaf Spot	0.99	0.99	0.99	
	Healthy	1.00	1.00	1.00	
MobileNet	Blight	0.98	0.98	0.98	99%
	Common Rust	0.99	1.00	0.99	
	Gray Leaf Spot	0.98	0.98	0.98	
	Healthy	1.00	1.00	1.00	
Vgg16	Blight	0.99	0.97	0.98	99%
	Common Rust	1.00	1.00	1.00	
	Gray Leaf Spot	0.97	0.99	0.98	
	Healthy	1.00	1.00	1.00	
LeNet	Blight	0.98	0.87	0.90	
	Common Rust	0.90	0.98	0.93	94%
	Gray Leaf Spot	0.93	0.90	0.92	
	Healthy	0.98	1.00	0.99	
Custom CNN (One dropout layer)	Blight	0.92	0.92	0.92	95%
	Common Rust	0.96	0.96	0.96	
	Gray Leaf Spot	0.93	0.90	0.92	
	Healthy	0.97	1.00	0.99	
Custom CNN (Batch	Blight	0.93	0.95	0.94	96%
	Common Rust	0.96	0.98	0.97	
	Gray Leaf Spot	0.97	0.92	0.94	
	Healthy	0.98	1.00	0.99	
Custom CNN (No Regularization)	Blight	0.92	0.91	0.92	95%
	Common Rust	0.94	0.97	0.96	
	Gray Leaf Spot	0.93	0.92	0.92	
	Healthy	0.99	1.00	1.00	
GoogleNet	Maydis Leaf Blight	0.9930	0.9930	0.9930	99.14
	Healthy	0.9889	0.9889	0.9889	
Xception Model	Common Rust	0.78	0.95	0.86	0.79
	Blight	0.86	0.52	0.65	
	Healthy	0.96	0.85	0.90	

Model Name	Class	Precision	Recall	F1-Score	Accuracy
	Gray Leaf Spot	0.54	0.82	0.65	
CNN-LSTM	Ascochyta	0.90	0.90	0.90	0.93
	Fusarium Wilt	0.97	0.95	0.96	
	Healthy	0.90	0.91	0.91	
CNN Model with SVM Classifier	BES	0.99	1.00	1.00	0.96
	CBD	0.89	0.99	0.94	
	CLR	0.99	0.87	0.93	
	CWD	1.00	1.00	1.00	
Proposed Sequential model	Common Rust	1.00	1.00	1.00	1.00
	Fall armyworm	0.99	1.00	1.00	
	Healthy	1.00	0.99	1.00	
Proposed EfficientNetB0 model	Common Rust	1.00	1.00	1.00	1.00
	Fall armyworm	1.00	1.00	1.00	
	Healthy	1.00	1.00	1.00	

CHAPTER FIVE

5. Conclusion and Recommendation

5.1. Conclusion

This study aimed to address the limited research on automated diagnosis of maize leaf diseases and pests by creating a model capable of identifying them at an early stage. The development of such a model would greatly benefit farmers, agricultural research centers, and agricultural experts, while also improving the quality and quantity of maize crop productivity. We proposed a method that was utilized in a preprocessing phase to combine a Median and Gaussian filter to remove noise and trained a sequential and EfficientNetB0 model to identify and categorize maize leaf diseases and pests. The experimental results of the sequential model demonstrated high accuracy, with 99.11% training accuracy, 99.74% validation accuracy, and 99.76% testing accuracy, but EfficientNet model achieved the best accuracy rates of 100% for training, validation, and testing, indicating it's perfect and effectiveness in disease identification and classification result demonstrates an exceptional performance of the EfficientNetB0 model in identifying Common Rust, Fall Armyworm, and Healthy classes of maize leaf samples. This level of accuracy and precision is highly desirable for research results and discussion reports, as it signifies the model's reliability in distinguishing between different diseases and pests affecting maize plants. The outcome supports the credibility of the research findings and provides a strong basis for further discussions and conclusions. Thus, our constructed model EfficientNetB0 better than the sequential model shows promising performance in identifying and classifying three classes of maize leaf diseases, pests and health.

5.2. Recommendation

Maize is a crucial crop in Africa, particularly in countries like Ethiopia, where it is the largest producer. However, maize is highly susceptible to various diseases caused by fungi, bacteria, viruses, and pests, which significantly impact productivity. In this study, we focused on a subset of diseases and pests, including the healthy class, fall armyworm, and common rust, and enhanced the dataset quality using a hybrid of Gaussian and median filtering techniques. Before going further, it is important to note that these represent only a fraction of maize leaf diseases and pests, and there are alternative noise-filtering techniques available. As research, to make the disease and pest detection task much more accurate, we suggest the next "generation" of researchers could consider increasing the classes of diseases and pests for the maize leaf data set and other noise filtering techniques. Moreover, it could have achieved even higher classification accuracy by comparing it with various deep learning algorithms and other pre-trained models.

To further advance the accuracy of disease and pest detection tasks, we recommend that future researchers consider expanding the classes of diseases and pests in the maize leaf dataset. This would entail including a broader range of diseases, pests, and other common issues affecting maize crops. By encompassing a more comprehensive set of classes, the model will become more robust and capable of identifying a wider array of potential threats to maize plants.

Reference

[1] O. Erenstein, M. Jaleta, K. Sonder, K. Mottaleb, and B. M. Prasanna, “Global maize production, consumption and trade: trends and R&D implications,” *Food Secur.*, vol. 14, no. 5, pp. 1295–1319, 2022, doi: 10.1007/s12571-022-01288-7.

[2] E. Alemu Bekele, “Contributing factors of maize production using multiple linear regressions in mizan-aman district, bench-shako zone, Southwest of Ethiopia,” *Open J. Plant Sci.*, no. December, pp. 040–045, 2020, doi: 10.17352/ojps.000023.

[3] T. Abate, B. Shiferaw, A. Menkir, D. Wegary, and Y. Kebede, “Factors that transformed maize productivity in Ethiopia,” pp. 965–981, 2015, doi: 10.1007/s12571-015-0488-z.

[4] E. Alehegn, “Ethiopian maize diseases recognition and classification using support vector machine,” *Int. J. Comput. Vis. Robot.*, vol. 9, no. 1, pp. 90–109, 2019, doi: 10.1504/IJCVR.2019.098012.

[5] Z. Li *et al.*, “Maize leaf disease identification based on,” pp. 1–20, 2022, doi: 10.1371/journal.pone.0267650.

[6] M. Smale, “Maize Revolutions in Sub-Saharan Africa,” *Policy Res. Work. Pap. 5659 Abstr.*, vol. WPS5659, no. May, p. 32, 2011.

[7] G. Demissie, T. Deressa, M. Haile, G. Bogale, and M. Dida, “Prevalence , Distribution and Impact of Maize Lethal Necrosis Disease (MLND) in Ethiopia,” vol. 1, pp. 37–49, 2016.

[8] O. Ekpa, N. Palacios-rojas, G. Kruseman, V. Fogliano, and A. R. Linnemann, “Sub-Saharan African Maize-Based Foods - Processing Practices , Challenges and Opportunities Sub-Saharan African Maize-Based Foods - Processing Practices ,” *Food Rev. Int.*, vol. 35, no. 7, pp. 609–639, 2019, doi: 10.1080/87559129.2019.1588290.

[9] H. Terefe, G. G. Mengesha, G. Yitayih, and G. A. Bogale, “A large-scale survey reveals agro-ecological factors influence spatio-temporal distribution and epidemics of maize leaf blight: Implications for prioritizing sustainable management options,” *J. Sustain. Agric. Environ.*, vol. 2, no. 4, pp. 513–528, 2023, doi: 10.1002/sae2.12070.

[10] Z. Rashid *et al.*, “Genome-wide association studies in tropical maize germplasm reveal novel and known genomic regions for resistance to Northern corn leaf blight,” *Sci. Rep.*, vol. 10, no. 1, pp. 1–16, 2020, doi: 10.1038/s41598-020-78928-5.

[11] M. L. Elliott and P. F. Harmon, “Gray Leaf Spot,” *Edis*, vol. 2011, no. 2, pp. 1–4, 2011, doi: 10.32473/edis-lh047-2011.

[12] M. Gidi, “Major Maize Diseases in Ethiopia and their Management,” 2023, doi: 10.23880/oajar-16000305.

[13] T. Keno, D. Wegary, M. Worku, and B. T. Ertiro, “Major biotic maize production stresses in Ethiopia and their management through host resistance,” no. May, 2018, doi: 10.5897/AJAR2018.13163.

[14] T. Tilahun, D. Wagary, G. Demissie, M. Negash, S. Admassu, and H. Jifar, “Maize pathology research in Ethiopia in the 2000s: A review,” *Meet. Challenges Glob. Clim. Chang. Food Secur. through Innov. Maize Res.*, no. January, p. 193, 2012.

[15] B. Paudel Timilsena *et al.*, “Potential distribution of fall armyworm in Africa and beyond, considering climate change and irrigation patterns,” *Sci. Rep.*, vol. 12, no. 1, pp. 1–15, 2022, doi: 10.1038/s41598-021-04369-3.

[16] S. Reddy *et al.*, “The fall armyworm : recent advances in biology and management,” *J. Pest Sci. (2004).*, vol. 96, no. 4, pp. 1341–1343, 2023, doi: 10.1007/s10340-023-01688-4.

[17] F. Assefa and D. Ayalew, “Status and control measures of fall armyworm (Spodoptera frugiperda) infestations in maize fields in Ethiopia: A review,” *Cogent Food Agric.*, vol. 5, no. 1, 2019, doi: 10.1080/23311932.2019.1641902.

[18] K. Saarinen, *Image processing, analysis and machine vision*, vol. 35, no. 1. 1994. doi: 10.1016/0165-1684(94)90202-x.

[19] R. C. Gonzalez and R. E. Woods, “Digital Image Processing Third Edition Pearson,” p. 954, 2008.

[20] U. Hasdiana, *A Beginner’s Guide to Image Preprocessing Techniques*, vol. 11, no. 1. 2018. [Online]. Available: http://link.springer.com/10.1007/978-3-319-59379-1%0Ahttp://dx.doi.org/10.1016/B978-0-12-420070-8.00002-7%0Ahttp://dx.doi.org/10.1016/j.ab.2015.03.024%0Ahttps://doi.org/10.1080/07352689.2018.1441103%0Ahttp://www.chile.bmw-motorrad.cl/sync/showroom/lam/es/

[21] S. Ahmed Medjahed, “A Comparative Study of Feature Extraction Methods in Images Classification,” *Int. J. Image, Graph. Signal Process.*, vol. 7, no. 3, pp. 16–23, 2015, doi: 10.5815/ijigsp.2015.03.03.

[22] N. J. Nilsson, “INTRODUCTION TO MACHINE LEARNING AN EARLY DRAFT OF A PROPOSED TEXTBOOK Department of Computer Science,” 2005.

[23] M. Mohammed, M. B. Khan, and E. B. M. Bashie, *Machine learning: Algorithms and applications*, no. July. 2016. doi: 10.1201/9781315371658.

[24] A. Vishwakarma, "a Review: Machine Learning Algorithms," *Data Sci. Pract. Approach with Python R*, vol. 9, no. 1, pp. 162–175, 2024, doi: 10.58532/nbennurch299.

[25] I.H.Sarker1, "M_Learning_Algorithms,_Real-World_Applications_and_Research1.." p. 21, 2021. doi: org/10.1007/s42979-021-00592-x.

[26] Y. H. Liu *et al.*, "Modification on the tribological properties of ceramics lubricated by water using fullerenol as a lubricating additive," *Sci. China Technol. Sci.*, vol. 55, no. 9, pp. 2656–2661, 2012, doi: 10.1007/s11431-012-4938-y.

[27] I. G. and Y. B. and A. Courville, "Deep learning Book pdf," *Nature*, vol. 29, no. 7553, pp. 1–73, 2016.

[28] R. Patel and S. Patel, "A comprehensive study of applying convolutional neural network for computer vision," *Int. J. Adv. Sci. Technol.*, vol. 29, no. 6 Special Issue, pp. 2161–2174, 2020.

[29] I. Goodfellow, Y. Bengio, and A. Courville, "deep learning English version," p. 800, 2017.

[30] A. Patil and M. Rane, "Convolutional Neural Networks: An Overview and Its Applications in Pattern Recognition," *Smart Innov. Syst. Technol.*, vol. 195, pp. 21–30, 2021, doi: 10.1007/978-981-15-7078-0_3.

[31] A. Ghosh, A. Sufian, F. Sultana, A. Chakrabarti, and D. De, *Fundamental Concepts of Convolutional Neural Network*, no. June. 2020. doi: 10.1007/978-3-030-32644-9.

[32] M. McGregor, "What Is a Convolutional Neural Network? A Beginner's Tutorial for Machine Learning and Deep Learning," *FreeCodeCamp*, pp. 1–11, 2021, [Online]. Available: https://www.freecodecamp.org/news/convolutional-neural-network-tutorial-for-beginners/

[33] K. T. O. Shea, "An Introduction to Convolutional Neural Networks An Introduction to Convolutional Neural Networks," no. November, 2015.

[34] O. Layer, "Different Types of CNN Architectures Explained : Examples," 2023.

[35] S. Indolia, A. K. Goswami, S. P. Mishra, and P. Asopa, "Conceptual Understanding of Convolutional Neural Network- A Deep Learning Approach," *Procedia Comput. Sci.*, vol. 132, pp. 679–688, 2018, doi: 10.1016/j.procs.2018.05.069.

[36] I. O. Muraina and S. O. Abam, "DATA ANALYTICS EVALUATION METRICS ESSENTIALS : MEASURING MODEL," no. August, 2023.

[37] D. Krstinić, M. Braović, L. Šerić, and D. Božić-Štulić, “Multi-label Classifier Performance Evaluation with Confusion Matrix,” pp. 01–14, 2020, doi: 10.5121/csit.2020.100801.

[38] A. Arias-Duart, E. Mariotti, D. Garcia-Gasulla, and J. M. Alonso-Moral, “A Confusion Matrix for Evaluating Feature Attribution Methods,” *IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Work.*, vol. 2023-June, no. Cv, pp. 3709–3714, 2023, doi: 10.1109/CVPRW59228.2023.00380.

[39] K. P. Panigrahi, H. Das, and A. K. Sahoo, “Maize Leaf Disease Detection and Classification Using Machine Learning Maize Leaf Disease Detection and Classification Using Machine,” *Publ. Maize*, no. January, 2020, doi: 10.1007/978-981-15-2414-1.

[40] E. Lucas, L. F. Rodrigues, and J. F. Mari, “Maize leaf disease classification using convolutional neural networks and hyperparameter optimization,” no. October, 2020, doi: 10.5753/wvc.2020.13489.

[41] A. Haque, S. Marwaha, C. K. Deb, and S. Nigam, “Deep learning - based approach for identification of diseases of maize crop,” *Sci. Rep.*, pp. 1–14, 2022, doi: 10.1038/s41598-022-10140-z.

[42] M. Sibiya and M. Sumbwanyambe, “Automatic Fuzzy Logic-Based Maize Common Rust Disease Severity Predictions with Thresholding and Deep Learning,” *MDPI*, p. 17, 2021, doi: https://doi.org/10.3390/pathogens10020131.

[43] L. M. Shita, “ETHIOPIAN COFFEE LEAF DISEASES IDENTIFICATION USING DEEP LEARNING,” *JETIR*, vol. 8, no. 10, pp. 498–508, 2021.

[44] A. Jenber, A. Olalekan, M. Ashagrie, and M. Bitew, “Informatics in Medicine Unlocked Development of a chickpea disease detection and classification model using deep learning,” *Informatics Med. Unlocked*, vol. 31, no. April, p. 100970, 2022, doi: 10.1016/j.imu.2022.100970.

[45] A. Dash, P. Kumar, and S. Kumari, “Maize disease identification based on optimized support vector machine using deep feature of DenseNet201,” *J. Agric. Food Res.*, vol. 14, no. August, p. 100824, 2023, doi: 10.1016/j.jafr.2023.100824.

[46] W. Xu, W. Li, L. Wang, and M. F. Pompelli, “Enhancing Corn Pest and Disease Recognition through Deep Learning : A Comprehensive Analysis,” p. 16, 2023, doi: 10.3390/agronomy13092242.

Appendices

Appendix1: Sequential model code

```
Import tensorflow as tf
from tensorflow.keras import models, layers
from matplotlib import pyplot as plt
import pathlib
import glob
import warnings
import json
import pandas as pd
import numpy as np
from tqdm import tqdm
import cv2
from collections import defaultdict
from urllib import request
import os
import pandas as pd
import numpy as np
from urllib import request
import cv2
from tqdm import tqdm
from collections import Counter
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
import tensorflow as tf
from tensorflow.keras.layers import Dropout,Dense
from keras.models import Model
from keras.callbacks import EarlyStopping,ModelCheckpoint
import os
from tensorflow.keras.applications import VGG16
from tensorflow.keras.applications import ResNet50
```

```
from tensorflow.keras.preprocessing.image import ImageDataGenerator
import json
from sklearn import preprocessing
from sklearn.preprocessing import OrdinalEncoder
import tensorflow
import tensorflow as tf
from collections import deque
file_path='D:/CNN/Paulos/Data'
dataset_path='D:/CNN/Paulos/Data'
print(dataset_path)
print("Types of class lables found: ",len(dataset_path))
# Set hyperparameters
IMAGE_SIZE = 224
BATCH_SIZE = 32
CHANNELS = 3
EPOCH = 50
LEARNING_RATE = 0.001
dataset = tf.keras.preprocessing.image_dataset_from_directory(
   dataset_path,
    shuffle=True,
    image_size=(IMAGE_SIZE,IMAGE_SIZE),
    batch_size=BATCH_SIZE
)
class_names = dataset.class_names
class_names
len(dataset)
plt.figure(figsize=(10,10))
for img_batch, label_batch in dataset.take(1):
    print(img_batch.shape)
    print(label_batch.numpy())
    for i in range(16):
        ax = plt.subplot(4,4,i+1)
```

```
        plt.imshow(img_batch[i].numpy().astype('uint8'))
        plt.axis('off')
        plt.title(class_names[label_batch[i]])
def get_tf_dataset_partitions(ds, train_split=0.8, val_split=0.1, test_split=0.1, shuffle= True,
shuffle_size=2000):
    ds_size = len(ds)
    if shuffle:
        ds = ds.shuffle(shuffle_size, seed=20)
    train_size = int(train_split * ds_size)
    val_size = int(val_split * ds_size)
    train_ds = ds.take(train_size)
    val_ds = ds.skip(train_size).take(val_size)
    test_ds = ds.skip(train_size).skip(val_size)
    return train_ds, val_ds, test_ds
train_ds, val_ds, test_ds = get_tf_dataset_partitions(dataset)
train_ds = train_ds.cache().shuffle(600).prefetch(buffer_size=tf.data.AUTOTUNE)
val_ds = val_ds.cache().shuffle(600).prefetch(buffer_size=tf.data.AUTOTUNE)
test_ds = test_ds.cache().shuffle(600).prefetch(buffer_size=tf.data.AUTOTUNE)
<h4> Preprocessing </h4>
IMAGE_SIZE = (224, 224)  # Specify the desired image size
resize_and_scale = tf.keras.Sequential([
    tf.keras.layers.Rescaling(1.0 / 255),
    tf.keras.layers.Resizing(*IMAGE_SIZE)
])
import tensorflow as tf
from tensorflow.keras import layers
data_augmentation = tf.keras.Sequential([
    layers.RandomFlip('horizontal_and_vertical'),
    layers.RandomRotation(0.2)
])
```

```
import tensorflow as tf
from tensorflow.keras import layers, models
BATCH_SIZE = 32
IMAGE_SIZE = 224
CHANNELS = 3
n_classes = 3
LEARNING_RATE = 0.001
input_shape = (BATCH_SIZE, IMAGE_SIZE, IMAGE_SIZE, CHANNELS)
model = models.Sequential([
    resize_and_scale,
    data_augmentation,
    layers.Conv2D(32, (3, 3), activation='relu', input_shape=(IMAGE_SIZE, IMAGE_SIZE,
CHANNELS)),
    layers.MaxPooling2D((2, 2)),
    layers.Conv2D(64, (3, 3), activation='relu'),
    layers.MaxPooling2D((2, 2)),
    layers.Conv2D(64, (3, 3), activation='relu'),
    layers.MaxPooling2D((2, 2)),
    layers.Conv2D(64, (3, 3), activation='relu'),
    layers.MaxPooling2D((2, 2)),
    layers.Flatten(),
    layers.Dense(64, activation='relu'),
    layers.Dropout(0.25),  # Add dropout layer with rate 0.25
    layers.Dense(n_classes, activation='softmax')
])
model.build(input_shape=input_shape)
model.summary()
model.compile(optimizer='adam',
              loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits=False),
              metrics=['accuracy'])
history = model.fit(train_ds,
        epochs=EPOCH,
```

```
        batch_size=BATCH_SIZE,
        verbose=1,
        validation_data=val_ds)
model.evaluate(test_ds)
acc = history.history['accuracy']
loss = history.history['loss']
val_acc = history.history['val_accuracy']
val_loss = history.history['val_loss']
plt.figure(figsize=(5,5))
plt.subplot(1,2,1)
plt.plot(range(EPOCH), acc, label='Training Accuracy')
plt.plot(range(EPOCH), val_acc, label='Validation Accuracy')
plt.legend(loc='lower right')
plt.title('Training and Validation Accuracy')
plt.subplot(1,2,2)
plt.plot(range(EPOCH), loss, label='Training loss')
plt.plot(range(EPOCH), val_loss, label='Validation loss')
plt.legend(loc='upper right')
plt.title('Training and Validation loss')
import numpy as np
for images_batch, labels_batch in test_ds.take(1):
    first_image = images_batch[0].numpy().astype('uint8')
    first_label = labels_batch[0].numpy()
    print('First image to predict')
    plt.imshow(first_image)
    print("First image's label:", class_names[first_label])
    batch_prediction = model.predict(images_batch)
    print("Predicted Label:",class_names[np.argmax(batch_prediction[0])])
def predict(model, img):
    img_array = tf.keras.preprocessing.image.img_to_array(images[i].numpy())
    img_array = tf.expand_dims(img_array,0)
    predictions = model.predict(img_array)
```

```
    pred_class = class_names[np.argmax(predictions[0])]
    confidence = round(100 * np.max(predictions[0]), 2)
    return pred_class, confidence
plt.figure(figsize=(10,10))
for images, labels in test_ds.take(1):
    for i in range(9):
        ax = plt.subplot(3,3,i+1)
        plt.imshow(images[i].numpy().astype('uint8'))
        predicted_class,confidence = predict(model, images[i].numpy())
        actual_class = class_names[labels[i].numpy()]
        plt.title(f"Actual class:{actual_class}, \n Predicted class:{predicted_class} \n
Confidence:{confidence}%")
        plt.axis('off')
from sklearn.metrics import confusion_matrix, classification_report
# Obtain predictions for test data
y_true = []
y_pred = []
for images, labels in test_ds:
    for i in range(len(labels)):
        y_true.append(labels[i].numpy())
        predicted_class, _ = predict(model, images[i].numpy())
        y_pred.append(class_names.index(predicted_class))
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.metrics import confusion_matrix, classification_report, accuracy_score
# Calculate confusion matrix
cm = confusion_matrixcm = confusion_matrix(y_true, y_pred)
# Plot confusion matrix as a heatmap with color
plt.figure(figsize=(8, 6))
sns.heatmap(cm, annot=True, fmt="d", cmap="Blues")
plt.title('Confusion Matrix')
plt.xlabel('Predicted Labels')
```

```
plt.ylabel('True Labels')
plt.show()
# Calculate accuracy score
accuracy = accuracy_score(y_true, y_pred)
# Print accuracy
print(f"Accuracy: {accuracy:.2%}")
# Calculate confusion matrix
cm = confusion_matrix(y_true, y_pred)
print("Confusion Matrix:")
print(cm)
# Calculate classification report
report = classification_report(y_true, y_pred, target_names=class_names)
print("Classification Report:")
print(report)
model.save("maize_leafm&g d0.25lr0.001.h5")
```

Appendix2: EfficientNetB0 model code

```
import tensorflow as tf
from tensorflow.keras import models, layers
from matplotlib import pyplot as plt
import pathlib
import glob
import warnings
import json
import pandas as pd
import numpy as np
from tqdm import tqdm
import cv2
from collections import defaultdict
from urllib import request
import os
import pandas as pd
import numpy as np
```

```
from urllib import request
import cv2
from tqdm import tqdm
from collections import Counter
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
import tensorflow as tf
from tensorflow.keras.layers import Dropout,Dense
from tensorflow.keras.models import Model
from tensorflow.keras.callbacks import EarlyStopping, ModelCheckpoint
import os
from tensorflow.keras.applications import VGG16
from tensorflow.keras.applications import ResNet50
from tensorflow.keras.preprocessing.image import ImageDataGenerator
import json
from sklearn import preprocessing
from sklearn.preprocessing import OrdinalEncoder
import tensorflow
import tensorflow as tf
from collections import deque
import efficientnet.keras as efn
file_path='D:/CNN/Paulos/Data'
dataset_path='D:/CNN/Paulos/Data'
print(dataset_path)
print("Types of class lables found: ",len(dataset_path))
# Set hyperparameters
IMAGE_SIZE = 224
BATCH_SIZE = 32
CHANNELS = 3
EPOCH = 50
LEARNING_RATE = 0.001
dataset = tf.keras.preprocessing.image_dataset_from_directory(
```

```
   dataset_path,
    shuffle=True,
    image_size=(IMAGE_SIZE,IMAGE_SIZE),
    batch_size=BATCH_SIZE
)
class_names = dataset.class_names
class_names
len(dataset)
plt.figure(figsize=(10,10))
for img_batch, label_batch in dataset.take(1):
    print(img_batch.shape)
    print(label_batch.numpy())
    for i in range(16):
        ax = plt.subplot(4,4,i+1)
        plt.imshow(img_batch[i].numpy().astype('uint8'))
        plt.axis('off')
        plt.title(class_names[label_batch[i]])
def get_tf_dataset_partitions(ds, train_split=0.8, val_split=0.1, test_split=0.1, shuffle= True,
shuffle_size=2000):
    ds_size = len(ds)
    if shuffle:
        ds = ds.shuffle(shuffle_size, seed=20)
    train_size = int(train_split * ds_size)
    val_size = int(val_split * ds_size)
    train_ds = ds.take(train_size)
    val_ds = ds.skip(train_size).take(val_size)
    test_ds = ds.skip(train_size).skip(val_size)
    return train_ds, val_ds, test_ds
train_ds, val_ds, test_ds = get_tf_dataset_partitions(dataset)
train_ds = train_ds.cache().shuffle(600).prefetch(buffer_size=tf.data.AUTOTUNE)
val_ds = val_ds.cache().shuffle(600).prefetch(buffer_size=tf.data.AUTOTUNE)
test_ds = test_ds.cache().shuffle(600).prefetch(buffer_size=tf.data.AUTOTUNE)
```

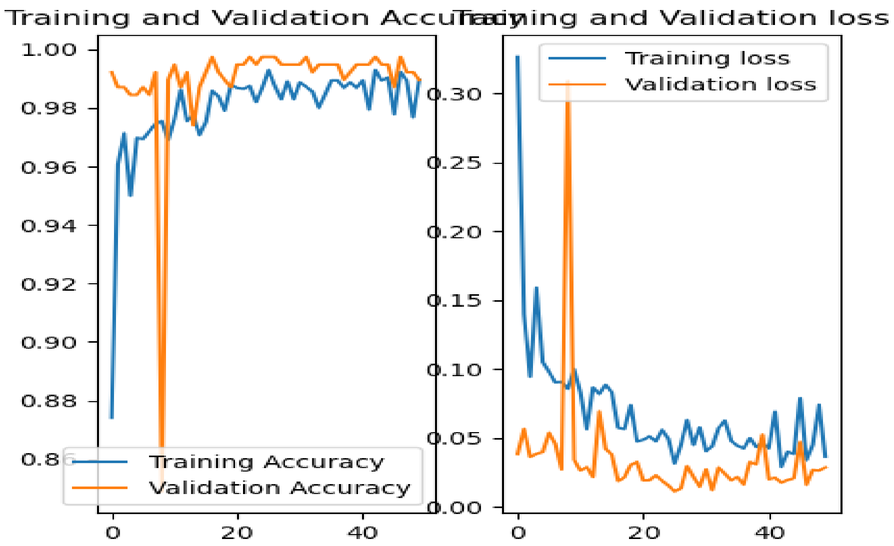
```
<h4> Preprocessing </h4>
IMAGE_SIZE = (224, 224)  # Specify the desired image size
resize_and_scale = tf.keras.Sequential([
    tf.keras.layers.Rescaling(1.0 / 255),
    tf.keras.layers.Resizing(*IMAGE_SIZE)
])
import tensorflow as tf
from tensorflow.keras import layers
data_augmentation = tf.keras.Sequential([
    layers.RandomFlip('horizontal_and_vertical'),
    layers.RandomRotation(0.2)
])
base_model = efn.EfficientNetB0(input_shape = (224, 224, 3), include_top = False, weights
= 'imagenet')
for layer in base_model.layers:
    layer.trainable = False
import tensorflow as tf
import efficientnet.tfkeras as efn
BATCH_SIZE = 32
IMAGE_SIZE = 224
CHANNELS = 3
n_classes = 3
LEARNING_RATE = 0.001
input_shape = (BATCH_SIZE, IMAGE_SIZE, IMAGE_SIZE, CHANNELS)
model = tf.keras.Sequential([
    resize_and_scale,
    data_augmentation,
    efn.EfficientNetB0(include_top=False, input_shape=(IMAGE_SIZE, IMAGE_SIZE,
CHANNELS)),
    layers.GlobalAveragePooling2D(),
    layers.Dense(64, activation='relu'),
    layers.Dropout(0.25),
```

```
    layers.Dense(n_classes, activation='softmax')
])
model.build(input_shape=input_shape)
model.summary()
model.compile(optimizer='adam',
              loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits=False),
              metrics=['accuracy'])
history = model.fit(train_ds,
          epochs=EPOCH,
          batch_size=BATCH_SIZE,
          verbose=1,
          validation_data=val_ds)
model.evaluate(test_ds)
acc = history.history['accuracy']
loss = history.history['loss']
val_acc = history.history['val_accuracy']
val_loss = history.history['val_loss']
plt.figure(figsize=(5,5))
plt.subplot(1,2,1)
plt.plot(range(EPOCH), acc, label='Training Accuracy')
plt.plot(range(EPOCH), val_acc, label='Validation Accuracy')
plt.legend(loc='lower right')
plt.title('Training and Validation Accuracy')
plt.subplot(1,2,2)
plt.plot(range(EPOCH), loss, label='Training loss')
plt.plot(range(EPOCH), val_loss, label='Validation loss')
plt.legend(loc='upper right')
plt.title('Training and Validation loss')
import numpy as np
for images_batch, labels_batch in test_ds.take(1):
    first_image = images_batch[0].numpy().astype('uint8')
    first_label = labels_batch[0].numpy()
```

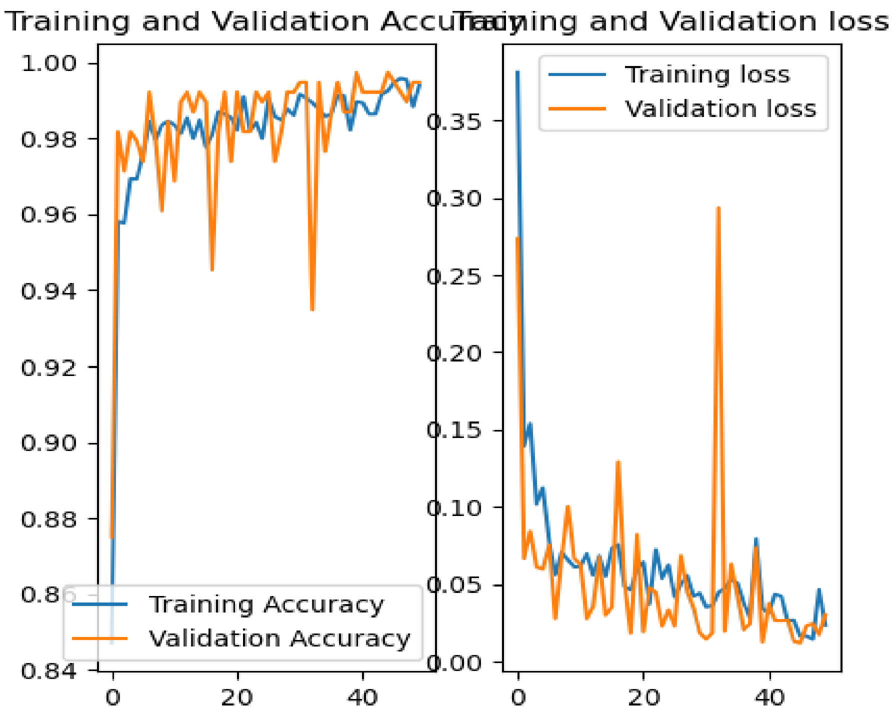
```
    print('First image to predict')
    plt.imshow(first_image)
    print("First image's label:", class_names[first_label])
    batch_prediction = model.predict(images_batch)
    print("Predicted Label:",class_names[np.argmax(batch_prediction[0])])
def predict(model, img):
    img_array = tf.keras.preprocessing.image.img_to_array(images[i].numpy())
    img_array = tf.expand_dims(img_array,0)
    predictions = model.predict(img_array)
    pred_class = class_names[np.argmax(predictions[0])]
    confidence = round(100 * np.max(predictions[0]), 2)
    return pred_class, confidence
plt.figure(figsize=(10,10))
for images, labels in test_ds.take(1):
    for i in range(9):
        ax = plt.subplot(3,3,i+1)
        plt.imshow(images[i].numpy().astype('uint8'))
        predicted_class,confidence = predict(model, images[i].numpy())
        actual_class = class_names[labels[i].numpy()]
        plt.title(f"Actual class:{actual_class}, \n Predicted class:{predicted_class} \n
Confidence:{confidence}%")
        plt.axis('off')
from sklearn.metrics import confusion_matrix, classification_report
# Obtain predictions for test data
y_true = []
y_pred = []
for images, labels in test_ds:
    for i in range(len(labels)):
        y_true.append(labels[i].numpy())
        predicted_class, _ = predict(model, images[i].numpy())
        y_pred.append(class_names.index(predicted_class))
import seaborn as sns
```

```
import matplotlib.pyplot as plt
from sklearn.metrics import confusion_matrix, classification_report, accuracy_score
# Calculate confusion matrix
cm = confusion_matrixcm = confusion_matrix(y_true, y_pred)
# Plot confusion matrix as a heatmap with color
plt.figure(figsize=(8, 6))
sns.heatmap(cm, annot=True, fmt="d", cmap="Blues")
plt.title('Confusion Matrix')
plt.xlabel('Predicted Labels')
plt.ylabel('True Labels')
plt.show()
# Calculate accuracy score
accuracy = accuracy_score(y_true, y_pred)
# Print accuracy
print(f"Accuracy: {accuracy:.2%}")
# Calculate confusion matrix
cm = confusion_matrix(y_true, y_pred)
print("Confusion Matrix:")
print(cm)
# Calculate classification report
report = classification_report(y_true, y_pred, target_names=class_names)
print("Classification Report:")
print(report)
#%%
model.save("maize_leafm d0.25lr0.001.h5")
```

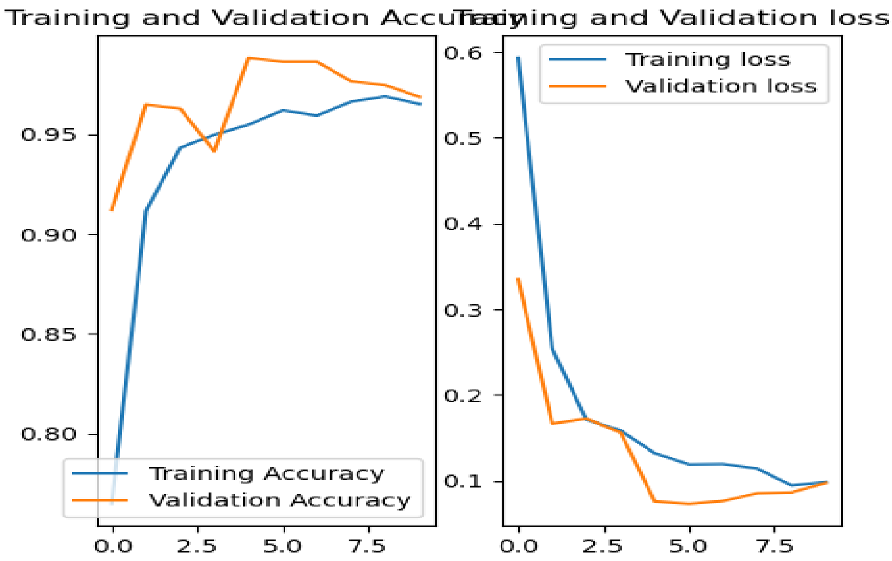
Appendix3: Experimental results in Gaussian, Median in different epochs and learning rate



Training and validation accuracy and loss in gausian filtering technique



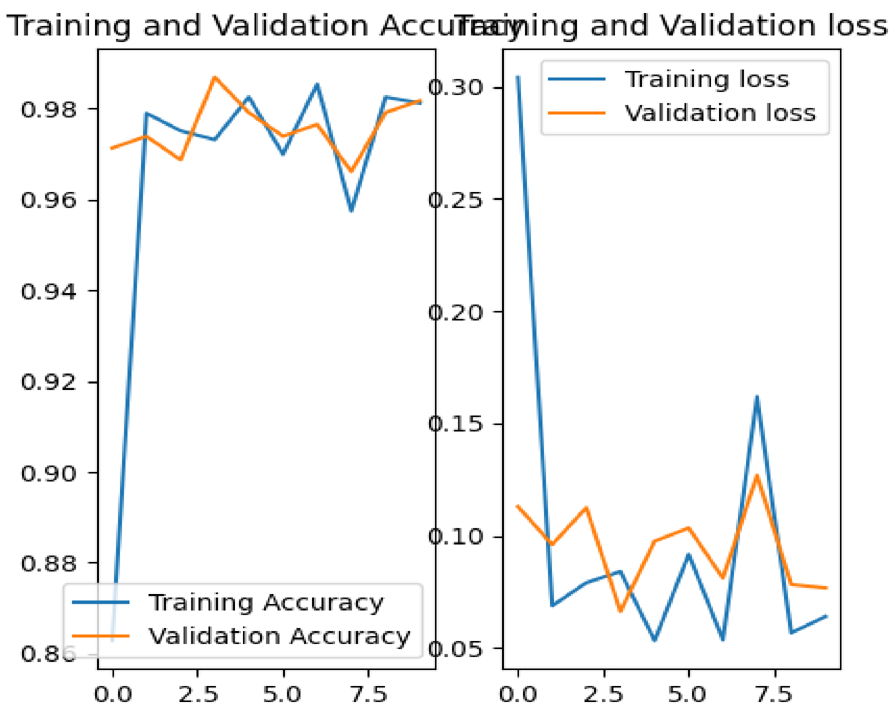
Training and validation accuracy and loss in median filtering technique in 50 epochs



Training and validation accuracy and loss in hybrid of G&M filtering technique in 10 epochs 0.01 learning rate

Table: Training validation accuracy and loss hybrid of Gaussian and median filtering technique in 10 epoch and 0.001 learning rate

Epoch	**Duration**	**Accuracy**	**Loss**	**Validation Accuracy**	**Validation Loss**	**Test Accuracy**
1	882s	0.8629	0.3042	0.9714	0.1129	98.31
2	346s	0.9790	0.0687	0.9740	0.0960	
3	333s	0.9752	0.0790	0.9688	0.1124	
4	332s	0.9732	0.0840	0.9870	0.0662	
5	332s	0.9826	0.0533	0.9792	0.0975	
6	331s	0.9700	0.0916	0.9740	0.1034	
7	330s	0.9855	0.0535	0.9766	0.0810	
8	334s	0.9574	0.1618	0.9661	0.1269	
9	337s	0.9826	0.0566	0.9792	0.0782	
10	337s	0.9813	0.0640	0.9818	0.0767	



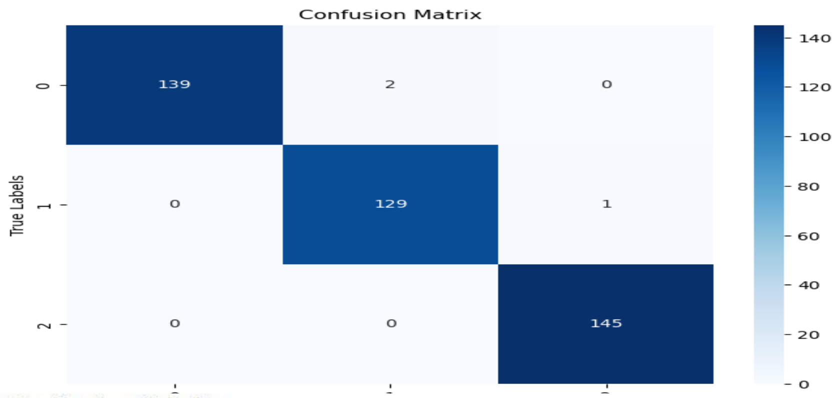
Training validation accuracy and loss hybrid of Gaussian and median filtering technique in 10 epoch and 0.001 learning rate

```
Confusion Matrix:
[[155   1   3]
 [  0 117   3]
 [  0   0 137]]
Classification Report:
              precision    recall  f1-score   support

 Common_rust       1.00      0.97      0.99       159
Fall_armyworm       0.99      0.97      0.98       120
     Healthy       0.96      1.00      0.98       137

    accuracy                           0.98       416
   macro avg       0.98      0.98      0.98       416
weighted avg       0.98      0.98      0.98       416
```

Classification report in Hybrid of gausian and median filter 10 epochs and 0.001 learning rate

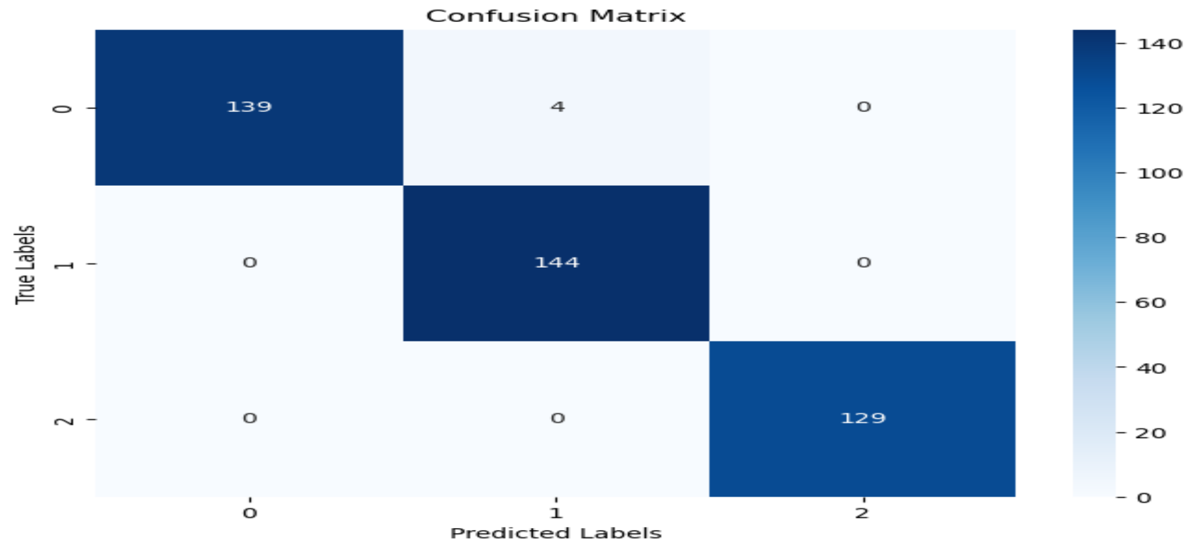



```
Confusion Matrix:
[[139   2   0]
 [  0 129   1]
 [  0   0 145]]
Classification Report:
              precision    recall  f1-score   support

 Common_rust       1.00      0.99      0.99       141
Fall_Armyworm      0.98      0.99      0.99       130
     healthy       0.99      1.00      1.00       145

    accuracy                           0.99       416
   macro avg       0.99      0.99      0.99       416
weighted avg       0.99      0.99      0.99       416
```

Confusion matrix and classification report in gausian filtering technique in 50 epochs



Confusion matrix in Median filtering technique in 50 epochs