An Efficient Cucumber Leaf Disease Diagnosis using Deep Learning Algorithms

A Dissertation

Submitted to the Council of Erbil Technical Engineering College at Erbil Polytechnic University in Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy in Information Systems Engineering

By

Saman Muhammad Omer

B.Sc. Software Engineering (2009)
M.Sc. Advanced Software Engineering (2012)

Supervised by

Asst. Prof. Dr. Kayhan Zrar Ghafoor

Prof. Dr. Shavan Kamal Askar

Erbil, Kurdistan

November 2023

DECLARATION

I declare that the Higher PhD. Dissertation entitled: “An Efficient Cucumber Leaf Disease Diagnosis using Deep Learning Algorithms” is my own original work, and hereby I certify that unless stated, all work contained within this dissertation is my own independent research and has not been submitted for the award of any other degree at any institution, except where due acknowledgment is made in the text.

Signature:

Student Name: Saman Muhammad Omer

Date: November / 2023

CERTIFICATE OF PROOFREADING

This is to certify that this dissertation entitled: “An Efficient Cucumber Leaf Disease Diagnosis using Deep Learning Algorithms” written by the postgraduate student (Saman Muhammad Omer) has been proofread and checked for grammatical, punctuation, and spelling mistakes. Therefore, after making all the required corrections by the student for further improvement, I confirm that this last copy of the dissertation is ready for submission.

Signature:

Name: Asst. Prof. Dr. Salih Ibrahim Ahmed

Phone No. : 07701502771

Email Address: salih.ahmed@uor.edu.krd

Date: 12/08/2023

SUPERVISOR CERTIFICATE

This dissertation has been written under my supervision and has been submitted for the award of the degree of Doctor of Philosophy in Information Systems Engineering with my approval as supervisor.

Signature:
Name: Asst. Prof. Dr. Kayhan Zrar Ghafoor (Supervisor)
Date: / / 2023

Signature:
Name: Prof. Dr. Shavan Kamal Askar (Co-Supervisor)
Date: / / 2023

I confirm that all requirements have been fulfilled.

Signature:
Name: Byad A. Ahmed
Head of the Department of Information Systems Engineering
Date: / / 2023

I confirm that all requirements have been fulfilled.

Postgraduate Office
Signature:
Name: Byad A. Ahmed
Date: / / 2023

EXAMINING COMMITTEE CERTIFICATION

We certify that we have read this Dissertation: “An Efficient Cucumber Leaf Disease Diagnosis using Deep Learning Algorithms” and as an examining committee examined the student (Saman Muhammad Omer) in its content and what related to it. We approve that it meets the standards of a dissertation for the degree of Doctor of Philosophy in Information Systems Engineering.

Signature:
Name: Prof. Dr. Tarik A. Rashid
Chairman
Date: / / 2023

Signature:
Name: Asst. Prof. Dr. Abdulbasit K. Faeq
Member
Date: / / 2023

Signature:
Name:Asst.Prof.Dr. Bzar Kh. Hussan (Member)
Date: / / 2023

Signature:
Name: Asst. Prof. Dr. Shahab W. Kareem
Member
Date: / / 2023

Signature:
Name: Asst. Prof. Dr. Firas H. Muhammad Amin (Member)
Date: / / 2023

Signature:
Name: Asst. Prof. Dr. Kayhan Z. Ghafoor
Supervisor
Date: / / 2023

Signature:
Name: Prof. Dr. Shavan K. Askar
Supervisor
Date: / / 2023

Signature:
Name: Prof. Dr. Ayad Z. Sabir Agha
Dean of Erbil Technical Engineering College
Date: / / 2023

ACKNOWLEDGMENTS

In the name of Allah, Most Gracious, Most Merciful. First and most thanks and gratitude for ALLAH, Who has made things easy for me. His unwavering support, perseverance, and strength have been helpful in my dissertation accomplishment. I would like to extend my sincere appreciation to Asst. Prof. Dr. Kayhan Zrar Ghafoor, my main supervisor, for his constant encouragement, exceptional guidance, unwavering support, motivation, and friendship. Additionally, I am deeply grateful to Prof. Dr. Shavan Kamal Askar, my co-supervisor, for his insightful guidance, advice, motivation, and friendship. They have significantly contributed in the success of this study.

I am grateful to all my family members, especially my parents for their prayers and moral support. I am also deeply indebted to my wife, for her continuous support and the inspiration during of the study. She has been an excellent companion. Special thanks to my two beautiful daughters, who have inspired me to keep on striving to complete the PhD dissertation.

During the preparation of this study, I was in contact with many people, researchers, farmers, and agricultural experts. They have significantly contributed towards my understanding and thoughts. Special thanks should also be given to Mr. Aso H. Hesamuddin, Assistant lecturer at university of Raparin, Mr. Ghazi F. Pirot, Mr. Wishyar H. Ali, and Mr. Zherko Mahmood for their valuable support and assistance in collecting data for the dissertation. I express my gratitude to University of Raparin (UOR) and Erbil Polytechnic University (EPU) for their support during my PhD dissertation. Special acknowledgment goes to the Artificial Intelligent Research (AIR) Lab at UOR, which served as the foundation for conducting all the experimental work in this dissertation.

ABSTRACT

In agriculture farming, pests and diseases are the most imperative factor that affects cucumber leaves. Farmers around the globe are currently facing difficulty in recognizing various cucumber leaf diseases. Unfortunately, current manual techniques to diagnose and detect cucumber leaf diseases consumes a large amount of human resources, subjective, laborious and exhibits poor real-time performance. Therefore, there is a demanding need for an effective algorithm that enables the diagnosis of cucumber leaf diseases and pests. This dissertation intends to propose and improve a model using deep learning techniques for the diagnosis and detection of cucumber leaf disease. Throughout the study, various challenges and issues have been identified, necessitating resolution. Foremost, reliable public dataset for real-world scenarios involving cucumber leaf disease images are currently lacking. Secondly, there is a need for an efficient convolutional neural network (CNN) algorithm to effectively balance the trade-off between classifying cucumber leaf diseases and performance. Thirdly, you only look once (YOLOv5) model has raised concerns related to time consumption, storage complexity, low detection accuracy, and limited ability to detect small symptom diseases.

Thus, a new dataset of cucumber leaf disease and pest has been constructed that includes two pests (spider, and leaf miner), two fungal diseases (downy mildew, powdery mildew), one viral disease, and healthy class leaves in a real-world scenario. The dataset has a total of 4868 images. Furthermore, this PhD dissertation focuses on proposing a new CNN algorithm with tuning of hyper-parameters to optimize the model's performance that comprises image enhancement, feature extraction, and classification. Data augmentation was used to enlarge the datasets and reduce overfitting, while CNN layers were employed to automatically extract features. Then, five cucumber leaf diseases and one healthy leaf are classified. Moreover, an improved YOLOv5 model for

precise detection of cucumber leaf disease and pest symptoms was developed. With the aim of reducing the model's size, modifications were applied to the model's hyper-parameters. Additionally, the BottleneckCSP module replaced the C3 module in both the backbone and neck network sections. The detection impact was notably enhanced through the reductions in parameters, number of layers, and computations; in addition to that, the improved model demonstrates the ability to detect even small leaf disease and pest spots. Furthermore, the integration of the convolutional block attention module (CBAM) into both the enhanced and standard YOLOv5l models further demonstrates the effectiveness of the proposed model.

The study evaluated the effectiveness of the proposed CNN model by comparing it to pre-trained models (AlexNet, Inception-V3, and ResNet-50). The experimental results confirmed that the proposed CNN algorithm outperformed the other algorithms in recognizing cucumber disease and healthy leaves, based on both datasets with and without data augmentation. The proposed CNN achieves a recognition accuracy of 98.19% with the augmented self-made dataset and 100% with cucumber plant disease dataset. Furthermore, The experimental results of the detection system indicated that the improved YOLOv5 model achieved a mean average precision (mAP) of 80.10%, along with precision and recall rates of 73.8% and 73.9%, respectively. In a comparative analysis, the improved YOLOv5 model demonstrated superior performance to the original YOLOv5l, YOLOv5n, YOLOv5s, YOLOv5m, and YOLOv5x networks. It also achieved significant reductions in storage complexity, decreasing from 92.8 MB to 13.6 MB, and in training time, reducing from 4 hours and 41 minutes to 2 hours and 58 minutes.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATION

Abbreviation	Meaning
AI	Artificial Intelligence
ANN	Artificial Neural Network
AP	Average Precision
AvgPool	Average Pooling
BN	Batch Normalization
BottleneckCSP	Bottleneck Cross-Stage Partial
CBAM	convolutional Block Attention Module
CNN	Convolutional Neural Network
Conc	Concatenation
Conv	Convolutional Layer
CSP	Cross-Stage Partial
CYSDV	Cucumber Yellow Stunting Disorder Virus
DICNN	Dilated And Inception CNN
DL	Deep Learning
DT	Decision Tree
Faster R-CNN	Faster Region-Based CNN
FN	False Negative
FP	False Positive
FPN	Feature Pyramid Network
GAN	Generative Adversarial Network
GLCM	Gray Level Co-Occurrence Matrix
GPU	Graphic Processing Unit
HOG	Histogram Of Gradient
IoU	Intersection over Union
k-NN	K-Nearest Neighbour
LR	Logistics Regression
LSTM	Long Short-Term Memory
mAP	mean Average Precision
Mask R-CNN	Mask Region-Based-CNN
MB	Megabyte
ML	Machine Learning
MLP	Multilayer Perceptron
MYSV	Melon Yellow Spot Virus
NB	Naïve Bayes
NIN	Network-In-Network
NN	Neural Network
P-R	Precision-Recall
PAN	Path Aggregation Network
PBPNN	Propagation Neural Network
PLI	Plant Leaf Image

PRC	Precision-Recall Curve
R-FCN	Region-Based Fully Convolutional Network
ReLU	Rectified Linear Unit
ResNet	Residual Network
RF	Random Forest
SGD	Stochastic Gradient Descent
SIFT	Scale-Invariant Feature Transform
SSD	Single Shot Multibox Detector
SURF	speeded up robust features
SVM	Support Vector Machine
TF	Texture Feature
TN	True Negative
TP	True Positive
YOLO	You Only Look Once
ZYMV	Zucchini Yellow Mosaic Virus

CHAPTER ONE

1. INTRODUCTION

1.1. Overview

Plants are a crucial part of life on Earth as they provide humans with breathable oxygen, food, etc. Furthermore, they provide food for insects and other animals, facilitate weather change, provide clean air, balance the ecosystem, and regulate flooding. In most countries, agriculture crops have become the chief source of economic development. Agricultural productivity also plays a significant role in a country's economic development and the effects of climate on crops can have a major impact on yield and quality. The agriculture sector will provide employment opportunities to many employers in rural areas, contribute to producing food, and be used in medicine and industry. Agriculture plant or crop cultivation has quickly developed in terms of quantity and quality of food production. Due to several factors in the agriculture field, farmers cannot simply control the weather and other environmental conditions that are affecting agricultural crops. Plant diseases and pests are the principal serious and longstanding problem factors that have to be considered in the case of farming practices. It has a devastating effect on interrupting normal plant growth (Shruthi *et al.*, 2019), production quality, quantity, and economic loss. Unfortunately, such diseases and pests are not always detected at an early stage (Faithpraise *et al.*, 2013).

Plants have been reported to have the following organs: leaf, stem, root, fruit, and flower. In agricultural plants, leaves are an important organ of plants for providing information about the amount and nature of gardening crop (Hammad Saleem *et al.*, 2020). Numerous studies have been conducted on plant leaves as a comparative tool for different purposes such as classification and detection. This is attributed to the ease of perception of leaves, given their typically green coloration and flattened structure. Plant disease prevention and control have been broadly discussed because plants are susceptible to diseases and are affected by their outer environment. Normally, plant disease diagnoses

have a significant role in monitoring farming systems accurately (Sun *et al.*, 2018). Cucumber is one of the widely used plant species. The cucumber is a broadly cultivated plant in the gourd family, Cucurbitaceae, and the most globally nutritious and favorite vegetable in the world, which is quick and easy to grow within a short time and easily grows in temperate regions.

In recent years, a deep learning algorithms have been widely used in many different fields in terms of recognition and detection. Deep learning as an area of machine learning has become well known in plant disease recognition and detection. Deep learning algorithms have also been modified by some researchers to enhance the recognition performance of the disease in numerous plant types. CNN is one of the best-performing techniques of deep learning algorithms for image recognition. The significant advantage of deep learning over traditional machine learning methods lies in its capability to automatically extract features from input images during the learning process (Zhang et al., 2021). Despite reducing complicated hand craft engineering process, CNN used to obtain a good performance of classification accuracy (Fujita et al., 2018). CNN has a good performance in terms of cucumber disease diagnosis and leaf classification (Kawasaki et al., 2015), as well as in flower recognition (Omer et al., 2020). In (Zhang et al., 2021) realized that deep CNN in plant diseases recognition is remarkably higher than the traditional algorithms. It also performed better than other machine learning methods to classify normal and diseased cells (Iqbal et al., 2021).

1.2. Problem Background

A wide range of factors affect agriculture production such as occurrence of pests and diseases on crops, which in turn requires increasing food security. It also leads to an increased vulnerability to diseases and pests, which can reduce crop yields and affect food security. Plant diseases and pests are among the

problems that affect crop yields and usual plant growth. They may be even more devastating and appear on plant leaves, causing economic losses, production quality, and quantity (Yang *et al.*, 2022)(Li, Ahmed, *et al.*, 2022). Plant pathogens (bacteria, fungi, and virus diseases), deficiency, plant nutrition (lack of microelements), pests, and insect feeding (sucking insect pests) cause leaf batches (Berger, 1985). Climate change also affects the availability of water for agriculture, which can affect crop growth and productivity. Plants experience severe stress or damage, leading to loss of their leaves and eventual death. Researchers have been motivated by these works to apply technology from computer vision and artificial intelligence disciplines to the diagnosis, detection and management of plant diseases and pests.

To respond to plant disease detection and identification problems, an efficient recognition and detection system for automatic symptom leaf disease identification is essential. Plant disease identification is an important mechanism for preventing plant diseases in a complicated environment. Farmers often recognize the symptoms of plant diseases using traditional means, for example, by making naked eye observations and referring to the information in books, internet, etc. (Shruthi *et al.*, 2019). Furthermore, traditional methods like microscope and Deoxyribonucleic acid (DNA) sequencing-based approaches have been used to classify and detect various types of diseases. Such methods, however, necessitate experienced experts in farming; and many farmers are not even permitted to use advanced tools, though most of them own a smartphone for capturing images (Amara *et al.*, 2017)(Lu *et al.*, 2017).

Cucumbers are affected by diseases due to nonbiological factors, the impact of various factors, and a bad ecological environment. This has a major economic impact on farmers, yield production, and quality. It has a property to be suffered from high disease occurrence, frequent and fast infection. The leaf, fruit, stem, and root are parts of the cucumber plant that are affected by different

kinds of diseases. Leaves are considered the best part to be used for disease diagnosis because of the appropriate macro environment (El-Helly et al., 2003); the symptoms of disease affection are visually apparent on leaves due to size, shape, and color (Zhang et al., 2017) (Ganatra and Patel, 2020).

Cucumber diseases are diagnosed through visual analysis by experts and biological inspection, this technique is time-consuming, inefficient, expensive (Shen *et al.*, 2008), and not the best way to recognize diseases (Sannakki *et al.*, 2013). Expert diagnoses method is used which depends on highly expert experience but it has low accuracy, in addition to that, pathogen analyses method is used that involves the cultivation and microscopic observation of pathogens, it is time consuming. These methods are complicated and require a lot of effort to complete. Chemical methods (these methods can include using indicators or stains that react with particular substances, like pathogens or nutrients, leading to visible changes in the leaf tissue or color) are also used to diagnose leaf injuries, which are not real-time diagnoses because of necessitating a lot of experiments (Bai *et al.*, 2017). Those lead to affect agricultural production in terms of increasing the danger of toxic residue levels and cost because of excessive use of pesticides for the treatment of plant disease in case of incorrect disease identification. In the meantime, early discovery of diseases, immediate attention, early diagnosis and detection, and avoiding infection are the most major efforts that have to be carried out by farmers to reduce damages and increase their income (Tani et al., 2018) (Ma et al., 2018).

With the development of technology and artificial intelligence in agriculture, computer vision and machine learning algorithms have become significant tools in terms of diagnosis and detection of cucumber leaf diseases. Additionally, intelligent systems based on computer vision have been used in agriculture for the purpose of achieving efficiency and increasing productivity (Tian *et al.*, 2020). Hence, in the agriculture field, computer vision technology with hardware development such as graphics processing units (GPUs) are

applied on the crop growth state monitoring, agricultural products quality examination and categorization, plant disease and insect pest identification (Shen *et al.*, 2008) (Tian *et al.*, 2020). In order to address the issues, this requires an efficient and accurate cucumber leaf disease diagnosis and detection model. Researchers are diligently working on developing computer-based algorithms to monitor vast crop fields and identify disease and pest symptoms. Their objective is to design a precise recognition system that can be accessible to a wide range of farmers. In this scenario, an accurate and timely computer-based diagnosis algorithm for cucumber leaf diseases and pests is essential that would be capable of disease recognition in a better, more reliable, and faster way.

1.3. Problem Statement

The investigation problems in this dissertation can be summarized as follows:

i. Despite the availability of public datasets for various plant species within the agricultural domain, there is a scarcity of reliable datasets dedicated to cucumber leaf diseases. Furthermore, one specific pest type, known as the spider, is notably non-existent from these datasets.

ii. The misdiagnosis of diseases and pests has negative effects on cucumber crops. The use of deep learning methods introduces complexity due to the involvement of numerous hyper-parameters during network training, thus intensifying the challenge. Furthermore, these algorithms demand fine-tuning and updates. Proposing a new CNN is challenging due to the need to determine optimal hyper-parameter values, such as learning rate, batch size, layer number, filter sizes, and dropout rates.

iii. Timely and the early detection of symptoms related with cucumber leaf diseases is challenging due to the presence of small disease spots. Improving a model to increase detection results while reducing training time and storage complexity is a particular manner to address this challenge in model development.

During the investigation of this research, several questions have arisen. The research questions to be explored are as follows:

- Is the availability of publicly available datasets adequate for cucumber leaf disease and pests?
- How does the influence of hyperparameter tuning impact the generalization of CNN models to achieve improved accuracy in diagnosing cucumber leaf diseases in this research study?
- How could a lightweight YOLOv5 model be enhanced to proficiently identify and precisely locate symptoms of cucumber leaf diseases and pests?
- Is there any possibility to reduce training time consuming and model parameter weight size in the area of YOLOv5 model in deep learning?
- Is it possible to enhance the accuracy in leaf disease detection by improving feature extraction and representation in the YOLOv5 model?

1.4. Research Objectives

The specific aims of this research are outlined below:

I. To come up with a new cucumber leaf disease image dataset generated from farms in the Kurdistan region, Sulaymaniyah, Rania. It will be available as a standard public dataset for the research community.

II. To develop and fine-tuned a new CNN model capable of diagnosing cucumber leaf disease through the utilization of hyper-parameter tuning to overcome the misdiagnosis issues.

III. To improve a lightweight YOLOv5 model to address training phase time consumption challenges and reduces storage requirements, with focus on overcoming the YOLOv5 limitation related to the detection of small leaf disease and pest symptoms.

IV. The intended models is expected to possess the desired attributes: adaptability, efficacy, and precision.

1.5. Research Contributions

The key contributions of this research can be outlined as follows:

- A new structured cucumber leaf disease image dataset is collected from farms in the Kurdistan region.
- A new CNN algorithm has been proposed by utilizing model hyper-parameters tuning, layer modifications. The proposed model yields faster and improved results.
- A lightweight YOLOv5 model has been improved, with hyper-parameters modifications to increase its detection accuracy. Firstly, this enhancement involves the replacement of the C3 module in the backbone and neck network sections with the BottleneckCSP module. This modification results in a reduction in the number of layers and parameters, ultimately leading to reduced time consumption and storage complexity. Secondly, the last backbone convolutional layer has been removed. Finally, the model efforts to overcome the challenge of enhancing feature representation through the incorporation of the CBAM into the improved YOLOv5 model.

1.6. Research Scope and Assumption

The scope of the dissertation covers the following:

I. In the constructed dataset, a total of five different types of diseases and pests were included, along with healthy cucumber leaves. Additionally, one publicly available dataset was used.

II. A new CNN model has been utilized to classify five disease and pest types with one healthy leaf. Moreover, as an evidence for evaluation of the proposed CNN model from scratch, three representative pre-trained deep learning recognition algorithms are used.

III. YOLOv5 model has been improved for automatic cucumber leaf disease symptom detection. in addition, The CBAM module, serving as a channel attention mechanism, is incorporated into the improved YOLOv5l model.

IV. MATLAB and Python environment have been used to conduct recognition and detection tasks, respectively.

1.7. Organization of the Research

This PhD dissertation is structured into six chapters as shown below: it begins with an introduction chapter, and the subsequent chapters are organized as follows:

- Chapter 2: In this chapter, a theoretical background, literature review and various related research techniques are explored, along with their working procedures.
- Chapter 3: The materials, methodology, collecting data and experimental setup are discussed in detail.

- Chapter 4: This chapter investigates into cucumber leaf disease and healthy recognition using deep learning algorithms. It presents the proposed new CNN algorithm, along with the analysis and presentation of performance evaluation results. Additionally, the examination of three pre-trained models is discussed. The chapter investigates the details of these approaches and their impact on improving the overall efficiency of the recognition system.
- Chapter 5: This chapter is dedicated to enhancing the effectiveness of the detection and identification system. It entails the implementation of a cucumber leaf disease and pest detection system to identify, detect, and localize symptoms. The chapter also presents and analyzes the results of conducted experiments and result evaluation. Moreover, it explores the influence of integrating CBAM into the improved model and assesses the overall efficiency of the enhanced system.
- In the final (chapter 6), the research conclusions are presented by summarizing the key contributions made throughout the research. It also offers suggestions for future work that can further advance the field.

CHAPTER TWO

2. THEORETICAL BACKGROUND AND LITERATURE REVIEW

2.1. Introduction

This chapter starts with a state-of-the-art CNN algorithm in the process of diagnosing, identifying, and detecting of cucumber plant leaf pest and diseases. It also presents some issues that face the models performance, and also indicates gaps that should be addressed in the future. A focused review is closely related to the diagnosis plant leaf disease and pest systems, specifically cucumber leaves. Moreover, it discusses the commonly employed methodology steps for plant disease and pest recognition. It starts with data acquisition, pre-processing, feature extraction, and classification. It also discusses existing various deep learning architectures-based solutions for plant disease recognition. Then, this chapter discusses cucumber leaf disease recognition and detection system design and development. Finally, the chapter ends by highlighting the research issues.

2.2. Traditional Pattern Recognition Workflow Steps

Many approaches have been used in agriculture domain for automatic plant disease recognition in various plant parts such as fruit, root, stem, and leaf. The system operates through four different stages, namely data acquisition, pre-processing, and the combination of feature extraction and classification within the same model architecture.

2.2.1. Data Acquisition

The first step in plant leaf disease classification and detection system is image acquisition. A wide variety of devices such as digital camera and smart phone camera can be used to capture images of healthy and diseased plant leaves.

2.2.2. Data Pre-Processing

The pre-processing step within machine learning and deep learning techniques plays a crucial role in constructing an effective dataset to develop generalizability of the model. In deep learning, a huge amount of data must be collected from different sources such as physical devices, tools, software programs like web crawlers, manual surveys, etc. The model performance may be affected during data collection because of hardware faults, software problems, tool failures, noise, and human errors. Data pre-processing might solve problems such as data not fitting into memory and local storage. It may also help visualize and accelerate the process.

Data pre-processing has an important effect on the performance of a supervised machine learning model. It can solve several kinds of problems on data using transformation, cleaning, normalization, feature extraction, and feature selection before being fed as input to the machine learning or deep learning models (Kotsiantis et al., 2006). Removing background noise and suppressing undesired distortions have been shown as pre-processing to advance some image features and make the input suitable for further processing (Shruthi et al., 2019)(Oo and Htun, 2018). To boost the reliability of their model, Sladojevic et al. proposed a method to pre-process input images by cropping them manually, thereby highlighting region of interest by creating the square around the leaves (Sladojevic et al., 2016).

In addition, Lu et al. resized an image from 5760×3840 into 512×512 RGB image to reduce the running time and dimension of training data (Lu et al., 2017). In another work, Ashqar et al. pre-processed input images by resizing them to 128×128 pixels, normalizing the pixel values to a [0,1] range, and balancing dissimilar classes (Ashqar et al., 2019). In (Chen et al., 2020), photoshop tools have been used to equally process images into RGB model for computations, and then these images are resized to 224×224 pixels. Table 2.1

shows various approaches for image pre-processing. Furthermore, data augmentation is another technique in pre-processing step as explained in Section 3.6.

Table 2. 1 Investigation of pre-processing techniques applied in plant identification

Authors	Pre-processing methods/Purposes
(Amara *et al.*, 2017)	images were resized and converted into grayscale
(Lu *et al.*, 2017)	Images were resized to smaller size to reduce a running time and dimensions
(Sladojevic *et al.*, 2016)	cropping all images manually and draw a square around the region of interest leaves
(Ashqar *et al.*, 2019)	Resized, normalized, and balance dissimilar class of image
(Chen *et al.*, 2020)	Photoshop tool used to equally processed images and resized images
(Nagasubramanian *et al.*, 2019)	RGB images transformed to HSV color spaces

2.2.3. Feature Extraction

In pattern recognition, image features play a significant role and are part of an object in image to identify it. Features, generally, describe image properties such as corners, edges, regions of interest points, ridges, etc. In plant disease recognition, color, shape, and texture have been used as characteristic descriptors to discriminate between plant object (foreground), and other unrelated objects (background). Image texture feature defines how the patterns of color are dispersed in an image. Image color feature is used to discriminate one disease from another. Moreover, due to diseases that have different shape features which are area, axis, and angle, they are used to discriminate diseases (Panigrahi et al., 2020).

While traditional pattern recognition approaches adopt handcrafted features, deep learning automatically adapts features in a better and modernized way

from a huge dataset (Karthikayani and Arunachalam, 2020). The latter is categorized as a group of machine learning algorithms wherein input layers are basically mapped onto output layers (Mohanty et al., 2016). Such methods involve various layers of nonlinear processing units for extracting and adapting features. All sequential layers use the previous layer's output as an input (Benuwa et al., 2016). Figure 2.1 illustrates a process of feature extraction comparison between traditional and deep learning models.

In a CNN architecture, feature extraction and the classification process are combined in the same model. CNNs use multiple feature extraction stages and avoid the complicated feature extraction procedure, and to learn the specific features more efficiently.

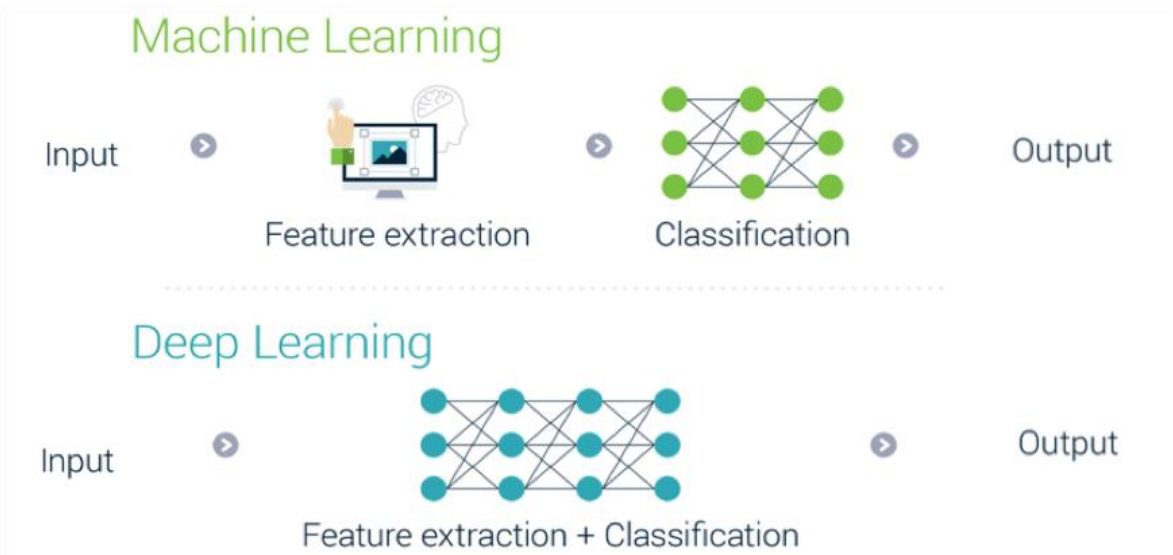


Figure 2. 1 Feature extraction procedure between a traditional Machine and deep Learning (Analytics Vidhya, 2020)

2.2.4. Classification

Classification is generally accomplished using fully connected layer with an activation function SoftMax, in which a computer program uses learned

features from input data to categorize the same into predefined classes (Amara et al., 2017) and uses various collections of features (Dara and Tumma, 2018). Many classification techniques have been used in agricultural domain for investigating plant diseases and pests. Traditional machine learning methods have been extensively implemented in agricultural domain. Additionally, the deep CNN techniques have been applied for object identification and plant disease categorization and have witnessed tremendous developments in past years. Deep learning has been extensively considered for computer vision tasks in recent years, and thus, a huge number of related techniques have been developed. Although it has been proven to be effective in different classification and detection problems, it is very challenging to grasp unknown objects because of the different shape and posture of objects (Jiang et al., 2021).

For example, LeNet model as a CNN has been used in (Amara et al., 2017) to classify two banana leave diseases, namely banana speckle and sigatoka. The authors of (Liu et al., 2017) designed a novel deep CNN architecture for accurately classifying four different types of apple diseases such as mosaic, rust, brown spot, and Alternaria leaf spot. They used a dataset of 13,689 images of unhealthy apple leaves and obtained overall accuracy of 97.62%. Lu et al. developed an innovative CNN-based identification method to categorize 10 common rice diseases. Using this model, they attained an accuracy of 95.48% on a dataset including 500 images of unhealthy and healthy rice leaves and stems (Lu et al., 2017). The authors of (Ashqar et al., 2019) selected a CNN (ConvNet-based) approach for classifying plant seedlings with a dataset containing approximately 5,000 images belonging to 12 different species. Table 2.2, summarizes that several algorithms had been carried out for plant leaf disease classification.

Table 2. 2 Summarizing various studies for plant disease classification

Authors	Methods	dataset	Accuracy(%)
(Hammad Saleem *et al.*, 2020)	Faster R-CNN, R-FCN with ResNet ,and SSD with Inception	PlantVillage	73.07
(Amara *et al.*, 2017)	LeNet	PlantVillage (2 types of banana leaves diseases)	97.57
(Lu *et al.*, 2017)	CNN	rice diseases (10 common diseases)	95.48
(Sladojevic *et al.*, 2016)	CNN (CaffeNet)	PlantVillage (13 different diseases)	96.3
(Ashqar *et al.*, 2019)	CNN (ConvNet)	plant seedling dataset	99.48
(Chen *et al.*, 2020)	VGGNet and Inception	rice plant images	92.00
(Nagasubramanian *et al.*, 2019)	Supervised 3D-CNN	4 soybean genotypes were selected from soybean stem samples	95.73
(Barbedo, 2018)	CNN (GoogLeNet)	freely available dataset contains almost 50,000 images	94 from separate lesions and spots
(Mohanty *et al.*, 2016)	AlexNet and GoogLeNet	PlantVillage	99.35 from GoogLeNet
(Durmus *et al.*, 2017)	AlexNet and SqueezeNet	PlantVillage (tomato leaves)	95.65 from AlexNet
(Liu *et al.*, 2017)	CNN (Goo- gLeNet)	Apple images (4 common types of apple disease)	97.62
(Fuentes *et al.*, 2017)	Faster R-CNN, R-FCN, and SSD with ResNet	PlantVillage (tomato leaves)	88.20 from RFCN with ResNet 50
(Ferentinos, 2018)	AlexNet , Overfeat , AlexNetOWTBn, GoogLeNet ,and VGG	openly available database contains 87,848 images	99.53 from VGG

2.3. Challenges in Plant Disease Classification and Detection

The domain of plant disease classification and detection faces some formidable challenges. Tackling these difficulties demands association among researchers, application of specialized models, and the continuous development

of advanced techniques to improve the accuracy and efficacy of plant disease identification systems. Issues and challenges that have been recognized in this study are hyper-parameter tuning, model overfitting, and plant organs, as shown in Figure 2.2.

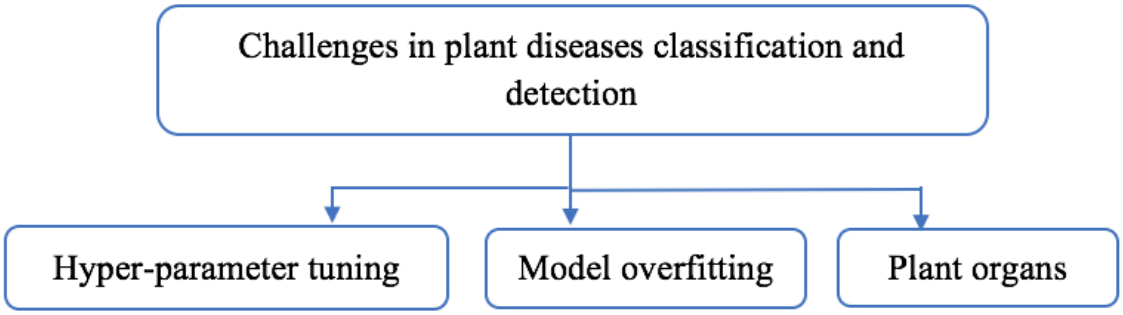


Figure 2. 2 Challenges in Plant Disease Classification and Detection

2.3.1. Hyper-parameter Tuning

Deep learning models automatically extract image characteristics or features and categorize objects based on these extracted features. Traditional machine learning models, on the other hand, manually extract features and tune them. Throughout the training and testing of the model, a set of parameters, for learning process, known as hyperparameters, are used. A large set of hyperparameters are used in various deep learning architectures (Hutter et al., 2015).

In every dataset, hyperparameter tuning has a significant effect on training the model to obtain a good performance and develops validation errors (Victoria and Maragatham, 2021). Hyperparameters include the parameters of (i) regularization, (ii) network architecture, such as layer numbers and sigmoid transfer function kinds, (iii) sample numbers and learning rates, (iv)

preprocessing, such as reducing dimensionality, and normalization, and (v) initialization weight parameters. Theoretically, several methods have been treated using hyper-prior and manually using optimization techniques.

Some of these hyperparameters pose a greater challenge of grounded mathematical treatment (Hutter et al., 2015). In such instances, hyperparameter tuning of a deep learning architecture is an issue that must be addressed based on empirical data using improving theoretical background and evaluating the performance of the network (Angelov and Sperduti, 2016), such as shown in Figure 2.3 (Analytics vidhya, 2020).

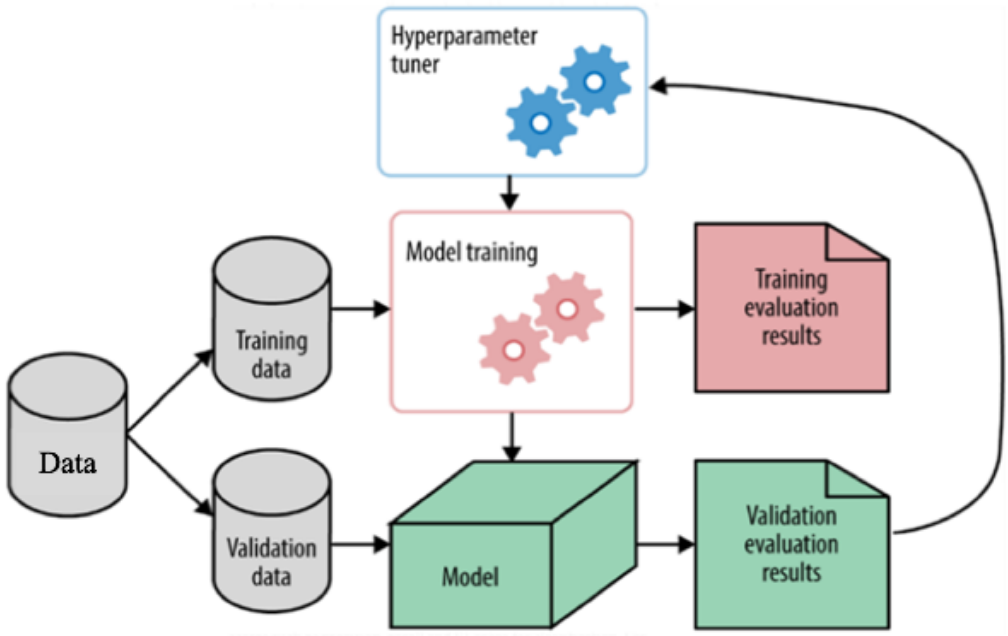


Figure 2. 3 Hyper-parameter tuning process (Analytics vidhya, 2020)

For instance, a simplified system or improved technique would require less hyperparameters. On the other hand, a complex system can be customized automatically using hyperparameter optimization algorithms in a given application for optimal performance (Hutter et al., 2015). In (Angelov and

Sperduti, 2016) a Bayesian hyperparameter optimization technique proposed for improving model performance, where all hyperparameter values are optimized.

2.3.2. Model Overfitting

Overfitting is an issue facing machine learning algorithms, especially deep learning models, in which errors or random noise occur rather than the underlying relationship described in the model (Liu et al., 2017). Overfitting has been shown to have a negative effect on robust performance of the training set across multiple datasets such as ImageNet, CIFAR-10, CIFAR-100, and SVHN (Rice et al., 2020), as shown in Figure 2.4. Liu et al. employed several techniques to avoid overfitting. They used dataset augmentation operations such as mirror symmetry, image rotation, PCA jittering, and brightness adjustment to increase the diversity of training images and enhance the generalizability of their model (Liu et al., 2017).

The authors of (Rice et al., 2020) studied data augmentation and regularization techniques to remedy overfitting. Their experimental testing showed that regularization methods do not robustly prevent overfitting and tend to make the model over-regularized. Also, the authors of (Arsenovic et al., 2019) used two different augmentation algorithms to prevent overfitting, namely traditional augmentation methods, like pixel-wise changes or rotations, and training using generative adversarial network (GAN). In (Liu et al., 2017) convolution layers were utilized in place of certain fully connected layers, and they also used local normalization utilizing response-normalization layers. Furthermore, retraining the last few layers while freezing the first layer is another way to reduce overfitting through using transfer learning (Barbedo, 2018). The authors of (Mohanty et al., 2016) changed the data ratio of train and test sets. In addition, two different methods such as training the network model using more examples and changing network complexity like changing structure

and parameters of the network have been used to reduce overfitting (Brownlee, 2018). In neural network, dropout means removing units from the network temporarily along with outgoing and incoming connections during training process. Srivastava et al. used the dropout algorithm for resolving the overfitting problem. They noted that this technique can provide a significant development over regularization algorithms and markedly reduce overfitting (Srivastava et al., 2014).

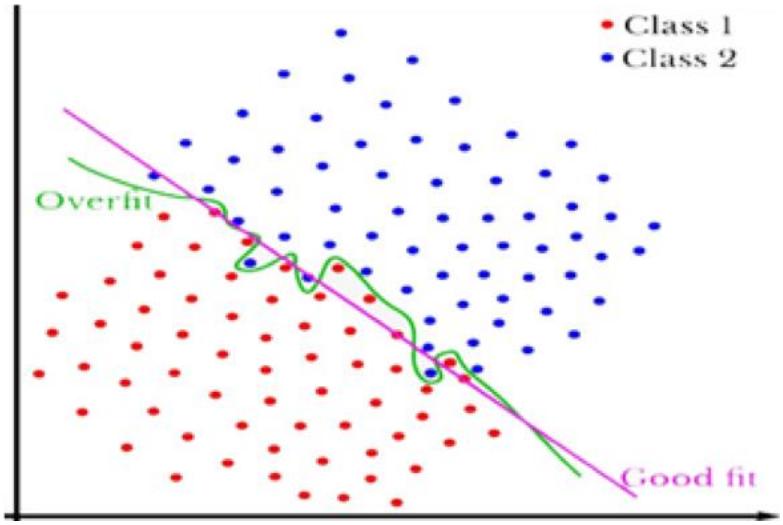


Figure 2. 4. Overfitting in machine learning (Mubasir, 2020)

2.3.3. Plant Organs

Plants have various organs that have been used as a characteristic to be studied by researchers in various fields, especially in disease recognition and detection task. Based on this review, leaf plant organ had been mostly used by researchers such as (Hammad Saleem *et al.*, 2020) (Amara et al., 2017) (Ashqar et al., 2019) (Nagasubramanian et al., 2019) (Dara and Tumma, 2018) (FatihahSahidan et al., 2019)(Ferentinos, 2018)(Ramcharan et al., 2017)(Victoria and Maragatham, 2021)(Angelov and Sperduti, 2016), for the

purpose of classifying and detecting plant diseases. In (Arnal Barbedo, 2019), instead of using entire leaf, separate spot and lesions have been used.

However, many diseases have been better categorized in other organs using their symptoms. For instance, the stem has been used in (Lu et al., 2017)(Barbedo, 2018). Also, in (Ashqar et al., 2019) seeding has been used for classification. Hence, a comprehensive plant image dataset must be constructed to incorporate images of other plant organs and better classify plant diseases.

2.4. Deep Learning Models for Object Classification and Detection

In recent years, artificial intelligence (AI) and machine vision algorithms have advanced significantly, leading to the development of new processing methods and computer vision technologies. This has resulted in numerous applications such as healthcare, finance, agriculture, and other complex scene applications (Chen et al., 2022)(Yang et al., 2022). AI is used to improve efficiency, accuracy, and overall performance. With the popularization of intelligent agriculture, agricultural intellectualization is developing rapidly.

The agriculture domain has witnessed massive developments with the aid of technology. Image processing and object detection methods have been used for detecting the infected region in the plant. In addition to their simplicity and accuracy, such techniques are fast (Shruthi et al., 2019) (Panigrahi et al., 2020). Hence, advancements in computer and internet technology can help addressing the problem of automatic plant disease and pest recognition. Such developments are essential in scientific research for classifying and detecting the symptoms of plant diseases and pests automatically by using innovative and intelligent techniques (Bashish et al., 2011).

In last decades, deep learning techniques have been emerged as highly effective machine learning methods for recognizing and detecting objects, particularly in plant disease diagnosing systems. Unlike traditional approaches

that rely on techniques such as scale-invariant feature transform (SIFT), HOG, and speeded up robust features (SURF), deep learning methods possess the ability to automatically and rapidly learn features directly from the raw pixel data of input images. Deep learning methods employ a layer-by-layer approach, where local receptive fields gradually expand. Fine features were extracted using the low level layers like lines, borders, and corners, while higher features were extracted using the higher-level layers such as specific parts of objects or the entire object itself. Essentially, deep learning makes it possible to represent objects at various levels of detail, providing a complete end to end picture. Deep learning architectures used to identify and detect diseases in plant leaves are shown in Figure 2.5.

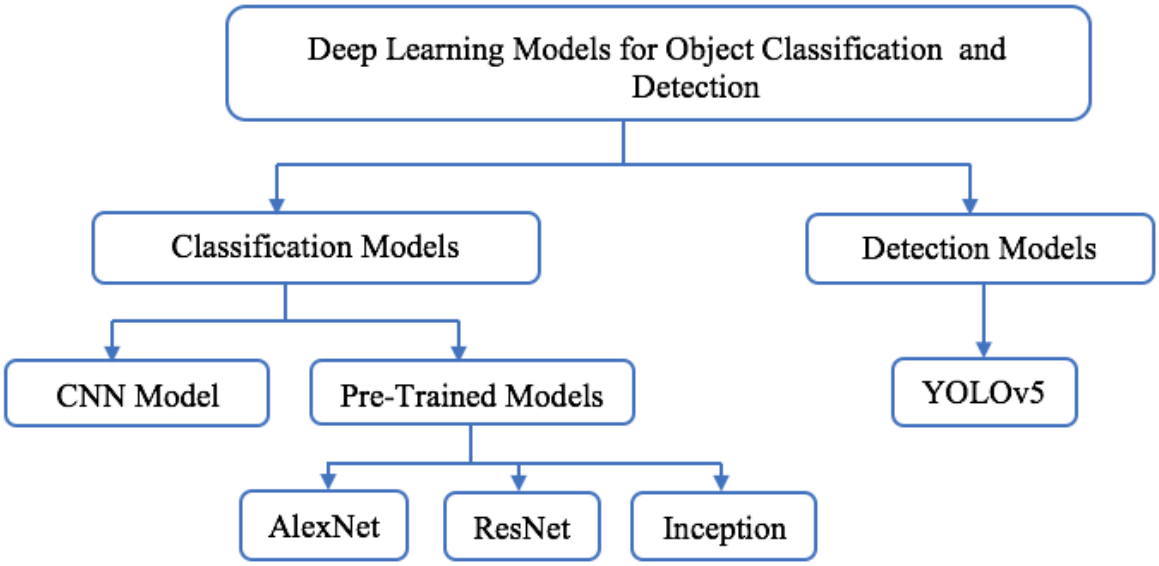


Figure 2. 5 Deep learning models for object classification and detection

2.4.1. Classification Models

In recent times, numerous techniques were used and developed by researchers in the field of agriculture with the goal of identifying plant diseases

based on image processing and pattern recognition methods. In agriculture field, inspection systems based on computer vision have play a significant role as a tool and its use has greatly increased. Tian et al. concluded that combining computer vision with artificial intelligence approaches would improve an agricultural automation systems in terms of general performance, economic performance, robust performance, and coordination performance (Tian et al., 2020).

A. Convolutional Neural Network (CNN)

Over the past few years, CNNs have become highly effective in representing images for different category-level recognition tasks, including object classification, and object detection. Although the fundamental principles of CNNs were established in the 1980s, in the initial stages of the 1990s, CNNs began to appear with the drawback of a large and complicated computational loads. Nowadays, with the combination of enhanced GPUs and the availability of extensive labeled image datasets, CNNs have become increasingly favored as highly effective tools for extracting features and classifying data (Arandjelovic *et al.*, 2018) (Uçar *et al.*, 2017). In a relatively short period, CNNs have achieved notable success in numerous domains of computer vision, like autonomous vehicles, speech recognition, medical imaging, and plant disease diagnosing.

A CNN is a stack of nonlinear transformation functions and can automatically learn representations from the data in order to use the numerous feature extraction steps (Khan, Sohail, *et al.*, 2020)(Reyes et al., 2015). CNN is a particular kind of feed-forward neural network (information is fed from layer to layer without reversing) (Zbakh *et al.*, 2019) and is motivated by biological processes that occur in the visual cortex in the living beings of mind. In the 1980s, CNNs were initially proposed for digit recognition (LeCun et al.,

1989). Recently, CNN-based deep learning architectures have enabled huge-scale object recognition tasks.

CNNs are capable of extracting features hierarchically and classifying them (Khan, Sohail, *et al.*, 2020). A CNN has several layers that hierarchically calculate features from images as an input. CNN architecture consists of multiple layers. CNNs can be formed through various combinations of convolution layers, a pooling layer, an activate function layer, and a fully connected layer at the end, as shown in Figure 2.6. Convolutional and pooling layers act as feature extractors (Amara et al., 2017). The layer specifications of CNNs are briefly outlined as follows:

Input layer: The images are directly inputted into the network.

Convolution layers: In CNN, the primary operation is carried out by the convolutional layer, which is responsible for learning feature representations from the inputs. Unlike conventional feed-forward neural networks that use matrix multiplication, CNNs utilize convolution to reduce the number of weights and overall network complexity (Uçar et al., 2017)(Gu et al., 2018). Additionally, convolution layers consist of multiple convolution kernels and produce a feature map by extracting features of an input image using a filter or kernel (Ibrahim et al., 2018). Specifically, each neuron in a feature map is connected to a neighboring region of neurons in the preceding layer. The input image convoluted convolution with the learnable filters or kernels, resulting in a feature map in the output image. Subsequently, an element-wise nonlinear activation function is applied to the convolved outcomes (Gu et al., 2018).

The Convolutional layer keeps the outcomes of the convolution of filters or kernels of the preceding layer (Durmus et al., 2017). These filters or kernels to be learned contain weights and biases; all filters are restricted spatially but expand with comprehensive depth of input volume (Dara and Tumma, 2018). The kernel (window) slides over the entire image step by step. The result is taken from summation over the entire image (Zbakh *et al.*, 2019). Different

feature maps are yielded from multiple convolutional layers and different filters to ensure complete extraction of various features.

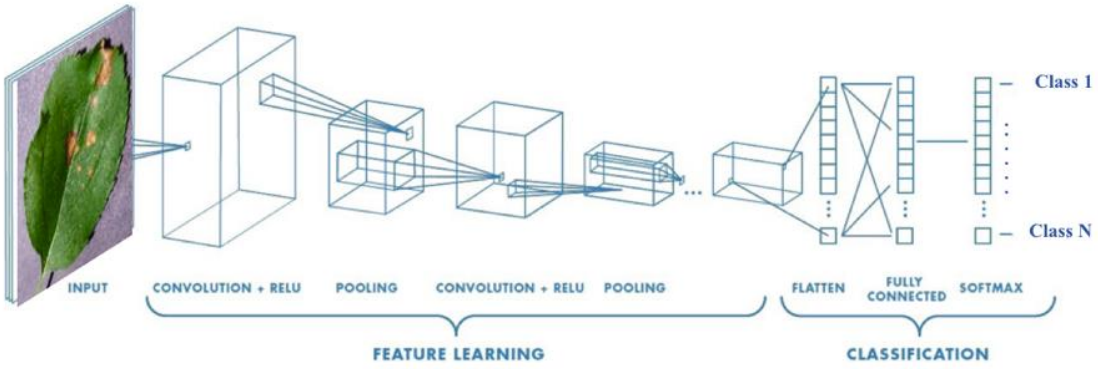


Figure 2. 6 CNN model architecture (Prabhu, 2018)

Activation function: The activation function has a significant role in the learning process, and thus, selecting a proper activation function would affect the training dynamics and task performance (Ramachandran et al., 2017). An activation function in CNNs introduces nonlinearities that are essential for enabling multi-layer networks to detect nonlinear features effectively (Gu et al., 2018). Various activation functions have been used to inculcate nonlinear combination of features (Khan, Sohail, *et al.*, 2020) and to increase nonlinearity of the network (Durmus et al., 2017). Commonly used activation functions include ReLU, sigmoid, and tanh. The most commonly used function is ReLU, which is a piecewise linear function in which all negative pixel values are replaced by zero, while positive pixel values are retained, as explained in equation (2.1) (FatihahSahidan et al., 2019)(Gu et al., 2018)

$$\mathrm{Relu}(x) = \begin{cases} 0 & \text{if } x < 0 \\ x & \text{if } x \geq 0 \end{cases} \qquad (2.1)$$

Pooling layer: It has a significant concept after activation function to obtain a strong feature versus noise and distortion (Saufi et al., 2018). The purpose of the pooling layer is to attain shift-invariance by reducing the resolution of feature maps due to reducing the dimensionality of features. It is also used to decrease the connection numbers between convolutional layers, reduce the sampling size (Ibrahim et al., 2018)(Gu et al., 2018), reduce neuron size, and reduce overfitting (Durmus et al., 2017). The pooling layer works independently over the entire input depth to rescale it. Typically positioned between two convolutional layers, each feature map in the pooling layer is connected to the corresponding feature map from the preceding convolutional layer (Gu et al., 2018) (Dara and Tumma, 2018). This layer facilitates faster convergence, improved generalization, and a degree of invariance to translation and distortion. Commonly used pooling operations include max pooling and average pooling. The output dimension size of pooling calculated based on the formula defined in the following equation (2.2) (Layton, 2019):

$$P_{out} = \left(\left(\frac{D_{in} - P_f}{S}\right) + 1\right) \qquad (2.2)$$

Where *Pout* is the pooling output dimension, *Din* is the input image dimension, *Pf* is the pooling filter size dimension, and *S* is the pooling stride number.

Fully connected layer: There is a possibility of the existence of one or multiple fully connected layers, after multiple convolutional layers, activation functions, and pooling layers, which serve the purpose of performing high-level reasoning. These layers exhibit similarities to the layers commonly seen in traditional feed-forward neural networks. They connect all neurons from the former layer to each individual neuron in the current layer to generate comprehensive semantic information. This layer acts as a final feature selector.

The outputs are computed using matrix multiplication and bias addition. Additionally, similar to conventional feed-forward neural networks, the weights of these layers are estimated by minimizing solely the training error (Uçar et al., 2017) (Gu et al., 2018).

Output layer: The output layer of a CNN is the final layer. The SoftMax function is commonly used in classification duties.

The training of a CNN involves solving a global optimization issue, aiming to reduce the loss function and attain the optimal set of parameters. The methods of stochastic gradient descent (SGD) and Adam are frequently employed for the improvement of CNN networks. The training process consists of several steps: First, the input data is fed forward through the network, passing through different layers. Second, the output values are computed by extracting meaningful features using digital filters at each layer. Lastly, the discrepancy between the network's predicted output and the actual output is computed as the error, which is subsequently reduced through backpropagation. This involves propagating the error backward through the network. By adjusting the weights of the CNN, the network's performance is fine-tuned and optimized (Uçar et al., 2017) (Gu et al., 2018). Generally, the learning process as end-to-end in CNNs enables a direct mapping from raw input image data to the target class without requiring prior knowledge or human intervention and external guidance.

B. Pre-trained Models

Transfer learning is the process of reusing a pre-trained model for solving a new problem that is different from scratch, which involves learning or training data from basic. For instance, the authors of (Chen et al., 2020) studied transfer learning of deep CNN to classify diseased leaves. They chose VGGNet and Inception models for improving the learning capability of small lesion signs.

The authors of (Arnal Barbedo, 2019) used a pre-trained CNN that employed GoogLeNet architecture to study the use of separate spot and lesions, instead of using whole leaves and classified various plant infections. They concluded that the accuracy attained from separate lesions and spots was 94%.

Mohanty et al. evaluated and focused on two famous deep CNN models, namely AlexNet and GoogLeNet, trained using scratch and transfer learning, to classify 14 crop classes and 26 diseases. They noted that GoogLeNet reliably performs better classification based on training transfer learning on images of unhealthy and healthy leaves, and attained an accuracy of 99.35% (Mohanty et al., 2016). In addition, Nagasubramanian et al. improved a technique named a supervised 3D-CNN for learning the spectral and spatial information of hyperspectral images of healthy leaves and charcoal rot disease categorization examples in soybean stems. They explained the significance of specific hyperspectral wavelengths in categorization using a saliency map-based visualization technique and obtained a 95.73% classification accuracy (Nagasubramanian et al., 2019). A state-of-the art CNN model from scratch proposed in (Omer *et al.*, 2022) to diagnose five cucumber leaf diseases and one healthy leaf. Comparative experiments were conducted based on applying pre-trained models (AlexNet, Inception-V3, and ResNet-50) to prove the authenticity of the proposed CNN. The pre-trained models were trained utilizing transfer learning. All model weights initialization were obtained from ImageNet.

a) AlexNet

The AlexNet model was introduced by Krizhevsky. It is a deep CNN architecture that made significant advancements within the realm of computer vision, particularly in the domain of image classification tasks. This model gained recognition by winning the imageNet large scale visual recognition

challenge (ILSVRC) in 2012. To train the AlexNet model, a large dataset called ImageNet, consisting of 1000 different labeled classes, was used. The architecture of the AlexNet model comprises eight layers, including five convolutional layers with some following pooling layers, as well as three fully connected layers. It is composed of approximately 650,000 neurons and has around 60 million parameters (Krizhevsky et al., 2012). The AlexNet model structure has shown in Figure 2.7. By incorporating convolutional layers, the network gains the ability to directly learn hierarchical features from the row pixel data. Comparing compression on test data to previous state-of-the-art techniques as an evaluation, the AlexNet model worked better and achieved top-1 and top-5 error rates of 37.5% and 17.0%, respectively (Krizhevsky *et al.*, 2017).

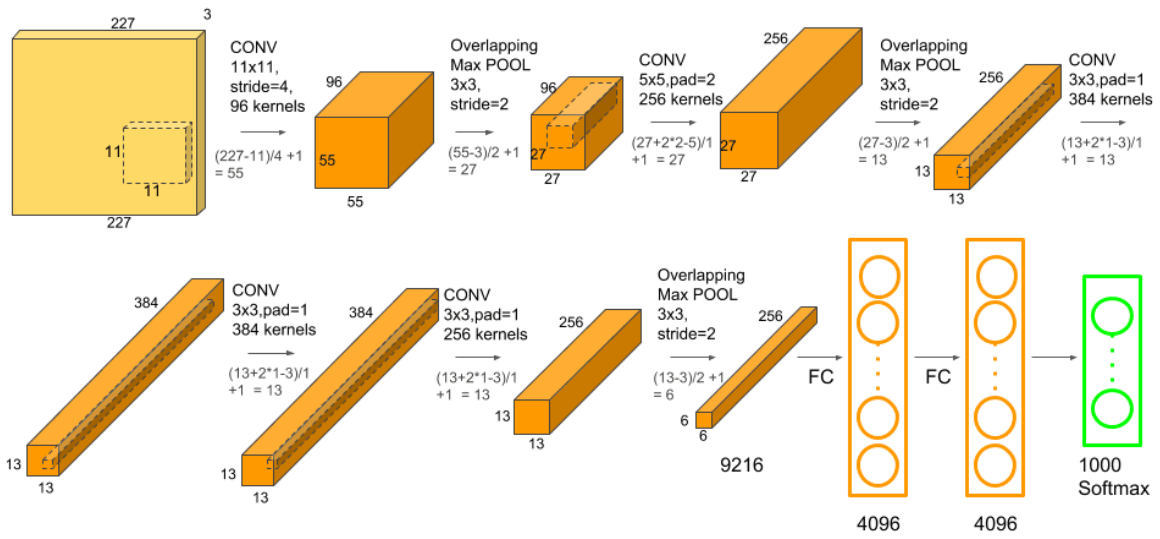


Figure 2. 7 AlexNet architecture (Pujara, 2020)

b) Inception

In 2014, Szegedy introduced the Inception concept model as an extension of the GoogleNet architecture (Szegedy et al., 2016). It has since become a widely used CNN architecture for tasks such as image recognition and classification. Different versions of the Inception model have been developed, denoted as Inception vN, where N represents the version number. The Inception model utilizes filters of different sizes to capture visual patterns of varying scales and achieve an optimal sparse structure using inception modules. These modules are composed of a pooling layer and three types of convolutional layers (1x1, 3x3, and 5x5) arranged in a stacked manner. This configuration allows the model to effectively capture features at multiple scales and learn a hierarchical representation of features in order to enhance the depth and width of the CNN while minimizing the impact on computational complexity the Inception model incorporates 1x1 convolutional filters as dimension reduction modules preceding the 3x3 and 5x5 convolutions as shown in Figure 2.8. This approach efficiently decreases the quantity of parameters within the network, resulting in a significantly lower parameter count compared to earlier architectures like AlexNet (Gu et al., 2018).

Over time, the Inception model has improved. Inception V3 introduced updates to the Inception module, leading to improved accuracy in ImageNet classification. As a result of improved model, spatial aggregation can be effectively achieved using lower dimensional embeddings without significant loss in representational power. Subsequently, Inception V4 model is introduced based on a combination of the residual connections with Inception architecture aiming to expedite the training process of Inception networks. In Inception V4, residual connections are used instead of the filter concatenation stage of the Inception architecture, which have shown effectiveness of the performance (Too et al., 2019) (Gu et al., 2018).

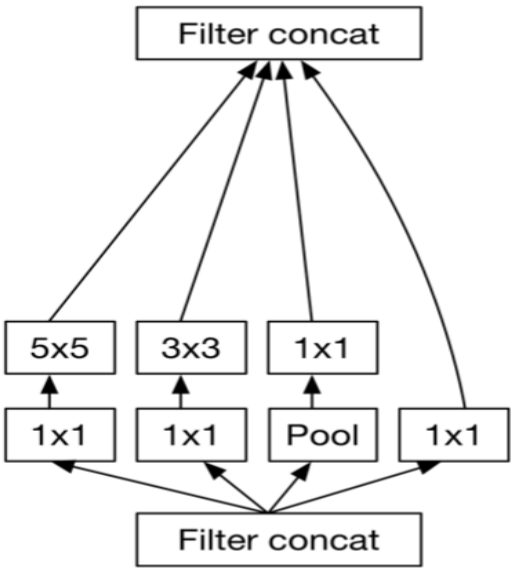


Figure 2. 8. Inception model block structure (Gu et al., 2018)

c) ResNet

In 2015, deep residual network (ResNet) was introduced based on CNN model for image recognition (He, Zhang, *et al.*, 2016). This model served as the foundation for the ILSVRC 2015 and COCO 2015 classification challenges. ResNet is a type of CNN model that employs a distinctive network architecture where the input from the former layer is concatenated into the output of the present layer as shown in Figure 2.9. CNN design organizes the architecture by sequentially combining fundamental units like convolution layers, activation functions, pooling, and batch normalization. ResNet is characterized as a network-in-network (NIN) architecture, relying on stacked residual units as its primary building blocks. These residual units are equipped with skip connections, allowing the network to learn more effectively and achieve superior performance. The residual units consist of convolutional and pooling layers. The pre-trained weights from ImageNet have been loaded into various ResNet models with 50, 101, and 152 layers. The ResNet architecture has

exhibited great success in various tasks, involving object detection, semantic segmentation, and image classification (Wu et al., 2018) (Too et al., 2019).

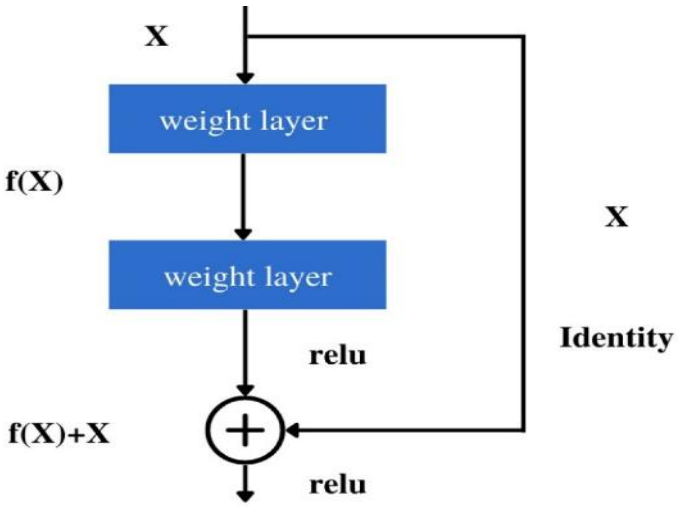


Figure 2. 9. Basic Structure of Residual learning block (He et al., 2016)

2.4.2. Detection Models

In general, most studies in the extant literature are dedicated to plant disease classification. However, plant disease identification (both localization and classification) is a complicated task. Some deep learning techniques have been developed for the purpose of plant disease detection. Deep learning meta-architectures such as Faster Region-based CNN (Faster R-CNN), Region-based Fully Convolutional Network (R-FCN), and Single Shot Multibox Detector (SSD) have been used as a detector for categorization and localization of plant leaves disease have been used in (Fuentes *et al.*, 2017) for detecting tomato diseases and pests, with suitable performance.

The authors of (Durmus et al., 2017) used AlexNet and SqueezeNet models to detect tomato diseases from leaf images and found that the former performed slightly better than the latter in terms of accuracy. Sladojevic et al. developed

an innovative technique based on deep CNN for detecting plant diseases automatically and classified 13 different kinds of plant diseases from the healthy leaf images using CaffeNet CNN architecture. The authors achieved an average accuracy of 96.3% (Sladojevic et al., 2016). Additionally, Hernández and López proposed a method for detecting plant diseases based on a probabilistic programming using Bayesian deep learning procedures (Hernández and López, 2020). Ferentinos used five CNN models, namely AlexNet, AlexNetOWTBn, GoogLeNet, Overfeat, and VGG to detect plant illnesses using images of healthy and unhealthy leaves. They reported VGG to be a successful model with a 99.53% success rate on test dataset containing 17,548 images (Ferentinos, 2018). In another study, Ramcharan et al. applied transfer learning for training a deep CNN Inception v3 to detect three cassava diseases and two kinds of pest damage (Ramcharan et al., 2017).

- **YOLOv5 Network Model**

Nowadays, the most popular approach for object detection tasks is YOLO. Jocher proposed a one-stage target recognition method known as YOLOv5 in 2020 (Jocher et al., 2021). YOLOv5 is improved based on YOLOv4 to reduce processing cost and increase detection performance (Li, Ahmed, et al., 2022). YOLOv5 categorized into five distinct network model variants: YOLOv5n, YOLOv5s, YOLOv5m, YOLOv5l, and YOLOv5x, distinguished by differences in network depth and width. The YOLOv5s network demonstrates the fastest computation speed, although with the lowest average accuracy, whereas the YOLOv5x network exhibits opposite traits (Chen et al., 2022). The YOLOv5 network comprises four main parts: input, backbone, neck, and head. The original structure of YOLOv5 is shown in Figure 2.10. The calculation of the adaptive anchor box module can adjust to various datasets and automatically present the initial anchor box's size (Yang et al., 2022). The

backbone is a pre-trained network used to extract, aggregate, and form different levels of image feature representation based on various image granularities when the image is entered, after which the image features are stitched and sent to the prediction layer, which includes a path aggregation structure (PAN) and feature pyramid network (FPN). The backbone consists of three parts: convolutional layers, C3, and spatial pyramid pooling fusion (SPPF). Among them, the convolutional layer is responsible for transforming the input image into feature maps of varying scales. The C3 module enhances the network's capability to understand the context and relationships between features at various scales (He and Wei, 2023). The structure of the C3 module, as shown in Figure 2.11, C3 module consists of three convolutional modules and a Bottleneck as shown in Figure 2.11. The Bottleneck is a residual block that outperforms ResNet's residual block in terms of computational speed.

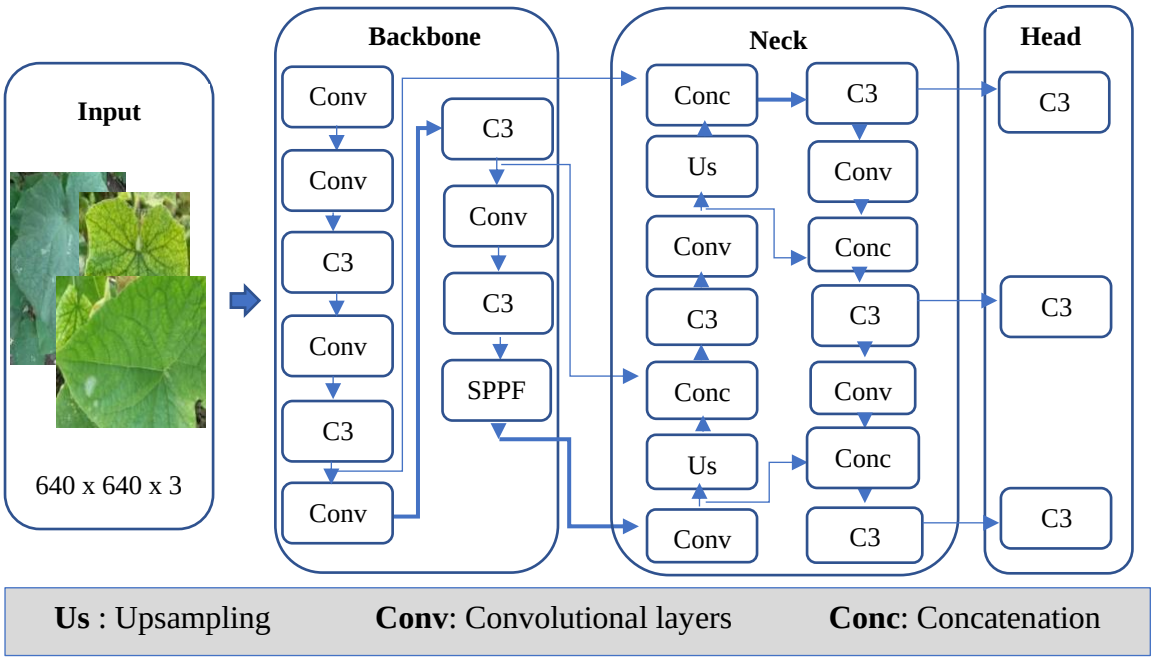


Figure 2. 10 The original structure of YOLOv5L

The SPPF layer enables the model to capture context and information at various resolutions (Zhou *et al.*, 2023). The neck section is an enhanced version of the FPN structure, designed to optimize the speed of feature fusion and information transmission throughout the network for improved performance. The neck section comprises of convolutional layer, concatenation, C3 and up sampling. Then, the head is the last detection part that predicts the image characteristics to provide bounding boxes and predicted target categories of various sizes (Chen et al., 2022) (Yang et al., 2022) (He and Wei, 2023).

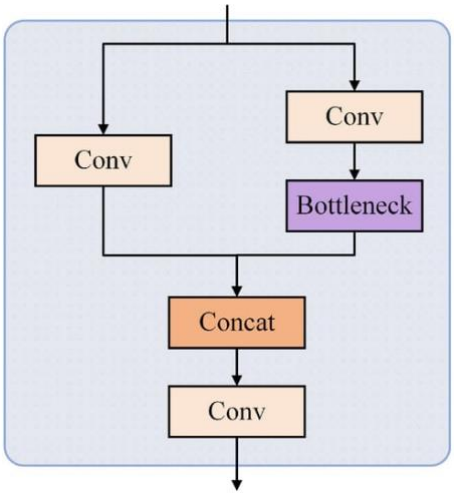


Figure 2. 11 The structure of C3 module (Zhou et al., 2023)

2.5. Related Works

Through an analysis of research conducted in the domain of deep learning models for recognizing and detecting cucumber leaf diseases and pests, we can categorize the relevant literature into two main sections: cucumber leaf disease recognition and cucumber leaf disease and pest detection.

2.5.1. Existing Plant Leaf Disease Classification Works

Many image processing concepts have been applied for plant disease recognition by researchers. Research has begun to consider the use of CNN to solve the plant leaf disease recognition problem as a result of their dedication and outstanding success in large-scale picture classification competition (Liu and Wang, 2020). Segmentation and feature extraction in traditional machine learning have a vital role to have an accurate classification system. Image segmentation technique has an important role in analysing images and identifying disease region. Following the trend of existing solutions, deep learning algorithms have used in segmentation process. CNN model based on the U-net architecture proposed for semantic segmentation of powdery mildew disease on cucumber leaves, their model outperformed as compared with K-means, random forest (RF), and GBDT segmentation techniques on 20 test image samples. The results show an accuracy of 72.11%, 83.45%, and 96.08% on intersection over union (IoU), dice and pixel respectively (Lin *et al.*, 2019).

In field of image processing, feature plays a vital role for classification process. Feature extraction is the essential step for getting the significant information. Color, shape, geometric features, and texture features that are extracted from image to determine diseases crops. Texture and color are the most important features that are considered in agricultural domain due to the range of differences in image samples (Khan, Akram, *et al.*, 2020). Convolutional and pooling layers have been used in deep learning techniques to extract features, whereas the process of extracting features and classification have been done automatically. CNN architecture is used to extract features automatically from plant leaves that is applied on different leaf datasets. The results utilized that CNN performed better, more efficient and more accurate in extracting feature compared with traditional machine learning methods (Agrawal *et al.*, 2021).

In the agriculture domain, a robust and accurate classification process is essential. Machine learning-based methods have been used for leaf diseases recognition in some studies. For instance, in (Krishnakumar and Narayanan, 2019) cucumber leaf disease classification and severity measures had been focused on to help farmers in terms of early diagnosis detection and discovering stages of affected leaf diseases, SVM used as a classifier. SVM used in (Zhang and Wang, 2016) for improving cucumber disease recognition, anthracnose, blight, and downy mildew leaf type diseases are used that each class includes 100 images. SVM was also used in (Zhang and Zhang, 2010) to cucumber leaf diseases recognition, it is trained using various kinds of kernel function such as Sigmoid, polynomial, and radial basis function (RBF) on both leaf spot disease and leaf as a sample. The experimental results indicate that SVM based on RBF achieved higher accuracy than others on 336 leaf spots samples. Three disease types which are downy mildew, brown spot and angular leaf spot were used.

In addition, an automated cucumber leaf diseases identification system proposed in (Kianat *et al.*, 2021) based on fusion and selected best features, six cucumber diseases are blight, powdery mildew, conrnespora, angular leaf spot, anthracnose, and downy mildew that includes 1,010 image used, the highest result achieved is 93.5% with quadratic SVM (QSVM) classifier. Decision tree (DT), logistics regression (LR), multi-class SVM (M-SVM), cubic SVM (C-SVM), Fine KNN, ensemble subspace discriminant analysis (ESDA), and neural network (NN) algorithms were used for detecting and identifying five cucumber leaf disease. The experimental results indicate that M-SVM obtained a higher result compared with other methods which is 98.08% (Khan, Akram, *et al.*, 2020). Additionally, direct feeding input and not scaling with data are the drawbacks of traditional machine learning algorithms, these lead to decrease the accuracy of the results.

Other methods such as, artificial neural network (ANN) used to classify cucumber crop disease in (Pawar *et al.*, 2016), it also give preventive measures

and remedies as a treatment, the result accuracy obtained is 80.45% on three classes like healthy, downy mildew, and powdery mildew. In (Zhang *et al.*, 2017), sparse representation (SR) method was used to recognize seven different kinds of cucumber leaf diseases. The recognition accuracy result obtained was 85.7% on 420 leaf images, which is higher than K-Means-based segmentation followed by neural-network-based classification (KMSNN), SVM, plant leaf image (PLI), and texture feature (TF) classifiers. Despite disease diagnosis, cucumber chilling injury had been detected using hyperspectral imaging system with feature selection methods such as mutual information feature selection (MIFS), max-relevance min-redundancy (MRMR), and sequential forward selection (SFS). SVM, naïve bayes (NB), and KNN as a classifier were used for identifying three-classes (normal, lightly chilling, and severely chilling) and two-classes (normal and chilling). The best accuracy result obtained using SFS with SVM are 100% and 90.5% for two and three classes respectively (Cen *et al.*, 2016).

In recent years, CNN have made significant progress in image recognition. For example, Mia et al. presented traditional ML and CNN-based algorithm for identifying cucumber diseases. As a result, RF provides the accuracy of 89.93%, and MobileNetV2 attains the highest accuracy rate of 93.23% compared with InceptionV3 and VGG16 (Mia et al., 2021). Omer et al. (Omer et al., 2023) presented a literature survey on the use of deep-learning algorithms for plant disease diagnosis. In (Agarwal *et al.*, 2021) CNN model proposed for identifying eight types of cucumber diseases. They used data augmentation and modification of Relu activation function to elevate accuracy result to 93.75%.

Further, the cucumber leaf diseases recognition system under the condition of small sample size and IoTs were proposed. The lesion of leave disease images acquired using one two-stage segmentation, by extracting the leaf disease spots such as color, texture, and border features. Data augmentation using activation reconstruction GAN (AR-GAN) applied to lesion leaf images.

Dilated and Inception CNN (DICNN) was used for classification. In their study, the obtained accuracy on raw diseased leaf and lesion images were 90.67% and 96.11% respectively (Zhang *et al.*, 2021).

In another attempt, symptom-wise of cucumber diseases recognition system proposed by (Ma *et al.*, 2018) on anthracnose, downy mildew, powdery mildew, and target leaf spots. In their study, disease symptom segmentation was used to segment symptom images by combining a comprehensive color feature with region growing. Augmentation methods such as flip horizontally, vertically, and rotate were used. The best accuracy result 93.4% was obtained using DCNN compared with RF and SVM classifiers on the unbalanced augmented data. A practical cucumber diseases diagnosis system proposed in (Fujita *et al.*, 2018), a new dataset has been made that includes seven viral diseases, healthy and downy mildew. They build CNN model from scratch and pre-trained VGG-net with fine-tuned and then they applied on 9000 images. The experimental results showed that VGG-net attained higher result accuracy than CNN, which are 93.6% and 86.6% respectively. Fujita et al. proposed a CNN system for diagnosing seven viral diseases and healthy cucumber leaf images. The obtained accuracy is 82.3% on 7520 cucumber leaf images, which is collected on good and bad conditions (Fujita *et al.*, 2016).

In addition, in terms of classifying multi-diseases on cucumber leaf, Tani et al. developed a CNN architecture based on a tunable threshold with sigmoid activation function on each output layer nodes instead of SoftMax function. An on-site cucumber leaf dataset constructed that included 11 single and 13 multi-diseases, the accuracy result obtained was 85.9% and 95.5% on multi-diseases and entire dataset respectively (Tani *et al.*, 2018). Furthermore, bases on wide-angle images, CNNdetect architecture designed to localize each leaf from the wide-angle then CNNdiag was improved from VGG-16 to classify diseased and healthy extracted leaf areas known as full leaf. Achieved accuracy result was 73.9% from leaf detection and 68.1% from detection and diagnosis on

13,601 images (Cap *et al.*, 2018). The literature review, as shown in Table 2.3, indicates that several algorithms had been carried out for cucumber leaf disease diagnosis. Some researchers worked on segmentation, extracting features and classification using traditional machine learning and deep learning algorithms.

Table 2. 3 Previous studies for plant leaf disease classification

Author	Pre-processing methods	Segmentation methods	Feature extraction methods	Classification methods	Diseases types	Accuracy result (%)
(Zhang *et al.*, 2017)	-	K-mean clustering	Color and shape	SR	Downy mildew, bacterial angular, corynespora cassiicola, scab, gray mould, anthracnose, powdery mildew. Total images : 420	85.7
(Zhang *et al.*, 2021)	AR-GAN	Combined GrabCut with SVM	Color, texture, and border features	DICNN	anthracnose, downy mildew, and powdery mildew	90.7 on raw leaf diseased, 96.1 on lesion images
(Fujita *et al.*, 2018)	-	-	CNN	CNN and VGG-net	MYSV, ZYMV, CCYV, CMV, PRSV, WMV, KGMMV, downy mildew, and healthy. Total images: 9000	93.6 for VGG-net 86.6 for CNN
(Kawasaki *et al.*, 2015)	Square crop, and square deformation	-	CNN	CNN	MYSV and ZYMV. Total images: 800.	94.9
(Krishnakumar and Narayanan, 2019)	Gaussian filtering used to blur the image to reduce the noise	K-mean clustering	HOG	SVM	Alternaria leaf blight, angular leaf spot, bacterial leaf spot, bacterial wilt, cercospora leaf spot, cucumber mosaic, target leaf spot, powdery mildew, downy mildew, phytophthora blight	86
(Zhang and Wang, 2016)	-	Watershed algorithm	Global-local SVD	SVM	Anthracnose, Blight, and Downy mildew. Each class includes 100 images	-
(Khan, Akram, *et al.*, 2020)	Improves the local contrast and makes infected regions more visible	SHSB	VGG-19 and VGG-M	DT, LR, M-SVM, C-SVM, Fine KNN, ESDA, and NN	Angular leaf spot, coryhespora, anthracnose, downy mildew, powdery mildew, and healthy	98.08 using M-SVM
(Pawar *et al.*, 2016)	Smoothing filtering used to eliminate noise	-	GLCM	ANN	healthy, downy mildew, and powdery mildew	80.45

(Kianat *et al.*, 2021)	Data augmentation and contrast enhancement performed	-	HOG, BRISK, and FAST	QSVM	blight, powdery mildew, conrnespora, angular leaf spot, anthracnose, and downy mildew. Total images: 1010	93.5
(Cen *et al.*, 2016)	-	-	MIFS, MRMR, SFS	SVM, NB, KNN	Cucumber chilling injury classes (normal, lightly chilling, and severely chilling). Also, two-classes (normal and chilling).	SFS 90.5, SVM 100
(Agarwal *et al.*, 2021)	Data augmentation	-	CNN	CNN	Angular spot, anthracnose, black spot, brown spot, downy mildew, gray mold, powdery mildew and target spot	93.75
(Ma *et al.*, 2018)	Flip horizontally, vertically, and rotate images	Combined color feature with region growing	-	DCNN, RF and SVM	anthracnose, downy mildew, powdery mildew, and target leaf spots. Total images: 14208	93.4 using DCNN
(Fujita *et al.*, 2016)	Augmentation with image shifting, rotation, and mirroring	-	CNN	CNN	MYSV, ZYMV, CCYV, CMV, PRSV, WMV, KGMMV, and healthy. Total images: 7520	82.3
(Tani *et al.*, 2018)	-	-	CNN	CNN	38821 leaves infected with any of 11 kinds of diseases. 1814 leaves infected with multiple diseases. 7676 healthy leaves	85.9 on multi-diseases 95.5 on entire dataset

2.5.2. Existing Plant Leaf Disease and Pest Detection Works

Image processing and pattern recognition techniques have been increasingly applied in the field of agriculture for plant disease recognition and detection. In particular, the use of computer vision systems has been instrumental in the development of systems that can accurately diagnose and detect various symptoms of plant diseases. In recent years, researchers have made significant progress based on deep learning to recognize and detect plant leaf disease and pests from images. This has helped farmers to detect and address the diseases at an early stage, leading to reduced crop losses and increased productivity.

Previous studies have demonstrated that deep learning techniques are successful for real-life object identification, recognition, and classification (Jiao et al., 2019). Currently, visible-light image recognition has been successfully used in the field of plant disease detection because of the

requirement for real-time monitoring and exchange of crop growth information (Chen et al., 2022). Plant disease and pest detection is a challenging task, to overcome this challenge deep learning techniques have been developed. In addition to that, deep learning networks have become the backbone of most state-of-the-art object detectors (Omer et al., 2023). A crucial task in computer vision is object detection, which involves identifying objects in an image or video and determining their location and extent (Jiao et al., 2019). Landmark detection algorithms, including YOLO, YOLOv3, YOLOv4, Faster R-CNN, SSD, and Mask R-CNN, have been effectively applied in crop disease and pest detection.

For example, the goal of (Barbedo, 2019) study was to provide a thorough literature review of the methods proposed to detect plant nutrient deficiencies based on proximal images. Faster region-based CNN (Faster R-CNN), region-based fully convolutional network (R-FCN), and SSD as DL meta-architectures used to detect diseases in different plant species (Hammad Saleem *et al.*, 2020). All these DL meta-architectures are also combined with "deep feature extractors" such as VGG net and ResNet in (Fuentes *et al.*, 2017) for tomato plant leaf diseases and pest detection. In another study, YueJu et al. developed and proposed a tiny YOLOv2 model for detecting immature mangoes (YueJu et al., 2018). YOLOv3 was improved by the authors of (Tian et al., 2019) to detect images of immature bloated apples, apples, and mature apples. The experimental results proved that the YOLOv3-dense algorithm can effectively detect apple fruit targets in various states. Furthermore, in (Liu and Wang, 2020) the YoloV3 algorithm was improved by using multi-scale feature detection to detect tomato insect pests and diseases based on object bounding box dimension clustering, image pyramid, and multi-scale training. From the experimental results, they attained a higher detection accuracy of 92.39% and improved detection speed compared to the original YoloV3, Faster R-CNN, and SSD algorithms. Li et al. proposed a cucumber leaf disease detection

algorithm based on an improved YOLOv4 model, in which CSPDarknet53 was replaced with MobileNetv3 in the backbone network (Li, Yue, *et al.*, 2022). They also constructed a small dataset of cucumber leaf diseases in real-world scenarios. They achieved superior recognition accuracy with 97.21% compared with Faster-RCNN and the original YOLOv4. In (Song et al., 2023) the DF-YOLO algorithm was proposed, which is based on the YOLOv4 network and used to identify pest species. With a self-made dataset of pests, the algorithm was evaluated. According to the data, the method's mAP is 94.89%.

Owing to the problem of slow detection speed in previous YOLO versions and high requests for detection conditions in other algorithms. Yang et al. used a lightweight Yolov5s network as a basic model and proposed a BCo-YOLOv5 model to enhance the ability of the model to improve the accuracy rates (Yang et al., 2022). As a result, they achieved better detection of the target fruits (citrus, apple, and grape). Chen et al. improved the YOLOv5 network for identifying plant leaf diseases accurately under complex natural conditions. The number of parameters and calculations was reduced on the model backbone using the InvolutionBottleneck module (Chen et al., 2022). The author of (Wang, Shang, et al., 2022) study proposed a method based on an optimized lightweight YOLOv5 network to improve the accuracy and speed of plant disease classification and detection. In their system, the model weight was reduced through the WBF structure and Ghostnet. The experimental results showed that the accuracy of the developed model was higher than those of the original model by 3.98%.

In addition, the author of (Wang, Cheng, et al., 2022) study constructed a CNN model YOLO-CBAM that incorporates the attention mechanism and YOLOv5 to detect Solanum rostratum dunal seeding. A method was developed for slicing high-resolution images by estimating the overlap rate to create datasets that reduce the possibility of detail loss owing to high-resolution photos being compressed during the training phase. The results proved that

YOLO-CBAM outperformed in terms of both precision and recall recognition rates of 0.9465 and 0.9017, respectively. In (Mathew and Mahesh, 2022) a YOLOv5 model was used to detect bacterial spot leaf disease in bell pepper plants. In (Lou et al., 2021) a one-stage detection model called YOLOv5 algorithm proposed for detecting cucumber leaf diseases. Their model obtained 84.6% with mAP accuracy result on the constructed cucumber leaf image dataset after labelling. Furthermore, an efficient detection model (EFDet) was proposed in (Liu et al., 2021) to detect a constructed cucumber leaf dataset that included bacterial angular spots, downy mildew, and healthy individuals. For comparison, the EfficientDet-D1, YOLO V3-ASFF, and YOLO V3-V5 models were applied. The results indicated that, when compared to the other models, the EFDet model performed better in terms of calculations, fewer model parameters, and model size. Based on the literature review, a deficiency in cucumber leaf disease detection using YOLOv5 is evident.

2.6. Research Challenges

Nowadays, deep learning models have attained good performance and shown encouraging outcomes in a variety of domains, including image classification and detection, speech recognition, and object detection. Different architectural models have been used in deep learning recently to obtain significant performance and efficiency. Despite the developments and improvements that have been applied to deep learning models in various research studies, especially in plant disease classification and detection, numerous significant research gaps and challenges still need to be addressed before implementing different deep learning architectures for plant disease recognition and detection. Addressing these challenges will contribute significantly to the sustainable management of plant health, agricultural productivity, and food security on a global scale.

2.6.1. Challenges Associated with Constructing Large Datasets

In deep learning, a huge dataset with a wide variety is required. Several challenges and issues came up during the process of seeking the dataset that may have affected the process of detection and recognition. For instance, the PlantVillage dataset was collected from different fields, and it may contain some class diversity. So, it leads to how data could be shown. Converting data into graphical representation is another problem because some sensitive information has been shown in graphical form, and it also will be clear to understand. Another issue is that the dataset has not shown the entire data, and some classes were missed. However, constructing such a dataset involves challenges. The author Barbedo realized that plant species, disease variety, variety of conditions in capturing image, and sample numbers in each class of the dataset affect and prevent deep learning models more widely to be used in practice (Barbedo, 2018). Data annotation is a critical task that necessitates expert involvement for precise labeling of input images. (Kamilaris and Prenafeta-Boldú, 2018).

In previous studies, researchers have used different datasets to train deep learning models for plant disease classification and detection tasks. For example, PlantVillage dataset is mostly used to calculate the accuracy and performance, which contains healthy and diseased images of five crops with simple plain background, namely apple (Liu et al., 2017), corn, grape, potato, and tomato (FatihahSahidan et al., 2019)(Ramcharan et al., 2017). Yet, most researchers have used similar architectural design and obtained a quite redundant result from their experiments on the dataset such as (Amara et al., 2017)(Dara and Tumma, 2018) (Victoria and Maragatham, 2021) (Brahimi et al., 2017)(Cruz et al., 2017). Though they have used several aspects of the model for training and testing the system for plant disease recognition, they have not still gained enough new information. Additionally, the authors of

(Amara et al., 2017) used a real banana disease dataset, which they derived from the PlantVillage dataset. The authors of (Ramcharan et al., 2017) used cassava disease dataset. Some researchers have also constructed a synthetic dataset (Sladojevic et al., 2016). According to the research findings in this domain, one of the primary and significant research gap is the lack of comprehensive datasets designed specifically for training and evaluating plant cucumber leaf disease and pest diagnosis. Although datasets exist for other crops, there is a limited availability of extensive and diverse datasets which specifically focuses on cucumber leaf diseases and pests. Constructing such datasets is crucial for training recognition models that are accurate, reliable, and capable of effectively handling the distinct characteristics and variations associated with cucumber leaf diseases and pests.

2.6.2. Challenges of Plant Diseases

Plant disease management and pathology are faced with ever-growing challenges. Plant diseases present a range of intricate challenges that impact agriculture and ecosystems. Navigating these challenges requires a comprehensive understanding of the various factors at play, as well as the development of innovative strategies to mitigate their effects. Accurate identification and classification are difficult due to varying symptoms across species and conditions.

On the one hand, agricultural productivity has reduced due to depleting natural resources and diminishing arable lands. On the other hand, due to increasing global population, requests for high quality and varied food have increased. Additionally, the evolution and epidemics of plant diseases have globally increased because of intensification, resources such as water, fertilizer, pesticides, globalization, and climate change (He, Zhan, *et al.*, 2016). Plant diseases and pests are the major reason that lead to substantial economic losses

and reduced plant yields. Pathogen resistance to conventional methods necessitates innovative controls.

In technological advancements, the theories of plant diseases and pest diagnoses such as detection and classification have been developed from symptoms and signs of the diseases (Balodi et al., 2017). Complex interactions between pathogens and hosts complicate management. Reliable disease data access and sharing remain critical. Solving these challenges demands interdisciplinary efforts to safeguard agriculture, and ecosystems worldwide. In the field of plant pathology advancements, some new avenues for specific and sensitive plant diagnosis procedures have been developed that are coupled with molecular biology, bioinformatics, and biotechnology (He, Zhan, *et al.*, 2016)(Balodi et al., 2017). In the future, plant disease management plans, such as an accurate plant disease recognition, are important and must be emphasized more for societal development, food security globalization, climate change, and disease prevention.

2.7. Problem Formulation on the Existing Diagnosis Systems

The literature review presents a comprehensive studies in the field of plant disease recognition as well as detection systems. This broad literature review serves as a roadmap to uncover the limitations of current approaches and leads to the improvement of a well-defined research problem. After such broad review, it was found that in past decades, deep learning-based CNN algorithms have been highlighted as one of the significant methods being researched in agriculture domains. Despite the fact that various deep learning algorithms were applied and developed to the application of cucumber leaf disease diagnosis and detection, this is still a fertile area of research and should result in improvements for better diagnosing of cucumber leaf diseases.

From this literature review, it is found that there is a lack of a reliable publicly cucumber leaf disease datasets for researchers. Besides, they did not consider an important type of pest named Spider that affect the whole cucumber surface leaf. Therefore, it is necessary to improve and construct a new dataset that includes various plant organs and different leaf diseases. As a result, constructing a reliable cucumber leaf disease types and healthy dataset becomes an essential task.

Furthermore, this literature review reveals several issues and challenges regarding the CNN algorithm's use in agriculture sectors, specifically, in classifying cucumber leaf disease. The issues revolve around model hyperparameters and overfitting. It can be concluded from this review, that the existing systems have not provided pre-trained models such as AlexNet, Resnet-50 and Inception-v3 for cucumber leaf diseases diagnosis. Further, they did not use two combined CNN models in parallel. A compelling demand arises for an effective CNN algorithm to address the intricate trade-off between efficiency and performance in terms of classifying cucumber leaf diseases. Therefore, the development of new CNN algorithm is crucial to further identify and classify cucumber leaf disease. It would has an ability to reduce the time consumed by farmers in recognizing leaf diseases. To tackle potential challenges, the algorithm incorporates techniques for model hyperparameter tuning and data augmentation. The review also emphasizes the problem of low accuracy results, indicating the necessity for continuous improvement and refinement in the algorithm's performance. By considering and addressing these factors, the new CNN algorithm shows promising potential in improving the diagnosis of cucumber leaf disease and supporting farmers in achieving efficient crop management.

Another finding based on the literature review, it is observed that even though the application and advancement of the YOLOv5 model based on deep learning in plant leaf disease detection, there are still several existing issues and

limitations despite its widespread use and development. For example, one such limitation is the scarcity research on cucumber leaf disease and pest detection via YOLOv5. Additionally, significant concerns have been noted regarding the YOLOv5 model, including time consumption, storage complexity, low detection accuracy, and small symptom disease detection. It has also a struggles with implementing multi-disease and pest detection on a single leaf surface because of various disease symptoms and varying appearances. Therefore, a lightweight improved YOLOv5 model has become essential to improve detection accuracy, reduce model parameters and weight sizes. In addition, addressing these improvements will contribute to reduce time consumption and storage complexity.

2.8. Summary

This chapter provides a review of the improvement in plant leaf disease and healthy recognition and detection systems; it also highlights challenges and issues concerning CNN based deep learning algorithms in the agriculture domain. In summary, there is still a need for an accurate and efficient model to diagnosis cucumber leaf disease and pests. The literature review further underscores the demand for developing a new cucumber leaf disease and healthy dataset, proposing and improving recognition and detection models to address the challenges faced by farmers in the agriculture farming environment.

CHAPTER THREE

3. RESEARCH DESIGN, MATERIALS AND METHODS

3.1. Overview

This chapter introduces the research framework employed in this study, discussing material and research methods for the cucumber leaf disease and pest diagnosis system. The existing models require enhancements to more accurately identify diseases, necessitating retraining. This enhancement is crucial for improving both detection and recognition accuracy. The roadmap of this chapter has been organized as the following: it begins by highlighting a problem that needs to be addressed, followed by an overview of the research methodology. Additionally, the chapter investigates into background information concerning the dataset constructed for cucumber leaf disease, including data acquisition, data pre-processing techniques such as data augmentation and labelling. Experiment setup, the metrics for evaluating the performance of the models together with the methodology for results analysis and validation are also described.

3.2. Research Framework Overview

The study framework implementation process is divided into three different phases: phase 1, constructing a new cucumber leaf disease and pest dataset; phase 2, design a recognition system for cucumber leaf diseases and pests, and phase 3, improve a detection system for cucumber leaf diseases and pests symptoms. The output of the first phase is a fundamental input for the second and third phases. Each phase is structured into different steps, with each step generating a crucial output that serves as an essential input for the subsequent steps. Overall dissertation framework for the design and improvement of the proposed model for cucumber leaf disease diagnosing and detection is presented in Figure 3.1.

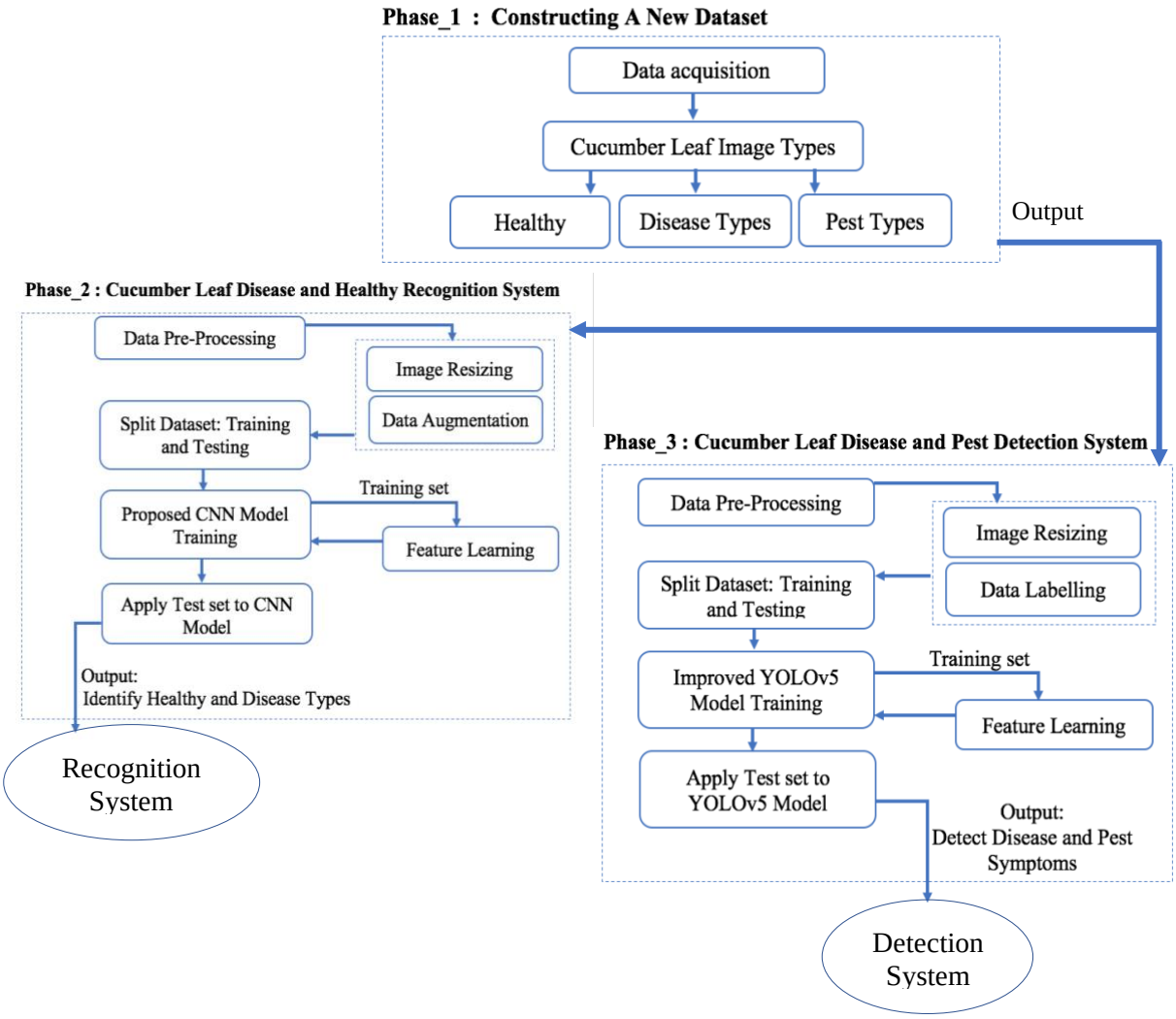


Figure 3. 1 Overall dissertation framework

3.2.1. Phase 1: Constructing A New Dataset

Phase 1 of the research framework focused on constructing and collecting a structured dataset including cucumber leaf disease, pest and healthy images. Phase 1, as described in Section 3.3, generated image leaf samples as output that are essential for diagnosing cucumber leaf health and disease types in phase 2. Simultaneously, the output of phase 1 serves as input to phase 3, where the detection and localization of leaf disease and pest symptoms take place.

3.2.2. Phase 2: Cucumber Leaf Disease and Healthy Recognition

Phase 2 discussed the design and improvement of diagnosing cucumber leaf disease and health. This phase includes the development of a CNN model utilizing deep learning techniques. This phase is divided into three different steps as described below and shown in Figure 3.2. Step 1 is named data preparation, which focuses on preparing leaf sample images including five infected and one healthy. In the second step of the pre-processing. Firstly, a white background was manually added to some healthy images, highlighting the leaf disease by attaching a white paper to the cucumber leaf background. Secondly, images are resized into 227x227x3 to be fitted into the model. Thirdly, five different augmentation methods have been applied on each training image samples. Finally, step 3 focuses on determining the approach to design and enhance the feature extraction and classification methods. Phase 2 yielded the development of an intelligent, fine-tuned CNN algorithm for the diagnosis of cucumber leaf health and disease types.

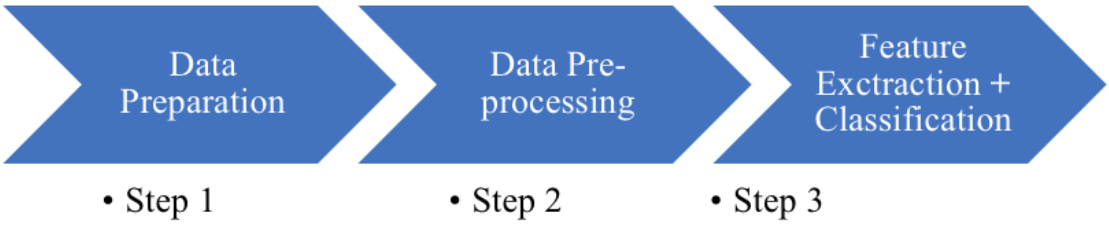


Figure 3. 2. Research flow framework of the recognition task

3.2.3. Phase 3: Cucumber Leaf Disease and Pest Detection

Phase 3 of the research framework focuses on the further enhancing ability of YOLOv5 model to detect cucumber leaf disease and pest symptoms. This

was achieved by detecting, localizing, and identifying disease and pest types in cucumber leaves. Phase 3 is divided into various steps. Starting with data preparation step, second step focuses on data pre-processing including image resizing, labelling and bounding box of each disease and pest symptom regions. Finally, in the third step, YOLOv5 model was improved and parameters were set to design a lightweight network model including feature extraction, classification and detection process. Phase 3 led to the improvement of the YOLOv5 detection model. The research detection framework task is shown in Figure 3.3.

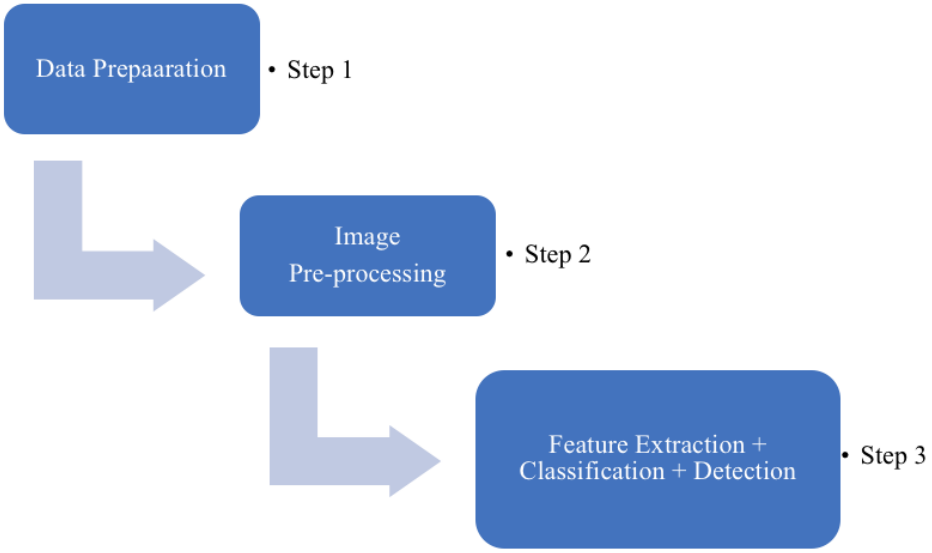


Figure 3. 3 Research flow framework of the detection task

3.3. Dataset Description (Data Acquisition)

In previous studies, various datasets have been used for plant disease classification and detection tasks. Most of the experimental studies are experimented on the PlantVillage dataset which include 38 classes of healthy and diseases of 14 different crop spices that contain 54323 images in total. This

section underscores the continued necessity for the creation of a new and diverse leaf image dataset for cucumber plant disease and health diagnosis. Existing datasets, in addition to not mentioning the cucumber Spider leaf pest, may not adequately represent various real-world conditions, such as different lighting situations and real backgrounds. It is also concluded that leaf is the most commonly used plant organ for classifying plant diseases as its image can be easily collected, and it is green and smooth during all four seasons. However, the actual environmental needs to be taken into account for a realistic scenario.

The datasets used in this dissertation include the descriptions of the leaves before and after the diseases affect them. The data comprises tables and images of the leaves that are taken in the fields. The data are analysed and classified in a way that is easy for readers to understand. There are a few publicly available datasets that are used by researchers in terms of the plant disease diagnosis system. In the cucumber disease recognition area, there is a lack of large public datasets that causes a major drop in the performance of classification and detection task. An accurate disease diagnosis system is in need for good training that depends on data collection. To address those issues, in this study, two different cucumber leaf diseases and pest datasets (local and publicly available) datasets are used for experimental setup.

3.3.1. Dataset-1 (Self-Made Dataset)

a) Introduction

In this study, a new structured dataset was constructed that includes healthy and infected cucumber leaves with single and multi-infections. The data are collected from natural scenes in Kurdistan region, Sulaymaniyah, Rania. It contains five cucumber leaf diseases and pest classes that comprise two pest diseases (spider, and leaf miner), two diseases (downy mildew and powdery

mildew), one viral disease cucumber yellow stunting disorder virus (CYSDV), and one healthy leaves. Total images of the dataset are 4868 images, each class having a sample image number in the range of 350–1500. The name and the number of images in each class are shown in Table 3.1. Sample images of Dataset-1 are also shown in Figure 3.4.

Table 3. 1 Sample dataset image number

Class No.	Disease and pest class	Sample image number
1	Spider	610
2	Leaf Miner	886
3	Downy mildew	349
4	Powdery mildew	1493
5	CYSDV	693
6	Healthy	837
	Total	4868

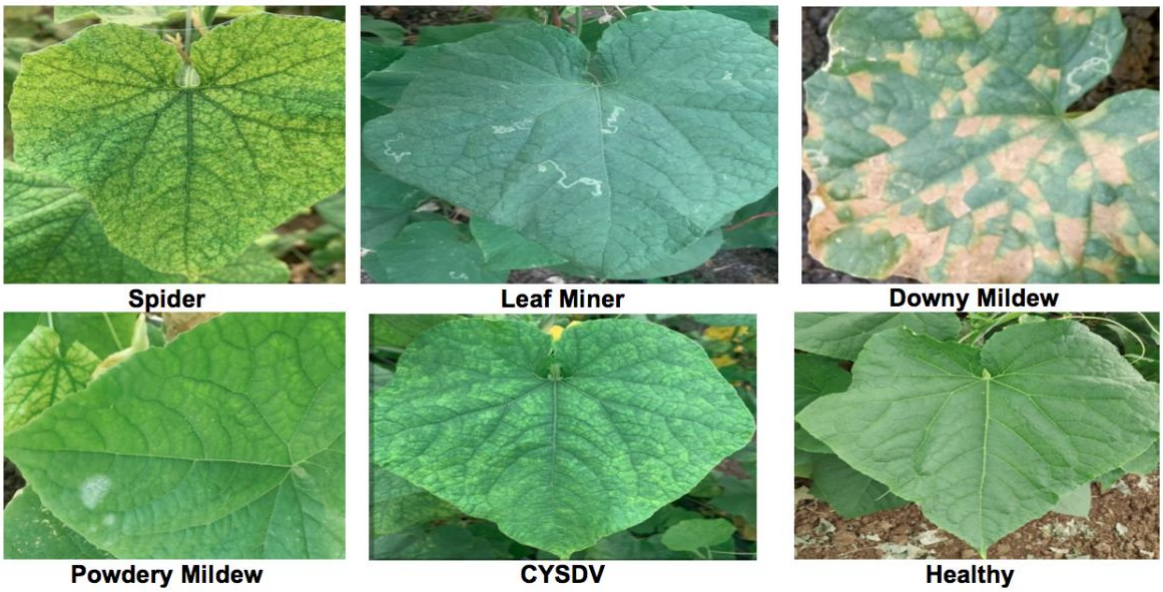


Figure 3. 4 Images of the five cucumber diseases and healthy leaves of constructed Dataset

Furthermore, it is important to acknowledge that the existing dataset includes supplementary information regarding leaf pests such as spider that affect cucumber leaves. This type of pests is commonly observed in agricultural environments and have a significant impact on the overall health and vitality of cucumber leaves. Spider mites, being small arachnids, infest the underside of leaves and puncture plant cells to extract their contents result in noticeable signs like stippling, yellowing, and eventual leaf death. By incorporating data related to these specific leaf pest, the dataset provides a more comprehensive understanding of the diverse factors influencing the health and vulnerability of cucumber plants to diseases. Figure 3.5 shows the spider mites that will have an impact on the cucumber leaves.

b) The Process of Collecting Leaf Image Samples

The dataset created in this study includes leaf image samples which were taken under specific field conditions at the green house of cucumber farms. A total of six different farms were used to collect and create the dataset in order to encompass a broad spectrum of environmental and cultivation circumstances, thus ensuring the inclusiveness and diversity of the dataset.

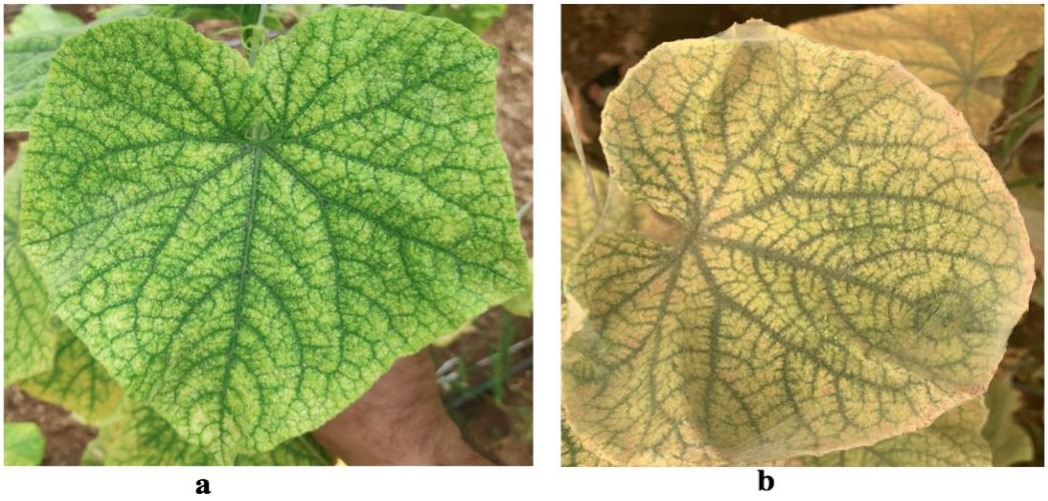


Figure 3. 5 Spider mites leaf affects (a) At early stage (b) At Nearly final stage

The name and the number of images in each class are shown in Table 3.1. Images have been taken from different weather conditions inside the green house (morning, middays, and evening), and angles (top and level angles). The images exhibit a various aspect ratios, orientations, and sizes (3024×4032, 1968×4160, 1801×1762, 1280×1280, 960×1280, and 606×1280) with pixel spatial resolution and intensities throughout different times of the day. Some images have a misshapen and shadows due to an order of magnitude, illumination, and distance changes.

In certain healthy class images, a manual effort was made to create a white background, with the intent of highlighting leaf diseases and pests. This was achieved by attaching a white paper to the cucumber leaf background, with the leaf positioned at the center of the image, as shown in Figure 3.6. All other healthy and unhealthy leaf images have a complex and an inconsistent background. They contain more than one leaf, stem, cucumber fruits, etc.. Finally, images are resized to $227 \times 227 \times 3$ for the purpose of reducing computational time, and improving efficiency processing.

c) Device Equipment and Time Period

All sample images have been captured using the smartphone devices (iPhone XS-Max, full HD, 12MP) and (Xiaomi Pocophone F1, full HD,12MP), optical and digital zoom are not used, and flash is always off. Over the course of a period spanning 10 months, data collection was conducted within the time frame of 7:00 AM to 6:00 PM. Specifically, the data collection period commenced on (March 15, 2021), and extended until (November 30, 2021), and then it resumed again up to (June 17, 2022). Throughout this extensive duration which takes 2 years, all samples were systematically collected and captured, adhering to the specified time constraints.

d) Dataset Disease List Names

The dataset construction involved labeling disease types based on guidance from pathologists, agricultural experts, farmers' experiences, and online references. Careful attention was given to the symptoms appearing on both the front and back sides of cucumber leaves during the collection of disease and pest image samples. This comprehensive approach improved the reliability and accuracy of the dataset. In the sample collection process, some disease and pest types were initially identified and labeled based on the front side of cucumber leaves. However, for other disease types, confirmation was necessary by considering both sides of the leaves. Figure 3.6 illustrates the process of considering both cucumber leaf sides in CYSDV.

(a) Frond Side **(b) Back Side**

Figure 3. 6 The process of considering both leaf sides of CYSDV

In the process of dataset collection, valuable insights and contributions were obtained from a team of experts with diverse backgrounds. This team included three agriculture academic experts affiliated with the agriculture department of the University of Raparin and Salahadin University, two pathologists

responsible for overseeing farmer farms. In addition, two professionals specializing in agriculture medicine, and four experienced farmer experts have collaborated. Their collective expertise and their contributions played a significant role in ensuring the quality and comprehensiveness of the dataset. In Table 3.2, a detailed description of the dataset is provided along with relevant information. The table offers a comprehensive information and characteristics that contribute to the constructed dataset. Various aspects of the dataset image samples were highlighted. The first column provides a class name, second column includes a description of disease causes that affect the cucumber leaves. Also, third column provides cucumber leaf disease types.

Table 3. 2. Detailed description of the dataset with relative information

Class No.	Disease Name	Disease Causes	Disease types
1	Healthy	-	-
2	Spider	• Spider mites • Excessive nitrogen fertilization • Environmental factors such as extreme temperatures, high humidity	Pest
3	CYSDV	• Transmission through the feeding activity of infected silverleaf whiteflies (Bemisia tabaci) • Environmental factors such as high temperatures, low humidity, and dry conditions	Viral disease
4	Leaf Miner	• The larvae of some moths, flies, sawflies, or beetles	Pest
5	Powdery Mildew	• Fungus Podosphaera xanthii • Environmental factors such warm and humid environments	Fungal
6	Downy Mildew	• Oomycete Pseudoperonospora cubensis	Fungal

3.3.2. Dataset-2 (Cucumber Plant Disease Dataset)

In the conducted experiments, the dataset used for analysis and evaluation was a publicly available dataset obtained from the Kaggle website. It is a platform renowned for hosting various datasets and machine learning competitions. This particular dataset, known as the "Cucumber Plant Disease Dataset" specifically focuses on cucumber plants and the various diseases they may encounter (Negm, 2020). The dataset comprises images and related information pertaining to the diseases that affect cucumber plants. Researchers, data scientists, and enthusiasts can access and download this dataset for analysis, study, and development of machine learning models aimed at detecting, diagnosing, and mitigating cucumber leaf diseases. Cucumber plant disease dataset contains two different cucumber classes: healthy and diseased, as shown in Figure 3.7.

Good Cucumber **ill Cucumber**

Figure 3. 7 Images of leaves of cucumber plant disease dataset

Table 3.3 illustrates the dataset class name and image numbers. Total images of the dataset were 695 images, each class having a small number of samples, which are 343 and 352 images for healthy and diseased classes respectively.

Table 3. 3. Sample dataset-2 image numbers

Class No.	Class Name	Sample image number
1	Good Cucumber	343
2	Ill cucumber	352
	Total	695

3.4. Dataset Enhancement Methods

It is common knowledge that machine learning algorithms become more powerful as they gain access to larger amounts of data. Despite the lower quality of the data, the algorithms can still exhibit improved performance by extracting valuable information from the original dataset (Mikołajczyk and Grochowski, 2018). The efficacy of deep learning models is significantly dependent on the availability of large volumes of training data to avoid overfitting. Overfitting is a common problem that arises when the model is trained using a limited dataset, resulting in subpar performance when applied to new validation and testing data due to a lack of generalization (Chlap et al., 2021). Another challenge associated with small datasets is that models trained on them tend to struggle in effectively generalized well data from the validation and testing sets (Perez and Wang, 2017). Various approaches have been proposed to address this problem, with one of the methods being the use of data augmentation. This technique is essential and powerful in expanding the size, improving the quality, and diversifying the training dataset. It can be viewed as

a form of regularization technique that helps decrease the model's generalization error.

Data augmentation procedure takes images from the dataset and introduces alterations to create additional representative variation samples in the dataset. By augmenting the data, the model becomes less likely to learn overly specific features tied to the original training data, resulting in enhanced generalization and improved performance on the test set. Another issue that often arises during model training is class imbalance, wherein certain classes have insufficient representation in the dataset. The imbalance issue can cause the model to exhibit a bias towards the over-represented classes (Chlap et al., 2021). Augmentation methods involve applying transformations to an image, such as relocating its points or manipulating its intensity values. This process generates an augmented image. By performing this operation on a single image from the original dataset, the dataset's overall size is expanded. Although these techniques can significantly enhance the performance of the trained model (Chlap *et al.*, 2021).

Numerous research studies evaluating the effectiveness of data augmentation rely on widely recognized academic image datasets to establish benchmarks and evaluate outcomes. These datasets include various collections such as MNIST (handwritten digit recognition), CIFAR-10/100, ImageNet, tiny-imagenet-200, SVHN (street view house numbers), Caltech-101/256, MIT places, MIT-Adobe 5K dataset, Pascal VOC, and Stanford Cars. The availability of open-source datasets has provided researchers with a diverse range of scenarios to compare the performance outcomes of data augmentation techniques (Shorten and Khoshgoftaar, 2019). This PhD dissertation applies five different image data augmentation techniques to dataset-1 and dataset-2. The increase in image samples is determined by the maximum number of samples for each class. In this circumstance, the sample number of powdery mildew has been chosen that contains 1493 images. The overall sample image

number of a new augmented dataset-1 increased to 9927 images. The data augmentation techniques utilized in the study are outlined as follows:

Rotation: The image rotation augmentations method involves rotating the image either to the right or left along an axis within the range of 1° to 359°. The purpose of image rotation is to adjust the image's orientation by a specific angle θ. The safety of rotation augmentations relies significantly on the chosen rotation degree parameter (Shorten and Khoshgoftaar, 2019)(Umer et al., 2022). In this study, the image rotation is carried out at an angle θ = 40°, as shown in Figure 3.8.

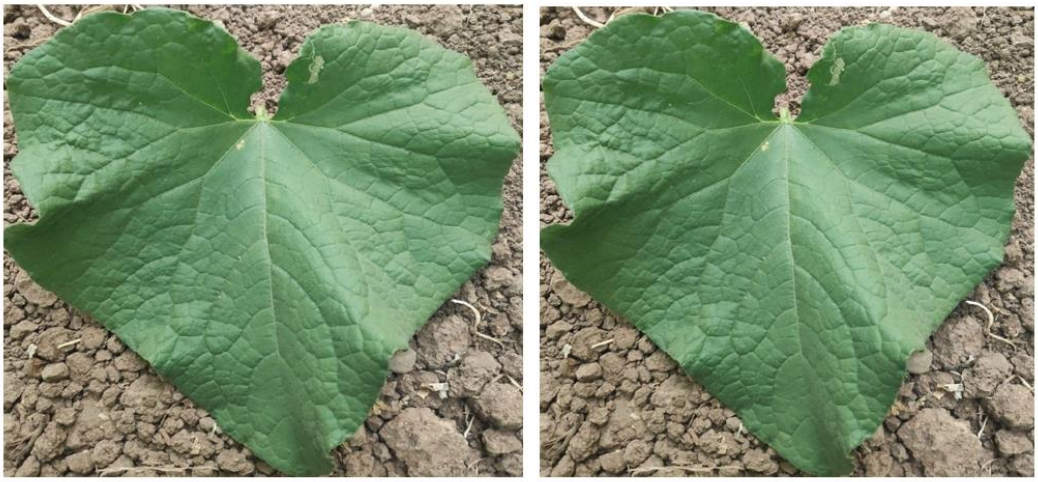
(a) Original Image **(b) Rotated Image**

Figure 3. 8 Illustrates the effects of rotate data augmentation technique on one image of the dataset

Shear: The shearing mapping technique is a linear transformation that shifts the position of each pixel in the image in a fixed direction relative to a line parallel to that direction, passing through the origin (Umer et al., 2022). By applying a shear transformation on the image, the transformation occurs along the horizontal or vertical axis. The magnitude rate for this transformation

ranges from -0.3 to 0.3. In this study, the image shearing is carried out with rate magnitude 0.20, as shown in Figure 3.9.

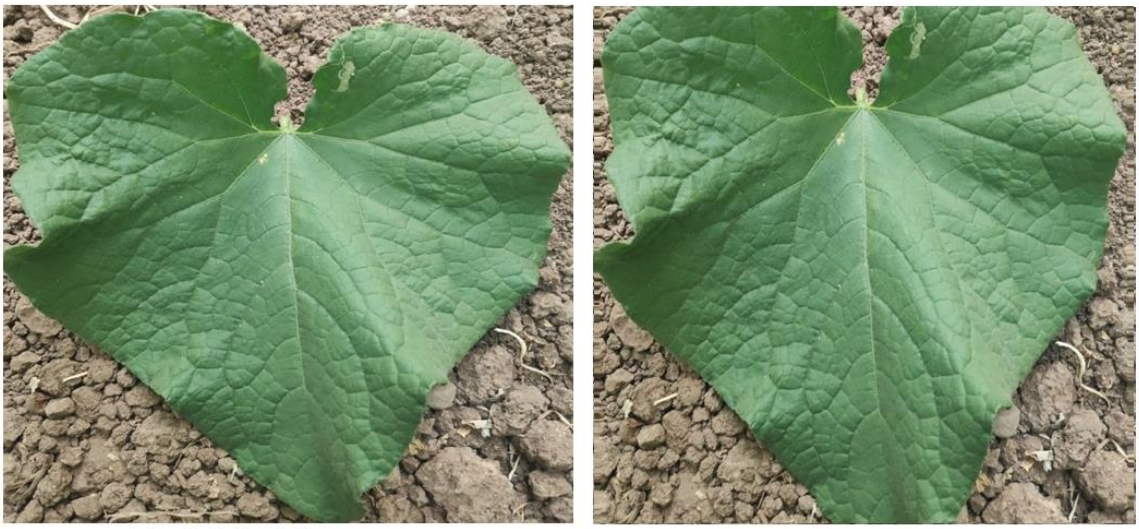
Zoom: Image zooming refers to the process of enlarging a specific region at the center of the image. This process is employed to extract the region based on the edge pixels, utilizing the nearest neighbor interpolation method with pixel replication (Umer et al., 2022). In this study, the image zooming is carried out with degree of 0.20, as shown in Figure 3.10.



(a) Original Image **(b) Sheared Image**

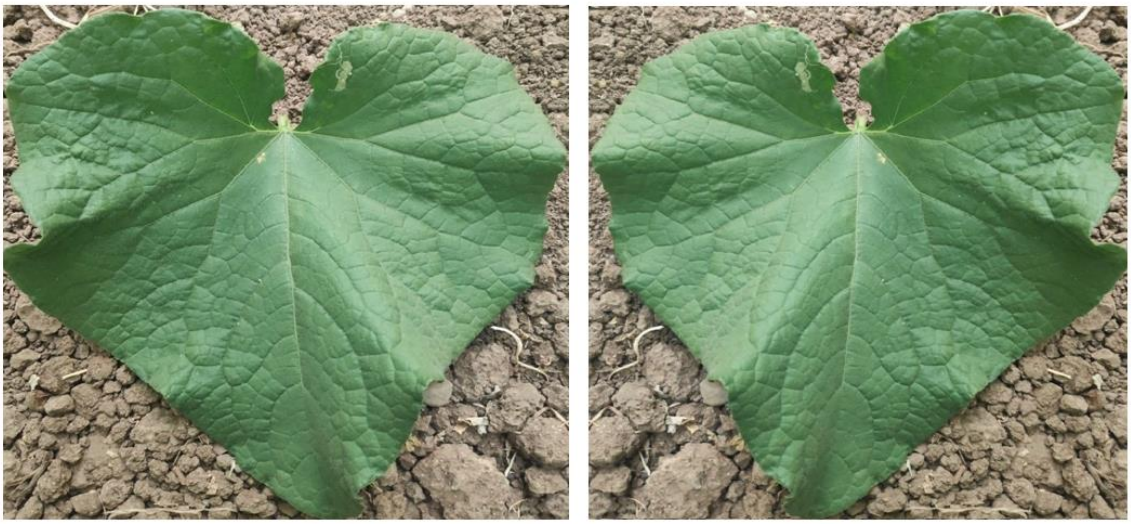
Figure 3. 9 Illustrates the effects of shear data augmentation technique on one image of the dataset

Horizontal flip: Horizontal axis flipping is a common technique in image processing, involving a special case of image rotation where the image is rotated 180 degrees. This is achieved by applying the rotation affine transformation to the image (Umer *et al.*, 2022). Flipping along the horizontal axis is more prevalent than flipping along the vertical axis. It is also one of the simplest methods to implement (Shorten and Khoshgoftaar, 2019). The effect of horizontal flip data augmentation technique is shown in Figure 3.11.



(a) Original Image **(b) Zoomed Image**

Figure 3. 10 Illustrates the effects of zoom data augmentation technique on one image of the dataset



(a) Original Image **(b) Horizontal Flipped Image**

Figure 3. 11 Illustrates the effects of horizontal flip data augmentation technique on one image of the dataset

Brightness: Brightness augmentation is an approach employed to alter the brightness levels of an image by manipulating its pixel values. This adjustment can make the image appear brighter or darker, effectively modifying its overall

intensity and introducing changes in lighting conditions. The effect of brightness data augmentation technique has been shown in Figure 3.12.

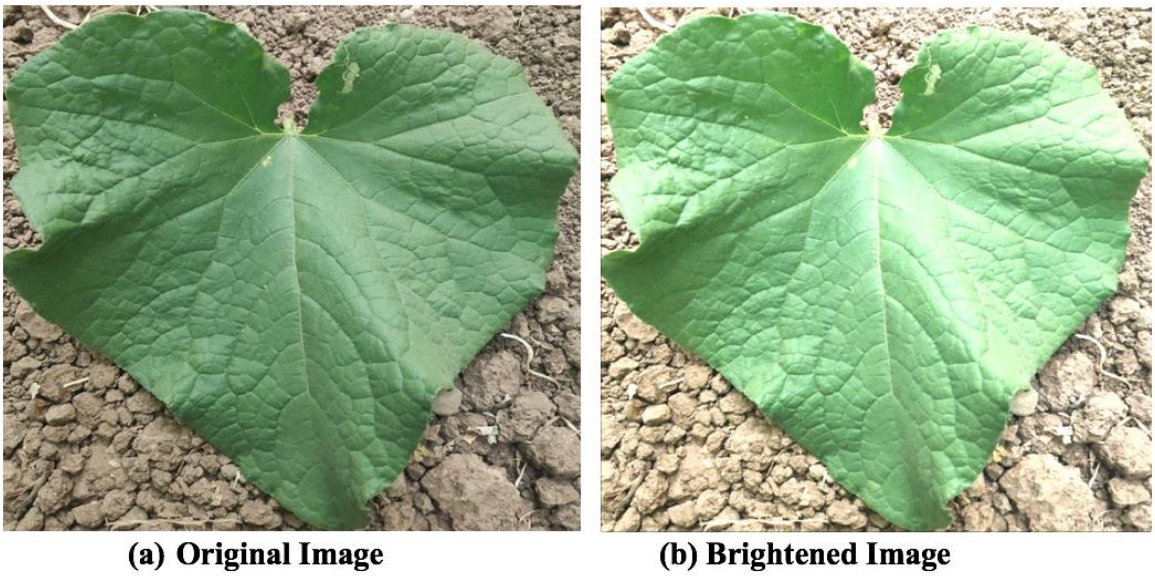


Figure 3. 12 Illustrates the effects of brightness data augmentation technique on one image of the dataset

Furthermore, five different augmentation techniques have also been applied on each training set images on dataset-2. The training images are increased by 6 times, the overall training image samples of a new augmented dataset-2 increased to 3308 images.

3.5. Image Data Annotations

Data labelling involves the process of assigning meaningful and precise annotations or labels to the training data utilized for training a model. The labeling process requires attributing specific class or attributing information to individual data samples, enabling the model to learn and make predictions based on these labeled instances. In deep learning, data labeling is essential for

the model to comprehend the patterns and relationships within in the data. The labeled data serves as the ground truth, providing the model with the necessary information needs for learning and make predictions on new, unlabeled data.

Data labeling process usually requires the involvement of human annotators or experts in the relevant field who carefully examine each data sample and assign appropriate labels according to the intended classification or detection tasks. Data labeling could be time consuming, especially when handling extensive datasets. It necessitates expertise and careful attention to ensure precise and consistent labeling.

However, cucumber leaf images in the constructed dataset that were collected in the real scene were not marked and annotated. Symptoms of cucumber leaf diseases and pests were required to be assigned and labeled accurately in order for the model to work better. Various tools and platforms are available to facilitate the data labeling process, allowing annotators to efficiently label data and collaborate on labeling tasks.

In this study, a labelling tool called ImageLabeler software was applied to annotate and mark the cucumber leaf diseases and pest symptoms. The ImageLabeler software facilitated the process of accurately identifying, localizing and highlighting the specific regions of the diseases and pest symptoms on the cucumber leaves. LabelImg is a software tool designed for the process of image annotation and labelling, specifically for object detection and computer vision tasks. The tool is free, open-source, and written in Python, utilizing quality threshold (QT) for its graphical user interface. We utilized the user-friendly LabelImg tool to load images of cucumber leaf diseases and pests. Within this tool, we marked specific symptoms or regions within the images by drawing bounding boxes around them. These bounding boxes functioned as annotations or labels, indicating the presence and location of the desired objects within the images, as shown in Figure 3.13.

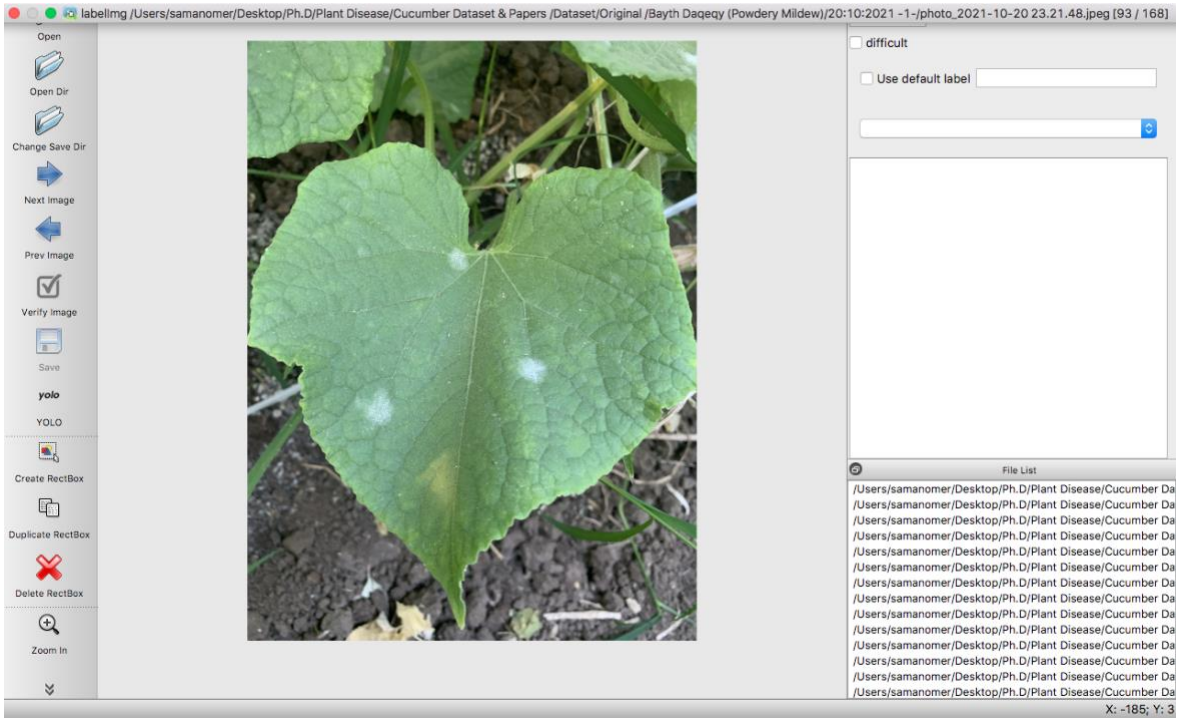


Figure 3. 13 LabelImg tool window

The images labeled using the LabelImg tool can be utilized to train deep learning techniques, particularly those designed for object detection tasks such as YOLO. The annotations derived from the labeled images serve as significant reference data, enabling the models to learn and detect cucumber leaf disease and pest symptoms. The LabelImg software supports image labeling in various formats, including VOC XML and YOLO text file. In our study, we utilized the VOC XML format, which offers the advantage of easy conversion to other formats as needed, as shown in Figure 3.14. Subsequently, the VOC XML file format is converted into a TXT file to ensure compatibility with the YOLO model.

In general, LabelImg proves to be as a valuable tool for researchers, data scientists, and developers working on computer vision projects. It simplifies the image annotation process, making it easier to manage labeled image datasets for training and building a highly performant object detection models.

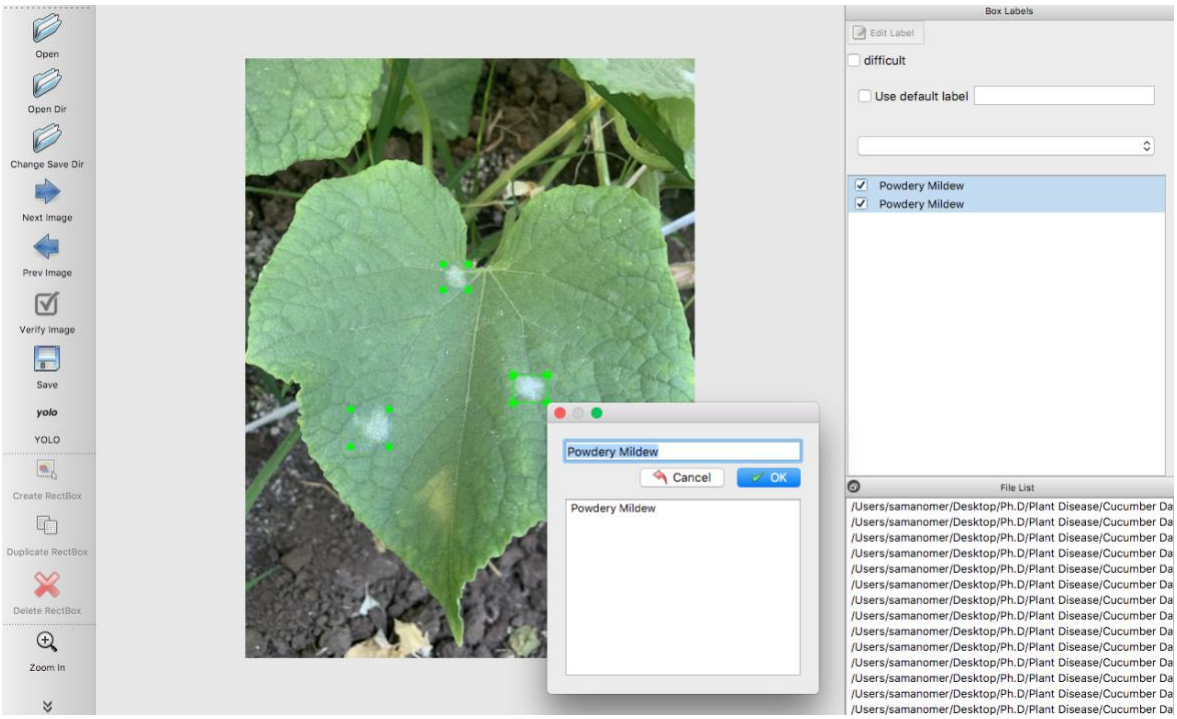


Figure 3. 14 Cucumber leaf disease symptom annotation using LabelImg tool

The performance and generalization of the trained deep learning models are significantly influenced by the quality and accuracy of the data labels. A well-defined and well-structured annotation process is crucial to ensure the dependability and high quality of labeled data. This, in turn, results in more robust and accurate predictions from the deep learning model.

3.6. Evaluation Metrics

Evaluation metrics are quantitative measurements used for evaluating a model's or algorithm's performance and efficacy in various fields, including deep learning. These performance metrics are widely recognized and considerably utilized in academic and professional studies. In Chapters 4 and 5, these metrics are used to demonstrate how well the proposed model performed. In the process of evaluating a model's performance based on a

confusion matrix, several criteria can be derived. These criteria are derived from the values present in the matrix. A confusion matrix is a tabular representation that summarizes the performance of a classification model. It is a square matrix that provides a detailed analysis of the predicted and actual class labels for a test set data. Table 3.4 illustrates the general form of two classes for the proposed model using confusion matrix.

Table 3. 4. Typical Cucumber leaf diagnosing evaluation metrics

		Predicted Class	
		Healthy	Unhealthy
Actual class	Healthy	*A* (TP)	*B* (FN)
	Unhealthy	*C* (FP)	*D* (TN)

Accuracy, precision, recall, and F_1-score are the commonly used evaluation criteria based on the confusion matrix for identifying cucumber leaf diseases and pests. From Table 3.4, *A* represents the count of healthy leaves that are accurately predicted, which is also referred to as true positive (TP), and *B* represents the count of healthy leaves that were wrongly predicted as unhealthy leaves; it is also referred to as false positive (FP). Meanwhile, *C* represents the count number of unhealthy leaves that were wrongly predicted as healthy; it is also referred to as false negative (FN). Finally *D* is the count of unhealthy leaves that were correctly predicted, it is also referred to as true negative (TN).

Accuracy is a metric used to evaluate the model classification rate; it is the rate at which the model correctly predicted. Accuracy is computed by dividing the count of accurate predictions by the total count of predictions. The following formula is the definition of accuracy (Agarwal *et al.*, 2021):

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TN} + \text{TP} + \text{FN} + \text{FP}} \tag{3.1}$$

While precision quantifies the number of positive class predictions of the model that actually belongs to the positive class, it is a true positive rate divided by the total true positive and false positive, as defined in the equations (3.2) (Ma *et al.*, 2018). Meanwhile, recall quantifies the number of positive class predictions made out of all positive classes in the dataset (all instances that have been classified as positive). Recall becomes crucial in situations where the impact of false negatives is high and there is a need to capture all positive instances, as defined in the equations (3.3) (Agarwal *et al.*, 2021). F_1-score is also a measure of model's accuracy on a dataset. It combines the precision and recall metrics into one metric. The formula of F_1-score is shown by the following equation (3.4) (Ma *et al.*, 2018):

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{3.2}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{3.3}$$

$$\text{F1} - \text{score} = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3.4}$$

In this dissertation, recall, precision, and mAP@0.5 were the three measures used to assess the improved YOLOv5 model's detection ability. Precision is the ability of a model to identify only the relevant objects, and the percentage of all detection results that are accurately detected is referred to as the precision, which is defined in equation (3.2). Where TP denotes the number of successfully recognized positive samples, FP indicates that the number of

negative samples that are falsely detected, and FN indicates the number of positive samples that are not detected, in another word, it means an object that should have been detected but not detected. Recall is a measurement of how accurately a positive prediction is made in the presence of a positive input. That simply refers to how well the model can detect it which is defined in the equation (3.3).

In object detection tasks, intersection over union (IoU) is a widely used metric for assessing bounding box accuracy evaluation predictions. The estimated bounding box overlap with the ground truth bounding box was evaluated in IoU by calculating the ratio of their shared area to their union area, as defined in the equation (3.5) (Li and Fang, 2023). Typically, the mAP value is determined at an IoU of 0.5, denoted as mAP@0.5. IoU plays a crucial role in computing mAP. The average precision (AP) is a common evaluation metric used to assess the model's accuracy and in detecting objects. It measures the precision-recall trade-off, indicating how well the model balances precision and recall, as defined in equation (3.6) (Zhou *et al.*, 2023). The mAP was calculated by determining the average precision (AP) of each class and then averaging over a number of classes, as defined in equation (3.7) (Zhou *et al.*, 2023). mAP@0.5 is the mean precision when the IoU is equal to 0.5.

$$\mathrm{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}} = \frac{\mathrm{X} \cap \mathrm{Y}}{\mathrm{X} \cup \mathrm{Y}} \qquad (3.5)$$

$$\mathrm{AP} = \sum_{\mathrm{i}=0}^{\mathrm{i}=\mathrm{n}-1} [\mathrm{Recall(i)} - \mathrm{Recall(i+1)}] * Precesion(i) \qquad (3.6)$$

$$\mathrm{mAP} = \frac{1}{\mathrm{N}} \sum_{\mathrm{i}=1}^{\mathrm{N}} \mathrm{AP(i)} \qquad (3.7)$$

As shown in equation (3.5), X represents the prediction box and Y represents the ground-truth box, while the area of overlap (intersection of the two boxes) is shown in denominator and the area of union (set of the two boxes) represents the numerator. While, in equation (3.6) recall(i) indicates current recall value and recall(i+1) indicates next recall value. Where, In equation (3.7) N represents all number of disease classes, AP(i) stands for the average precision of each diseased class.

3.7. Experimental Equipment

In this study, two different environment programming platform tools, and two desktop computers with different properties have been used to conduct the experiments and evaluation of the models, for cucumber leaf disease and pest diagnosing and detection.

In cucumber leaf disease and pest recognition system, MATLAB, a widely used and powerful programming language for scientific computing, was chosen as the primary tool to implement the models. MATLAB offers a rich set of functions and libraries, making it suitable for various computational tasks, including machine learning and deep learning. To conduct the experiments and evaluate the models, a high-performance computer system was employed. The computer was running on the Windows 10 operating system, and it was equipped with an Intel Core i7 CPU, which operated at a speed of 3.20 GHz. The Intel Core i7 series is known for its robust performance and multitasking capabilities, making it ideal for handling complex computational tasks efficiently. The computer was equipped with an NVIDIA GeForce GTX 1160 GPU. The GPU, with its 6 GB dedicated memory, with 32 GB of RAM.

Furthermore, in cucumber leaf disease and pest detection system, the PyTorch framework and the YOLOv5 environment were utilized for conducting the experiments. The implementation was done in Python,

specifically using version 3.8 of the language. The experiment was conducted on a desktop computer, running on the Windows 10 operating system. The computer used for the experiment was equipped with an Intel Core i7-8700K CPU, operating at a speed of 3.70 GHz. In terms of graphics card processing, the computer has an NVIDIA GeForce GTX 1080 Ti GPU. This GPU had a dedicated memory capacity of 2 × 6 GB, with 48 GB of RAM.

3.8. Analysis and Validation of Results

A statistical significance test, specifically the chi-square test, was carried out to evaluate the improvements obtained from the proposed and designed CNN model. Pre-trained models termed as baseline was developed and used to benchmark against the proposed CNN model. The effectiveness of the CNN's performance was verified through experimental evaluations, which were statistically analyzed using the chi-square test using a local dataset and public available dataset. This test is commonly employed to compare observed outcomes with expected results.

3.9. Summary

In this chapter, a research framework has been presented employed in this study, which serves as a comprehensive guide for the detailed investigations outlined in Chapters 4 and 5. It begins by providing background information on the datasets constructed and used in this study, followed by the focus on the design considerations and process involved in developing the proposed models for cucumber leaf disease and pest diagnosis systems. This includes important steps including data collection, data augmentation as a pre-processing approach. Furthermore, the proposed model design considerations were evaluated and examined using the commonly recognized performance metrics

which were employed in the relevant field. The modular structure of the research plan enables specific attention to be given to each aspect of the investigation. In conclusion, the research materials and methodology were discussed in this chapter which has been thoughtfully developed to effectively unify the investigation, aiming to achieve an enhanced model for an adaptive and automated recognition and detection cucumber leaf diseases and pests system.

CHAPTER FOUR

4. CUCUMBER LEAF DISEASE AND PEST RECOGNITION SYSTEM BASED ON TUNED CNN ALGORITHM

4.1. Overview of the Proposed System

In this study, the state-of-the-art based deep learning algorithms have been applied for cucumber disease and pest recognition using leaf disease symptoms. The first objective of this chapter is to examine our proposed system for diagnosing cucumber leaf diseases and pests by fine-tuning the CNN algorithm from scratch. The system takes an image as input, which includes symptoms of cucumber leaf diseases and pests, based on a constructed leaf image. The second objective in this chapter is, two combined CNNs from scratch with three pretrained models such as AlexNet, ResNet-50 and Inception-V3 have been used. The purpose of the developed system is diagnosing diseases accurately and will help farmers to timely control the spreading of cucumber diseases. Additionally, the influence of data augmentation techniques examines the accuracy of recognition within the study. Figure 4.1 gives an overview of the experimental diagnosing procedure for cucumber leaf diseases and pests.

4.2. Data Preparation

In this chapter, two different datasets are used to conduct the model experiments. Foremost, a locally created dataset named dataset-1 that includes five unhealthy leaf classes, i.e. two pest diseases (spider and leaf miner), two fungal diseases (downy mildew and powdery mildew), one viral disease CYSDV, and one healthy leaf class. The total images of the dataset are 4868 images, each class having a sample image number in the range of 350–1493**.** Class diseases' names and sample numbers were described in Table 3.1. The dataset were divided into two different sets: training and testing set. The total images are split into 80% for training and 20% for testing set. A total of 3895 images and 973 images were used for training and testing sets respectively. Secondly, publicly available dataset named dataset-2 (Cucumber Plant Disease

Dataset), which contains two different cucumber classes healthy and diseased. The total images of the dataset are 695 images (Negm, 2020). Class names and sample numbers were described in Table 3.3.

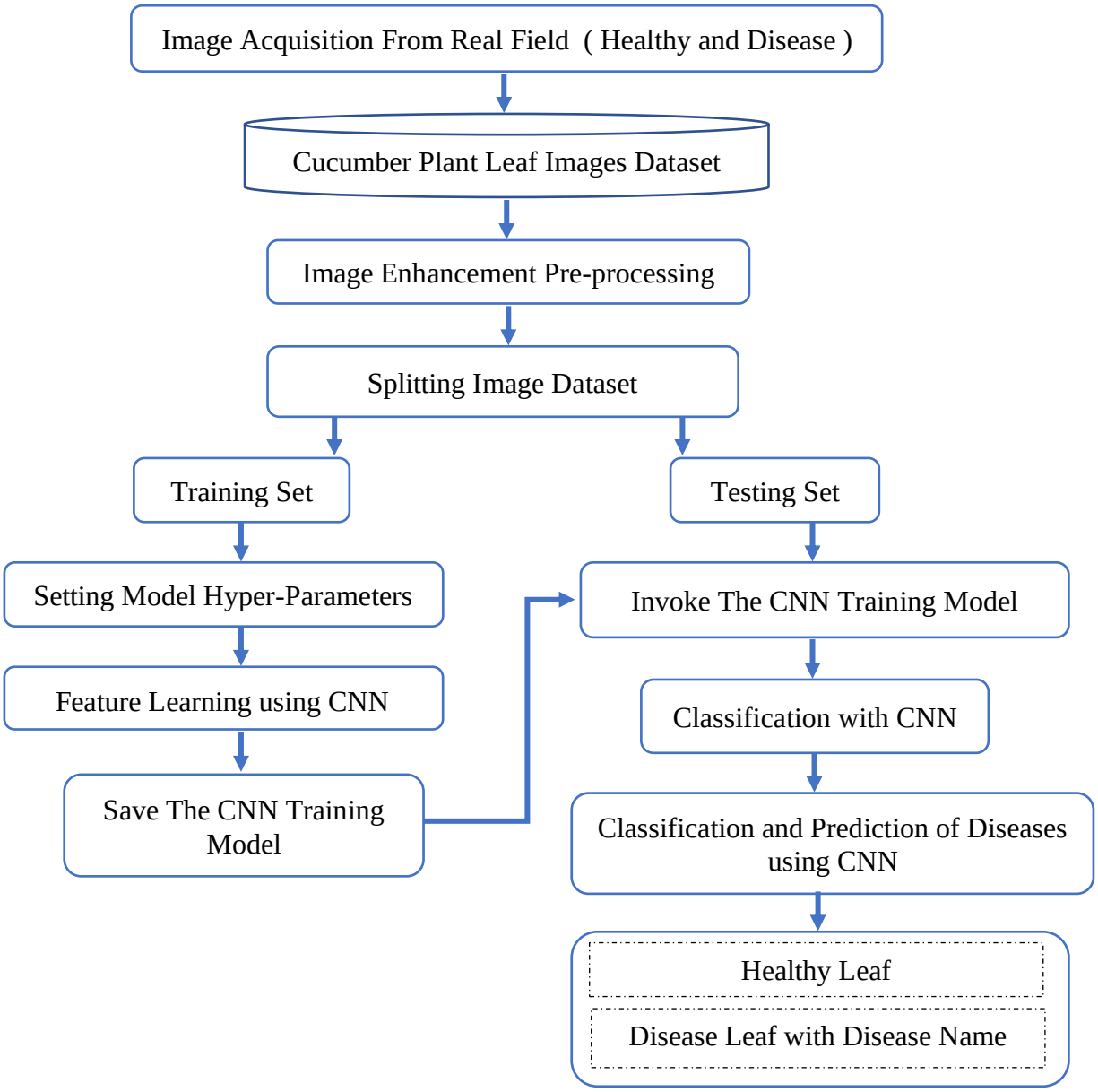


Figure 4. 1 An overview of the procedure of diagnosing cucumber leaf diseases and pests

4.3. Data Pre-Processing

In this study, data augmentation approaches as a pre-processing step has been applied to the datasets before they were used for model training. By utilizing multiple datasets, we aimed to enhance the robustness and reliability of our findings and gain a more comprehensive understanding of cucumber leaf diseases and pests. Data augmentation has been used on both datasets to improve the generalization and adaptability of deep learning models.

a) Pre-Processing of Dataset-1 (Self-Made Dataset)

In order to illustrate the effect of the number of sample images of each class on the performance of the model results, it has augmented according to the maximum image number of class samples. For these experiments, the training sample number of powdery mildew has been chosen because it contains the maximum sample number of 1493 images. All other training classes increased to the range of (1320 –1418) images. Likewise, the number of test class images increased to the range of (265–285) images. All training and testing augmented class samples have nearly the same number of images. For this purpose, five different augmentation methods have been applied on each training and testing images such as rotation, shear, zoom, flip horizontally, and brightness, as illustrates in Figure 4.2. The augmented dataset was divided into training and testing sets, approximately 80% and 20%, respectively. The total image samples of a new augmented dataset increased to 9927 images, and among them: 8267 images were used for training, and 1660 images for testing, as shown in Table 4.1. Finally, images were resized to 227 × 227 × 3.

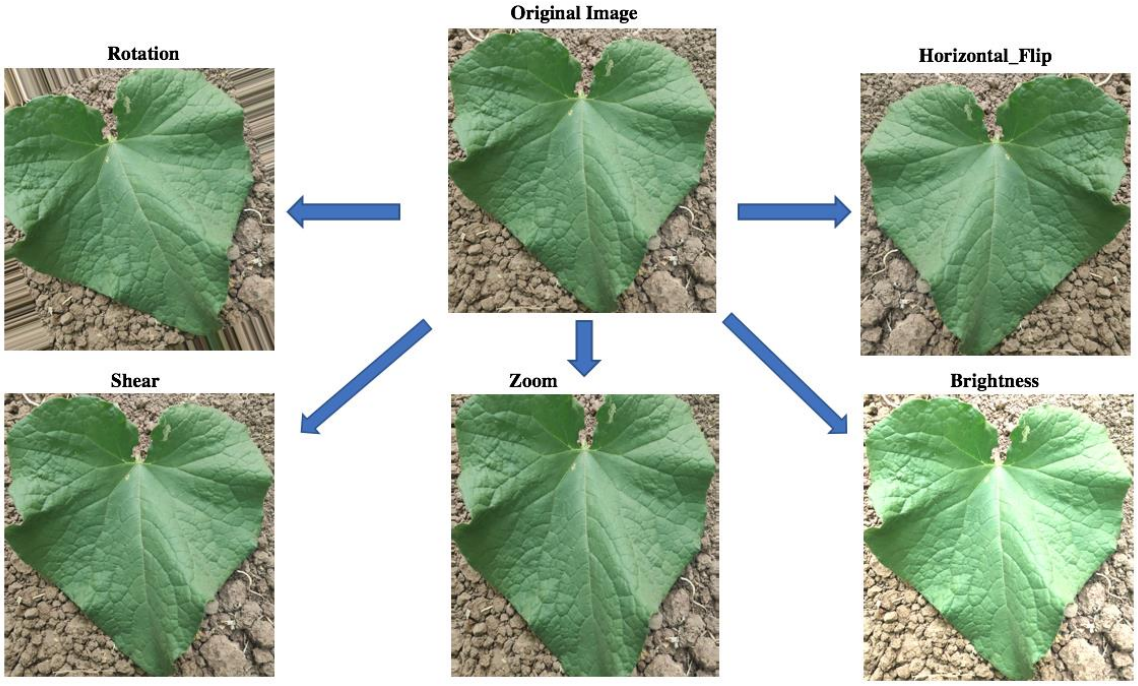


Figure 4. 2 Illustrates the effects of data augmentation techniques on one image of the dataset

Table 4. 1 Statistics of Dataset-1 used for model performance

Class No.	Class Name	Original dataset samples	Samples After Augmentation		
			dataset samples	Training images	Testing images
1	Spider	610	1746	1461	285
2	Leaf Miner	886	1702	1418	284
3	Downy mildew	349	1676	1400	276
4	Powdery mildew	1493	1599	1320	279
5	CYSDV	693	1593	1328	265
6	Healthy	837	1611	1340	271
	Total	4868	9927	8267	1660

b) Pre-Processing of Dataset-2 (Cucumber Plant Disease Dataset)

To address the issue of small sample number of Dataset-2 images, the data augmentation process had been performed on training set only to enlarge data samples and improve accuracy performance. Five different augmentation techniques have been applied on each training image such as rotation, shear, zoom, flip horizontally, and brightness. The training images are increased from 557 images into 3308 image samples, as shown in Table 4.2. Finally, images were resized to 227 × 227 × 3.

Table 4. 2. Images of training and testing in each class of dataset-2

Class No.	Class Name	Original image	Augmented training images	Testing images
1	Good Cucumber	343	1636	68
2	Ill cucumber	352	1672	70
	Total	695	3308	138

4.4. Cucumber Leaf Disease Recognition using Proposed Methods

The increasing popularity of deep learning models leads to the well-known implementation of CNN as the primary tool for image analysis and classification. CNNs have demonstrated remarkable performance in different classification tasks, but despite their immense potential, they still face several challenges. The challenges arise from the extensive scale of the networks, characterized by numerous layers and a multitude of parameters, alongside the concerns involve overfitting and limited generalization abilities. Furthermore, researchers are increasingly concerned about the improving of CNNs on their data (Mikołajczyk and Grochowski, 2018). To address these issues and

improving CNNs performance in image recognition, researchers have actively made modifications to the deep learning networks.

The development of designing an intelligent and automated diagnosing cucumber leaf diseases and pest system was achieved by effectively classifying unhealthy and healthy cucumber leaves. In this section, we present our proposed cucumber leaf disease and pest recognition system. The system takes a leaf image as input that comprises various symptoms associated with cucumber leaf diseases and pests. The main objective of the proposed system is to predict whether the cucumber leaves are healthy or unhealthy, and identifying specific types of cucumber leaf diseases and pests such as (spider, leaf miner, downy mildew, powdery mildew, and CSYDV). During the implementation phase, this study, proposes two CNN architectures from scratch, named 1-CNN and 2-CNN which are utilizing principles derived from deep learning methodologies, to diagnose the cucumber leaf diseases and pests. Those proposed CNN models includes an input layer, convolutional layers, batch normalizations layer, Activation function, pooling layers, dropout layers, fully connected layers, and an output layer.

4.4.1. Proposed 1-CNN Architecture

The proposed 1-CNN model has taken an input image with $227 \times 227 \times 3$ pixel size. It contains five convolutional layers, each with a distinct number of filters and window sizes. In the convolutional layer, an input image is convolved with multiple kernels, resulting in the generation of convoluted images (feature maps) associated with each filter. The convolution operation involves computationally intensive calculations, which become more complex as the image size and the number of convolutional layers increase. The adjustable weights within the filter serve as the parameters in this context.

The filter sizes are 7x7, 5x5, 5x5, 5x5, and 3x3 for the five layers respectively, while, the numbers of filters are 20, 32, 40, 64, and 96 with padding 2, 2, 2, 1, and 1, respectively. The complete feature maps are obtained by using several different kernels. Mathematically, the feature value at location (i , j) in the k-th feature map of the l-th layer, $Z^{l}_{i,j,k}$, is calculated as follows (Gu et al., 2018):

$$Z^{l}_{i,j,k} = W^{l^{T}}_{k} X^{l}_{i,j} + b^{l}_{k} \qquad (4.1)$$

Where W^{l}_{k} and b^{l}_{k} are the weight vector and bias term of the $k_{\text{-th}}$ filter of the $l_{\text{-th}}$ layer respectively, and $X^{l}_{i,j}$ is the input patch centered at location (i,j) of the $l_{\text{-th}}$ layer. Each convolutional layer's dimension output of its feature maps was calculated using the formula explained in equation (4.2).

$$D_{out} = \left(\left(\frac{D_{in} - D_f + 2P}{S}\right) + 1\right) \qquad (4.2)$$

where D_{out} is the number of output feature map dimension, D_{in} is the number of input image dimension, D_f is the convolutional filter size dimension, P is the amount of convolutional padding used on the border, and S is the convolutional stride number. In the next step, batch normalization used.

The process of training CNN is complicated due to the dynamic nature of input distributions in each layer throughout the training process, resulting from the modifications in parameters of preceding layers. Furthermore, to enhance the stability, performance, and efficiency of the CNN, the proposed architecture included the batch normalization (BN) layer. By implementing BN, effectively leverage higher learning rates and alleviate concerns regarding initialization, thereby reducing the necessity for dropout in certain cases. BN achieves equivalent accuracy when applied to a state-of-the-art image classification

technique, a substantially reduced number of training steps with 14 times, in regard to efficiency, it outperforms as the original model by a notable margin (Ioffe and Szegedy, 2015). For activation function, rectified linear unit (ReLU) is used for all layers, which is the most well-known activation function employed in CNN for feature learning. ReLU is calculated based on the following equation (4.3) (Gu *et al.*, 2018):

$$a_{i,j,k} = \max (x_{i,j,k} , 0) \qquad (4.3)$$

Where $x_{i,j,k}$ is the input of the activation function at location (i, j) on the k_{-th} channel. The computational overheads of the network is reduced by reducing the overall number of parameters based on employed Maxpooling layers with the CNN model. This reduction aids in optimizing computational resources and enhancing the efficiency of the model. The max pooling layer is used and sequentially applied after layers in varying sizes 3, 3, 3, 3, and 2 with stride 2 to extract features automatically and minimize the amount of parameters and computation in the network.

The last three layers have used dropout technique with the rate 50%, 50%, and 40% respectively. During the forward pass training process, a few neurons randomly were dropped out from the network to reduce the model size, those neuron weights will not be updated during backward pass. Dropout layer is employed to overcome the overfitting issue and offers an efficient means of effectively merging the predictions made by multiple distinct neural networks. After performing convolutional and pooling layers, the output of the previous layers have been fed into a fully connected layers. It is utilized to gather and transform all the features from the preceding layers into a vector of one dimension. The proposed CNN architecture consists of two fully connected layers, which have 512 and 256 neurons respectively, with the last dense layer. A SoftMax function used in the final dense layer to calculate the estimated

probability for five different cucumber leaf diseases and one healthy leaf type. The proposed 1-CNN architecture is shown in Figure 4.3.

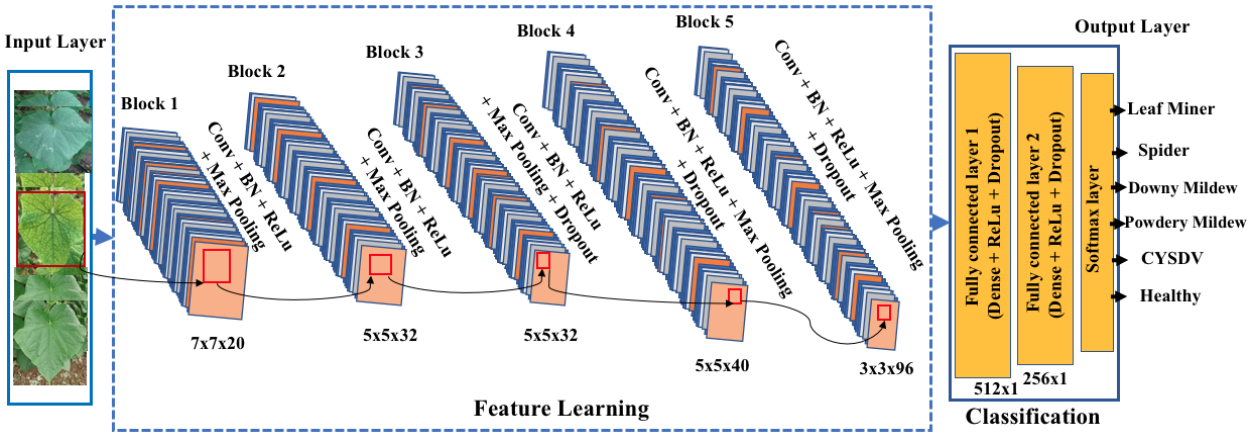


Figure 4. 3 Proposed 1-CNN architecture

From Figure 4.3, it can be seen that the proposed 1-CNN architecture has five blocks (where the first two blocks including convolution layers, BN, Activation, and Max pooling, while the last three blocks includes convolution layers, BN, Activation, Max pooling, and Dropout layers). In addition, Table 4.3. demonstrates the proposed 1-CNN architecture descriptions for better understanding the recognition system.

Table 4. 3. Description of the proposed CNN parameter architecture

Layer Name	Image size	Learnable Parameters	Total Parameters
Input image	227 x 227 x 3	0	0
Block_1			
Convolution2D (7x7) x 20	225 x 225 x 20	Weights 7*7*3*20 Bias 1*1*20	2960
Batch Normalization	225 x 225 x 20	Offset 1*1*20 Scale 1*1*20	40
Activation ReLU	225x 225 x 20	-	0

Max Pooling	112 x 112 x 20	-	0
Block_2			
Convolution2D (5x5) x 32	112 x 112 x 32	Weights 5*5*20*32 Bias 1*1*32	16032
Batch Normalization	112 x 112 x 32	Offset 1*1*32 Scale 1*1*32	64
Activation ReLU	112 x 112 x 32	-	0
Max Pooling	55 x 55 x 32	-	0
Block_3			
Convolution2D (5x5) x 40	55 x 55 x 40	Weights 5*5*32*40 Bias 1*1*40	32040
Batch Normalization	55 x 55 x 40	Offset 1*1*40 Scale 1*1*40	80
Activation ReLU	55 x 55 x 40	-	0
Max Pooling	27 x 27 x 40	-	0
Dropout	27 x 27 x 40	-	0
Block_4			
Convolution2D (5x5) x 64	25 x 25 x 64	Weights 5*5*40*64 Bias 1*1*64	64064
Batch Normalization	25 x 25 x 64	Offset 1*1*64 Scale 1*1*64	128
Activation ReLU	25 x 25 x 64	-	0
Max Pooling	12 x 12 x 64	-	0
Dropout	12 x 12 x 64	-	0
Block_5			
Convolution2D (3x3) x 96	12 x 12 x 96	Weights 3*3*64*96 Bias 1*1*96	55392
Batch Normalization	12 x 12 x 96	Offset 1*1*96 Scale 1*1*96	192
Activation ReLU	12 x 12 x 96	-	0
Max Pooling	6 x 6 x 96	-	0
Dropout	6 x 6 x 96	-	0
Classification Block			
Fully connected	1 x 1 x 512	Weights 512*3456 Bias 512 * 1	1769984
ReLU	1 x 1 x 512	-	0
Dropout	1 x 1 x 512	-	0
Fully connected	1 x 1 x 256	Weights 256*512 Bias 256 * 1	131328
ReLU	1 x 1 x 256	-	0
Dropout	1 x 1 x 256	-	0
Fully connected	1 x 1 x 6	Weights 6*256 Bias 6 * 1	1542
Softmax	1 x 1 x 6	-	0
Classification Output	1 x 1 x 6	-	0
Total	227 x 227		2073846

- **Training Process (Hyper-parameters)**

The proposed 1-CNN model was trained using a MATLAB script. During the training process of the proposed CNN architecture, the number of layers, convolutional layer parameter values such as the filter size and window size, and pooling window size were considered in the experiments. Furthermore, three other factors to be considered were optimization techniques, batch size and the number of epochs. The number of training images used in single iteration referred to the batch size, while the epochs denote the count of iterations during which the learning algorithm processes the entire training dataset.

In this study, the model was trained using two optimization algorithms, namely Adam and stochastic gradient descent (SDGM), with Adam obtaining the best accuracy recognition results. Initially, the training was conducted for 50 epochs, but instability was observed during the process. Therefore, the number of epochs was fine-tuned and tested with values of 75, 100, 125, 150, and 200, with the best results achieved at 200 epochs. To regulate the model's response to error, a learning rate parameter was employed during the training process, with an optimal value of 0.001. Additionally, an optimal batch size of 32 was utilized to achieve the best performance during training. As can be seen in Table 4.4, various sets of hyperparameters were used to execute the proposed 1-CNN model in terms of achieving the best accuracy result.

Therefore, the chosen CNN architecture has been utilized for the development of the proposed system aimed at diagnosing and recognizing cucumber leaf diseases and pests.

Table 4. 4. Hyper-parameter setting for the proposed 1-CNN

Hyper-parameter	Descriptions
Number of Convolutional Layer	5
Number of Max Pooling Layer	5
Number of Batch Normalization Layer	5
Convolutional Dropout Layer rate	0.5, 0.4, 0.5
Fully Connected Layer	2
Fully Connected Dropout Layer rate	0.5 , 0.5
Activation Function	ReLu
Learning rate	0.001
Max Epoch number	200
Batch Size	32
Optimizer	Adam
Validation Frequency	5
Shuffle Every Epoch	Yes

4.4.2. Proposed 2-CNN Architecture

In this section, a CNN model has been proposed, drawing inspiration from the state-of-the-art CNN algorithms, which have demonstrated exceptional performance. Specifically, the architecture of this CNN model is derived from the 1-CNN model outlined in section 4.4.1. Particularly, in this design, two separate CNN models are employed in parallel and seamlessly merged together named 2-CNN model, resulting in a synergistic combination of their respective capabilities. The architecture of the 2-CNN model comprises five convolutional layers, each with distinct numbers of filters and window sizes. The filter sizes are 7x7, 5x5, 5x5, 5x5, and 3x3 set for all layers of both parallel CNN models respectively; while, the number of filters are (32, 20), (40, 32), (45, 40), (75, 64), and (96,96) with padding 2, 2, 2, 1, and 1, respectively. Batch normalization and ReLu activation function were used for all layers. The max pooling layer was employed and sequentially applied after layers in varying sizes 3, 3, 3, 3, and 2 with stride 2 to extract features automatically in both

models. A dropout layer has been used in the last three layers, with the rate (50,40%),(40,40%), and (50,50%) respectively, to address or reduce the overfitting issue. It is used to dropout a few neurons from the network during the process of training to reduce the size of the models. The output of both parallel CNNs' last layers was combined using a concatenation block and subsequently fed into fully connected layers. The 2-CNN architecture consists of two fully connected layers, which have 512 and 256 nodes respectively, with the last dense layer. A SoftMax function was used in the final dense layer to calculate the estimated probability for five different cucumber leaf diseases and one healthy leaf type. Figure 4.4 showing the proposed 2-CNN architecture.

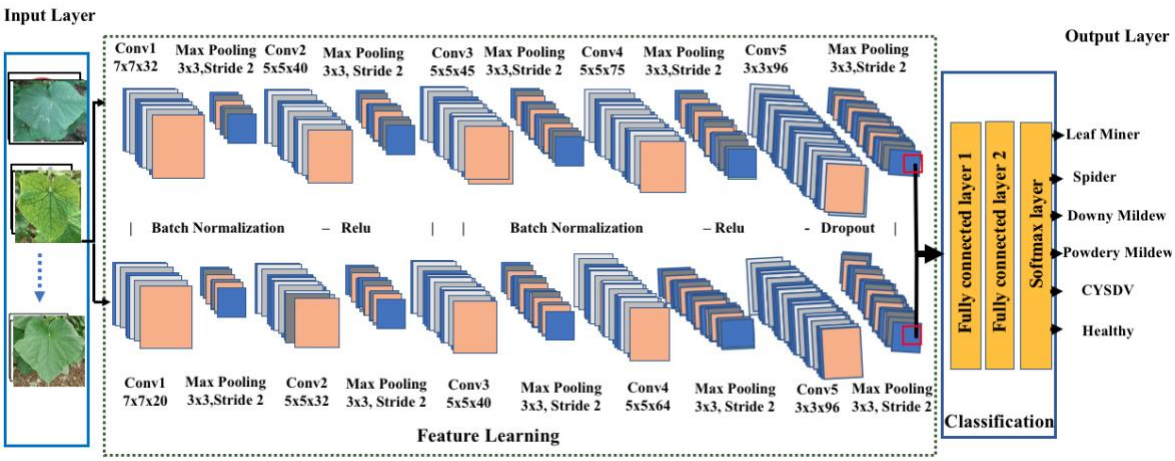


Figure 4. 4 Proposed two combined CNN Architecture (2-CNN)

4.5. Evaluation of the CNN with Transfer Learning Models

This section outlines the material and classification method used in this research. In this study, AlexNet, Inception-V3, and ResNet-50 were utilized as transfer learning models for conventional recognition evaluation on two different datasets. Furthermore, the proposed 1-CNN and 2-CNN models were compared with these models in terms of diagnostic performance of cucumber

leaf diseases and healthy recognition, using the confusion matrix. Accuracy, recall, precision, and F_1-score were used in the evaluation of the model criteria based on the confusion matrix as they were clearly defined in section 3.6.

4.6. Results and Discussion

This section provides a comprehensive analysis and interpretation of the results that have been empirically obtained from conducting different experiments on both proposed algorithms and pre-trained algorithms. This section aims to offer a thorough comprehension of the results and conclusions obtained from the conducted experiments through particular analysis and investigation.

4.6.1. Result of Cucumber Leaf Disease and Pest Recognition

This section describes the experiments and procedures that we have conducted concerning the proposed system for the recognition of cucumber leaf diseases and pests. To facilitate the experiments, two different datasets containing images of cucumber leaves have been utilized. In this context, each dataset was partitioned, whereby 80% of the available data has been allocated to the training set, and the remaining 20% has been assigned to the testing set.

Numerous experimental tests have been conducted to diagnose cucumber leaf diseases and healthy leaves using the best combination of hyper-parameters to produce the highest recognition accuracy. During the training process of the proposed 1-CNN architecture, the number of layers, convolutional layer parameter values such as the filter size and window size, and pooling window size were considered in the experiments. Two different scenarios have been performed on Dataset-1, and another one has been performed on dataset-2.

In first scenario, all classes are used with different class image numbers from 349 to 1493 (unbalanced data) to calculate model classification accuracy. The proposed 1-CNN performed better than other models on the unbalanced data. In particular, our proposed model attains the accuracy result with 97.53%, which is much higher than other pre-trained algorithms with accuracy, whereas 97.13%, 97.44%, and 96.9% are achieved from F_1-score, recall, and precision based on confusion matrix, respectively, as can be seen in Table 4.5. It also determines that most true values and predicted samples are matched.

According to the performance of each class, the best performance was obtained on the powdery mildew disease class with an F_1-score of 99%. Out of 299 powdery mildew prediction images, 99.7% were correct. In addition, leaf miner and healthy classes observed similar recognition results, which are correctly classified as the images of 98.8% accuracy out of 177 and 167 predictions, respectively. In case of other disease classes such as spider, downy mildew and CYSDV achieved a predictive accuracy of 98.3%, 94.4%, and 91.4%, respectively. It can be seen in Table 4.5 that 1-CNN demonstrated higher results on the powdery mildew, leaf miner, and healthy classes; meanwhile, these classes have a larger amount of images than the others.

Table 4. 5. Confusion matrix of the 1-CNN with unbalanced data on test dataset-1

Actual Classes	Predicted Classes						Evaluation metrics (%)		
Leaf disease	Downy mildew	Powdery mildew	Leaf miner	Spider	CYSDV	Healthy	Recall	Precision	F_1-score
Downy mildew	67	0	0	0	1	1	97.1	94.4	95.73
Powdery mildew	0	294	1	1	3	0	98.3	99.7	99
Leaf miner	0	1	169	1	5	1	95.5	98.8	97.12
Spider	1	0	1	119	2	0	96.7	98.3	97.5
CYSDV	0	0	0	0	138	0	100	91.4	95.5
Healthy	3	0	0	0	2	162	97.0	98.8	97.9
Total Accuracy (%) : 97.53									

Second scenario: involves examining whether the proposed 1-CNN was influenced by the amount of data. This aims to ensure the impact of data quantity on the model, all experiments were performed on the proposed model with balanced data. In this experiment, the data augmentation process was applied separately to the training and test sets of Dataset-1. This was done to ensure that all classes within the datasets have an almost equal number of images. An augmented Dataset-1 including 9927 leaf images in total. The training set contained 8267 images, while the testing set comprised 1660 images. Intuitively, the accuracy of the proposed 1-CNN model demonstrated superior performance in comparison to its performance when dealing with unbalanced data. Table 4.6 illustrates the test results. From Table 4.6, it can be seen the 1-CNN algorithm obtained average accuracy result is 98.19%, whereas 98.2%, 98.21%, and 98.21% were achieved from F_1-score, recall, and precision, respectively. In essence, 1-CNN has achieved satisfactory accuracy results because of large data rather than unbalancing data. This result is in accordance with the experimental outcome of (Zhang and Zhang, 2010).

According to the performance of each class as shown in Table 4.6, the best performance was obtained on the downy mildew class with an F_1-score of 99.09%. Out of 276 downy mildew prediction images, 98.2% were correct. In addition, healthy and leaf miner classes have similar recognition results, which are correctly classified images with 100% and 98.5% out of 271 and 284 predictions, respectively. Other disease classes, spider, powdery mildew, and CYSDV, achieved a predictive accuracy of 97.6%, 97.5%, and 97.4%, respectively. The recall value for each class was 100%, 98.6%, 95.1%, 98.9%, 98.9%, and 97.8%, respectively. According to the results, 1-CNN exhibited a higher rate of errors when predicting leaf miner and healthy classes. These misclassifications are attributed to the similarity in color, shape, and vein patterns of the leaves. Leaf miner shares the most similarities with CYSDV and

spider classes, while healthy leaves exhibit resemblances to downy mildew and CYSDV classes.

In contrast, downy mildew and spider class symptom images were correctly classified. The performance of all models was improved significantly. More precisely, 1-CNN accuracy was improved from 97.53% to 98.19%. Based on the results, it can be concluded that the size of the input dataset significantly impacted the results. This observation aligns with the finding that the 1-CNN model can achieve satisfactory results with a large amount of data.

Table 4. 6. Confusion matrix of the 1-CNN model with balanced data on test dataset-1

Actual classes	Predicted Classes						Evaluation metrics (%)		
Leaf disease	Downy mildew	Powdery mildew	Leaf miner	Spider	CYSDV	Healthy	Recall	Precision	F1score
Downy mildew	276	0	0	0	0	0	100	98.2	99.09
Powdery mildew	0	275	2	2	0	0	98.6	97.5	98.04
Leaf miner	0	5	270	4	5	0	95.1	98.5	96.8
Spider	0	2	0	282	1	0	98.9	97.6	98.2
CYSDV	0	0	2	1	262	0	98.9	97.4	98.1
Healthy	5	0	0	0	1	265	97.8	100	98.9
Total Accuracy (%):			98.19						

In the third scenario, to demonstrate the robustness of the 1-CNN model concerning the variety of classes, an additional dataset named Dataset-2 was used along with a data augmentation scheme. For this experiment, 1-CNN and other models have also been tested and implemented on Dataset-2. As expected, the performance of 1-CNN outperformed better on the accuracy compared to that with large class numbers. Table 4.7 presents the test results. It is evident from the confusion matrix that the 1-CNN model achieved an accuracy of 100%. Additionally, the F_1-score, recall, and precision have achieved the highest result which is 100%. Based on the test findings, it is determined that the number of classes in the input dataset was an influenced of

factor, which agreed with the conclusion that 1-CNN can attain adequate results.

Table 4. 7. Confusion matrix of the 1-CNN model on test dataset-2

Actual Classes	Predicted Classes		Evaluation metrics (%)		
Classes	Ill cucumber	Good cucumber	Recall	Precision	F1score
Ill cucumber	70	0	100	100	100
Good cucumber	0	68	100	100	100
Total Accuracy (%): 100					

The obtained test results were achieved from the evaluation of both the proposed and pre-trained models. The analysis led to the conclusion that the proposed 1-CNN model has the superior capability and better performance in diagnosis cucumber diseases, pests and healthy image leaves better than other models (AlexNet,ResNet-50,Inception-V3,and 2-CNN). Particularly, when considering factors such as dataset size (including the number of classes and the quantity of sample images) as well as the presence of both balanced and unbalanced data, as shown in Figure 4.5.

Furthermore, based on Figure 4.5, the performance of the proposed 1-CNN system demonstrates significant improvement as a result of data augmentation techniques applied to both datasets. Additionally, when utilizing a smaller dataset with fewer leaf class types, the learned 1-CNN model achieves superior performance.

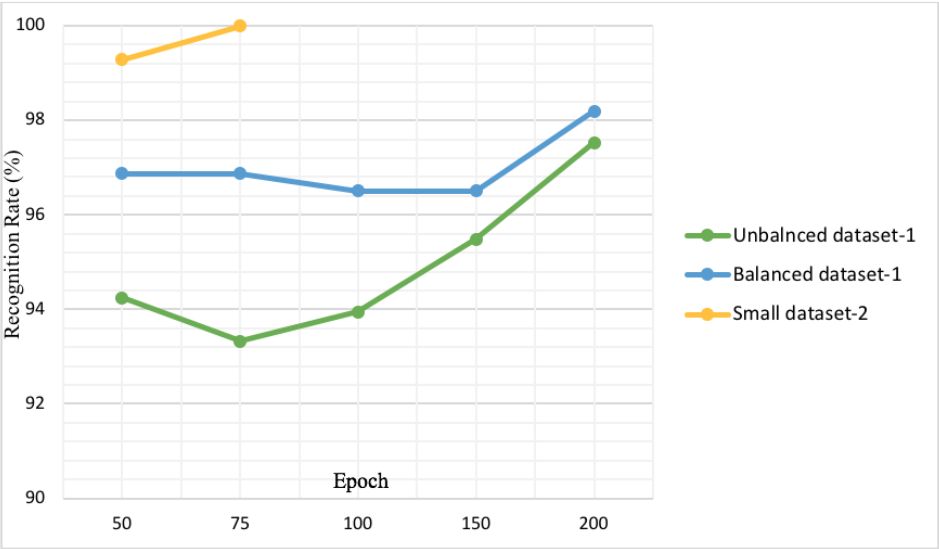


Figure 4. 5 Performance comparison for the trade-off between the number of classes and the quantity of sample images employed in the proposed CNN model

4.6.2. Performance Evaluation of Cucumber Leaf Disease Recognition

The correct diagnosis of cucumber leaf disease is significant to construct a robust and effective model for an automated cucumber leaf disease, pest and healthy leaf recognition. The performance of AlexNet, Inception-V3, and ResNet-50 as pre-trained models and also 1-CNN, 2-CNN as a proposed model from scratch have been tested on both datasets (Datasets-1 and 2). Numerous calculations are performed based on confusion matrix such as accuracy, recall, precision, and the F_1-score as shown in Figures 4.6 – 4.8 and Tables 4.8–4.10. Figure 4.6 results indicate clearly that the proposed 1-CNN model outperforms AlexNet, Inception-V3, ResNet-50, and 2-CNN models in terms of performance.

It can be seen in Table 4.8 that the proposed 1-CNN outperformed better on dataset-1 with unbalanced data than Inception-V3, Resnet-50, 2-CNN, and AlexNet. The test accuracy results were obtained as shown in Figure 4.6 are

97.53%, 97.02%, 96.30%, 95.07%, and 94.24%, respectively, whereas the higher accuracy result which is 97.53% was achieved from the 1-CNN model.

Moreover, the degree of misclassification is examined. From Table 4.8 and Figure 4.6, we can see a relatively large number of misclassifications that were found between the models' performance. It shows that 1-CNN has the lowest misclassification based on accuracy result that is 2.47% compared with other models Inception-V3, ResNet-50, 2-CNN and AlexNet that are 2.98%, 3.70%, 4.93%, and 5.76%, respectively.

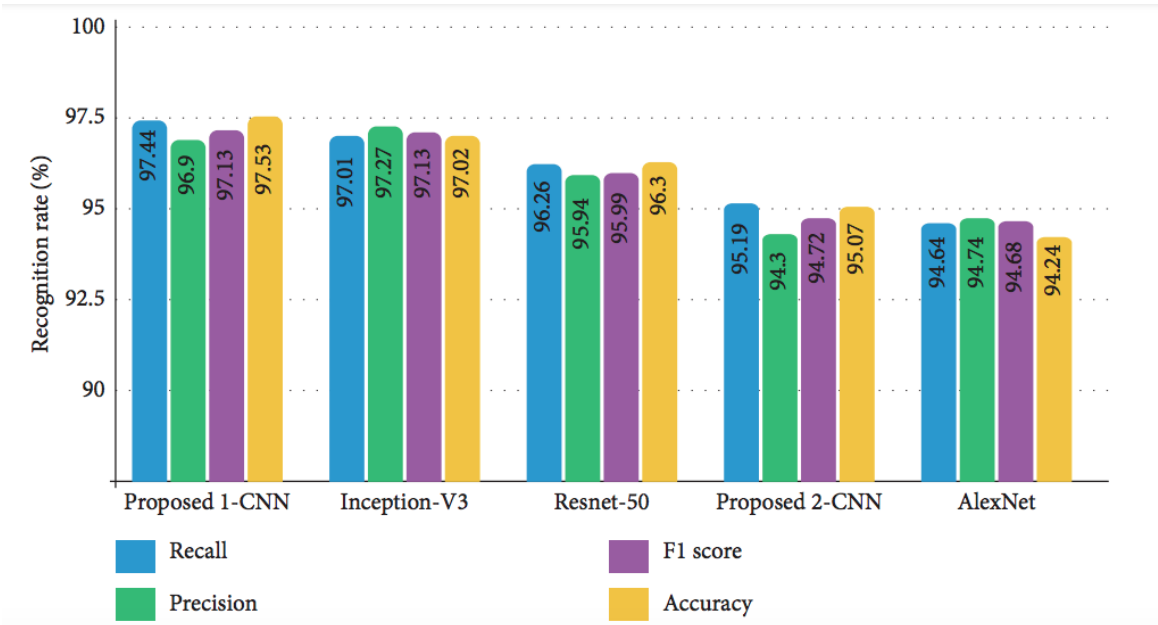


Figure 4. 6 Performance evaluation of different models on unbalanced class images on Dataset-1

Table 4. 8. Test results of the models with unbalanced dataset-1

	Evaluation metrics (%)			
Methods	Recall	Precision	F1 score	Accuracy
Proposed 1-CNN	**97.44**	**96.90**	**97.13**	**97.53**
Inception-V3	97.01	97.27	97.13	97.02
Resnet-50	96.26	95.94	95.99	96.30
Proposed 2-CNN	95.19	94.30	94.72	95.07
AlexNet	94.64	94.74	94.68	94.24

Furthermore, all models were experimented and tested on Dataset-1 with balanced data. The comparative results of the 1-CNN, Inception-V3, ResNet-50, 2-CNN, and AlexNet models have been shown in Figure 4.7. Values have clearly indicated that 1-CNN is more accurate as compared with other models. From Figure 4.7, it can be obviously seen that how 1-CNN significantly improved the recognition accuracy for cucumber leaf disease and pest recognition (from 97.53% to 98.19%). This improvement reflects data augmentation that was applied on Dataset-1 to enlarged and balanced data.

It was revealed in Table 4.9 that the recognition accuracy was 98.19%, 97.77%, 97.53, 96.69%, and 96.14%, respectively on 1-CNN, Inception-V3, ResNet-50, 2-CNN, and AlexNet models. Comparing to the conventional models, 1-CNN demonstrated superior results of recognition accuracy. By analyzing the results showed in Figure 4.7 and Table 4.9, It is evident that when the quantity of data samples rises, the recognition accuracy rate of the proposed 1-CNN model demonstrates an upward trend. The findings demonstrate a positive relationship between the quantity of data samples and the accuracy rate of the proposed 1-CNN model. In other words, as more data samples are incorporated, the proposed model enhances proficiency in accurately recognizing and classifying cucumber leaf diseases and pests, thus observation suggests that a larger dataset positively impacts the model's performance and effectiveness.

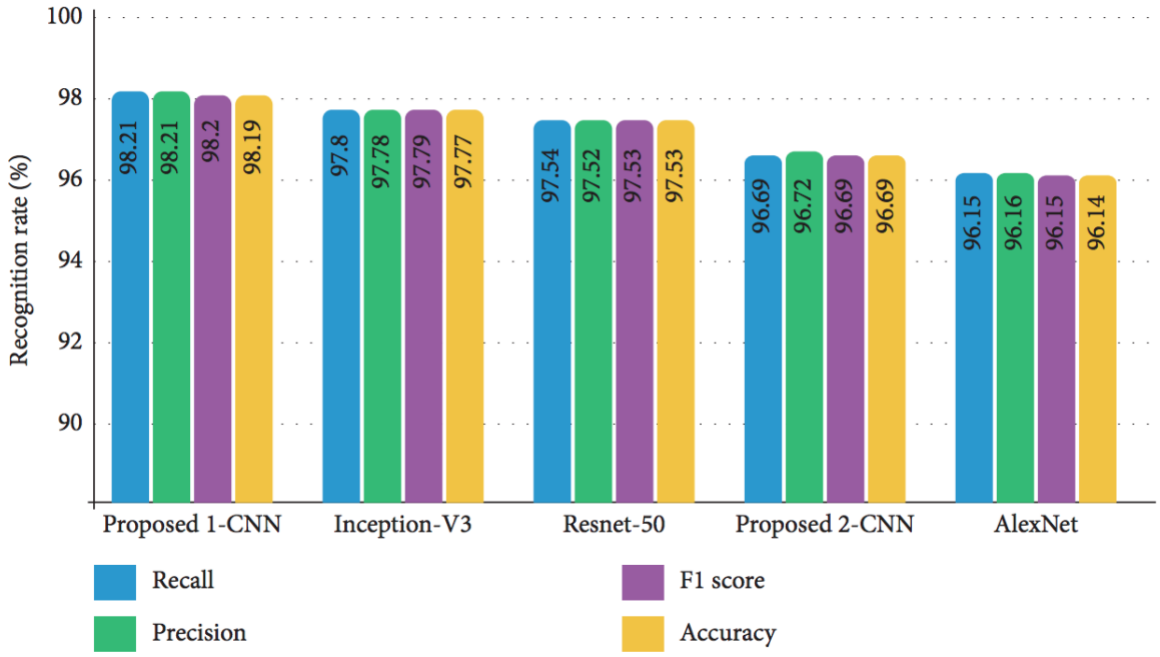


Figure 4. 7 Performance Evaluation of Different models on balanced images of the dataset-1

Table 4. 9. Test results of the models with balanced dataset-1

	Evaluation metrics (%)			
Methods	Recall	Precision	F1 score	Accuracy
Proposed 1-CNN	**98.21**	**98.21**	**98.20**	**98.19**
Inception-V3	97.8	97.78	97.79	97.77
Resnet-50	97.54	97.52	97.53	97.53
Proposed 2-CNN	96.69	96.72	96.69	96.69
AlexNet	96.15	96.16	96.15	96.14

In order to demonstrate the impact and show the effectiveness of class numbers for the performance authenticity of the 1-CNN model, a series test experiments were conducted using 1-CNN model based on Dataset-2. The objective of these experiments, as illustrated in Figure 4.8 and detailed in Table 4.10, was to assess the efficacy of the 1-CNN model. The findings illustrated in Figure 4.8 show that precision, recall, F_1-score, and accuracy result for the AlexNet model were worse. This indicates that the proposed 1-CNN model is more accurate at classifying cucumber leaf diseases and pests than the AlexNet

and other models, emphasizing its effectiveness and superiority in handling varying class numbers present in the dataset.

From Figure 4.8, it can be obviously and clearly seen that how 1-CNN recognition accuracy was significantly improved for cucumber leaf disease and healthy leaf recognition (from 98.19% to 100%). This improvement reflects decreasing the class number and image samples on Dataset-2. According to the results that are shown in Table 4.10, 1-CNN model has obtained the accuracy result with 100%, while 2-CNN, Inception-V3, ResNet-50, and AlexNet have achieved 99.28%, 99.28, 98.55%, and 97.10%, respectively. From the achieved results in Figure 4.8 and Table 4.10, we can see a relatively large number of misclassifications that were found between models accuracy performance. 1-CNN has no misclassification result compared with other models.

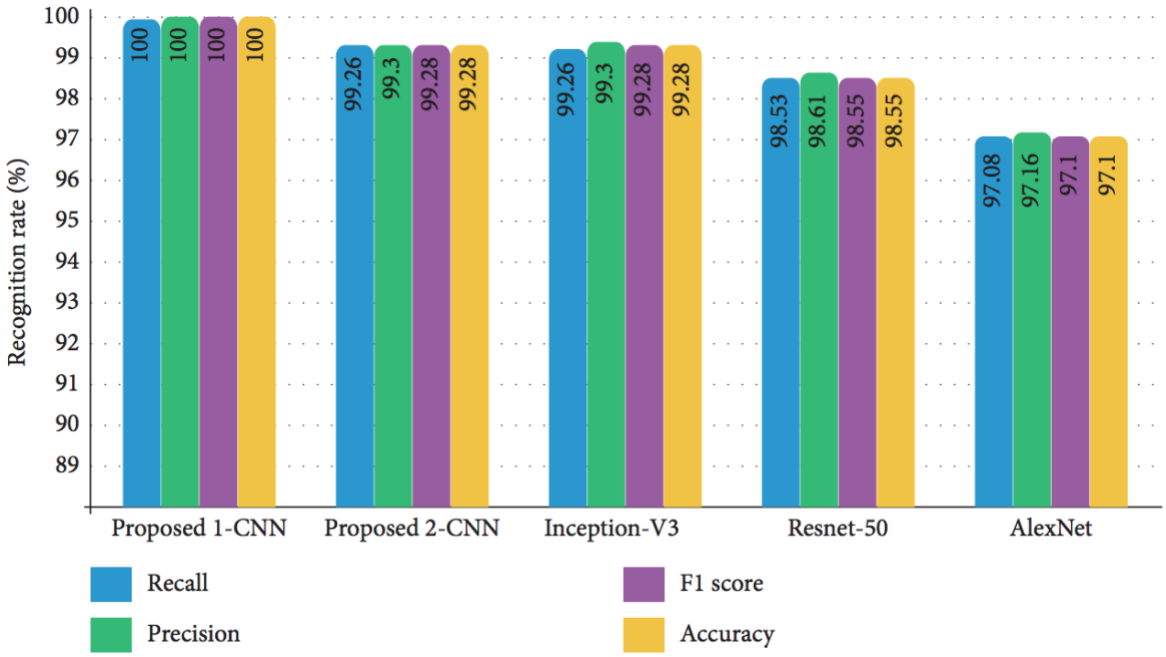


Figure 4. 8 Performance Evaluation of Different models on images of dataset-2

By analyzing the results showed in Figure 4.8 and Table 4.10, it indicates that the recognition accuracy rate of the proposed 1-CNN model shows an upward trend as the number of classes decreases. In addition, the presented test results provide evidence indicating that both the 1-CNN and 2-CNN models had superior performance when admitted with situations involving a fewer class numbers and image samples. These results obviously determine that in scenarios characterized by fewer classes, both the 1-CNN and 2-CNN models demonstrate well performance and accuracy.

Table 4. 10 Results of testing models in images of dataset-2

	Evaluation metrics (%)			
Methods	Recall	Precision	F1 score	Accuracy
Proposed 1-CNN	**100**	**100**	**100**	**100**
Proposed 2-CNN	99.26	99.30	99.28	99.28
Inception-V3	99.26	99.3	99.28	99.28
Resnet-50	98.53	98.61	98.55	98.55
AlexNet	97.08	97.16	97.1	97.1

The performance of the pre-trained models (AlexNet, Resnet-50, and Inception-V3) based on confusion matrix regarding the abovementioned scenarios have been illustrated in Tables 4.11-4.13. Table 4.11 illustrates the performance of AlexNet model based on confusion matrix for unbalanced and balanced dataset-1 and dataset-2.

Table 4. 11. Performance of AlexNet model on both dataset-1 and dataset-2

Models	Confusion Matrix							
Unbalanced Dataset-1			Predicted Classes					
		Leaf disease	Downy mildew	Powdery mildew	Leaf miner	Spider	CYSDV	Healthy
	Actual Classes	Downy mildew	68	0	0	0	0	1
		Powdery mildew	0	284	8	0	5	2
		Leaf miner	0	7	156	2	10	2
		Spider	0	1	0	121	1	0
		CYSDV	0	11	5	0	122	0
		Healthy	1	0	0	0	0	166
	Recall(%): 94.64		Precision(%):94.74		F1_Score(%): 94.68		Accuracy(%): 94.24	
Balanced Dataset-1			Predicted Classes					
		Leaf disease	Downy mildew	Powdery mildew	Leaf miner	Spider	CYSDV	Healthy
	Actual Classes	Downy mildew	273	0	0	0	0	3
		Powdery mildew	0	257	16	2	3	1
		Leaf miner	0	3	267	3	10	1
		Spider	0	2	0	281	2	0
		CYSDV	0	4	6	1	254	0
		Healthy	6	1	0	0	0	264
	Recall(%): 96.15		Precision(%): 96.16		F1_Score(%):96.15		Accuracy(%):96.14	
Dataset-2			Predicted Classes					
	Actual class	Classes	Ill cucumber	Good cucumber				
		Ill cucumber	69	1				
		Good cucumber	3	65				
	Recall(%): 97.08		Precision(%):97.16		F1_Score(%):97.10		Accuracy:97.10	

Table 4.12 illustrates the performance of ResNet-50 model based on the confusion matrix for unbalanced and balanced dataset-1 and dataset-2.

Table 4. 12. Performance of ResNet-50 model on both dataset-1 and dataset-2

Models	Confusion Matrix							
Unbalanced Dataset-1			Predicted Classes					
		Leaf disease	Downy mildew	Powdery mildew	Leaf miner	Spider	CYSDV	Healthy
	Actual Classes	Downy mildew	67	0	0	0	0	2
		Powdery mildew	0	296	2	0	1	0
		Leaf miner	1	3	152	6	13	2
		Spider	0	1	0	121	0	1
		CYSDV	0	1	0	2	135	0
		Healthy	1	0	0	0	0	166
	Recall(%): 96.26		Precision(%): 95.94		F1_Score(%): 95.99		Accuracy(%): 96.30	
Balanced Dataset-1			Predicted Classes					
		Leaf disease	Downy mildew	Powdery mildew	Leaf miner	Spider	CYSDV	Healthy
	Actual Classes	Downy mildew	276	0	0	0	0	0
		Powdery mildew	0	267	10	0	2	0
		Leaf miner	0	5	266	2	10	1
		Spider	0	0	0	285	0	0
		CYSDV	0	3	5	0	257	0
		Healthy	3	0	0	0	0	268
	Recall(%): 97.54		Precision(%): 97.52		F1_Score(%): 97.53		Accuracy(%): 97.53	
Dataset-2			Predicted Classes					
	Actual class	Classes	Ill cucumber	Good cucumber				
		Ill cucumber	70	0				
		Good cucumber	2	66				
	Recall(%): 98.53		Precision(%):98.61		F1_Score(%):98.55		Accuracy:98.55	

Table 4.13 illustrates the performance of Inception-v3 model based on the confusion matrix for unbalanced and balanced dataset-1 and dataset-2.

Table 4. 13. Performance of Inception-v3 model on both dataset-1 and dataset-2

Models	Confusion Matrix							
Unbalanced Dataset-1			Predicted Classes					
		Leaf disease	Downy mildew	Powdery mildew	Leaf miner	Spider	CYSDV	Healthy
	Actual Classes	Downy mildew	67	0	0	0	0	2
		Powdery mildew	0	291	5	0	3	0
		Leaf miner	0	4	166	2	3	2
		Spider	0	3	1	119	0	0
		CYSDV	0	0	1	0	134	3
		Healthy	0	0	0	0	0	167
	Recall(%): 97.01		Precision(%): 97.27		F1_Score(%):97.13		Accuracy(%): 97.02	
Balanced Dataset-1			Predicted Classes					
		Leaf disease	Downy mildew	Powdery mildew	Leaf miner	Spider	CYSDV	Healthy
	Actual Classes	Downy mildew	276	0	0	0	0	0
		Powdery mildew	0	269	8	0	2	2
		Leaf miner	0	3	271	3	4	3
		Spider	0	6	1	278	0	0
		CYSDV	0	0	2	0	263	0
		Healthy	4	0	1	0	0	266
	Recall(%): 97.08		Precision(%): 97.78		F1_Score(%): 97.79		Accuracy(%): 97.77	
Dataset-2			Predicted Classes					
	Actual class	Classes	Ill cucumber		Good cucumber			
		Ill cucumber	70		0			
		Good cucumber	1		67			
	Recall(%): 99.26		Precision(%):99.30		F1_Score(%):99.28		Accuracy:99.28	

We note that the proposed model is well fitting the training data, while the validation loss indicates how well the model fits with new data for both datasets-1 and 2 as shown in Figures 4.9(a)–4.9(c). In Figure 4.9(b), the training and validation loss had been improved due to data argumentation as compared to Figure 4.9(a). It also shows that data are trained well and loss function reduced as compared with unbalanced data. In addition, Figure 4.9(c) demonstrates that a smaller number of classes and images led to a reduction in

model loss functions. It is evident that the training process is a good fit, and the loss function clearly well with new data.

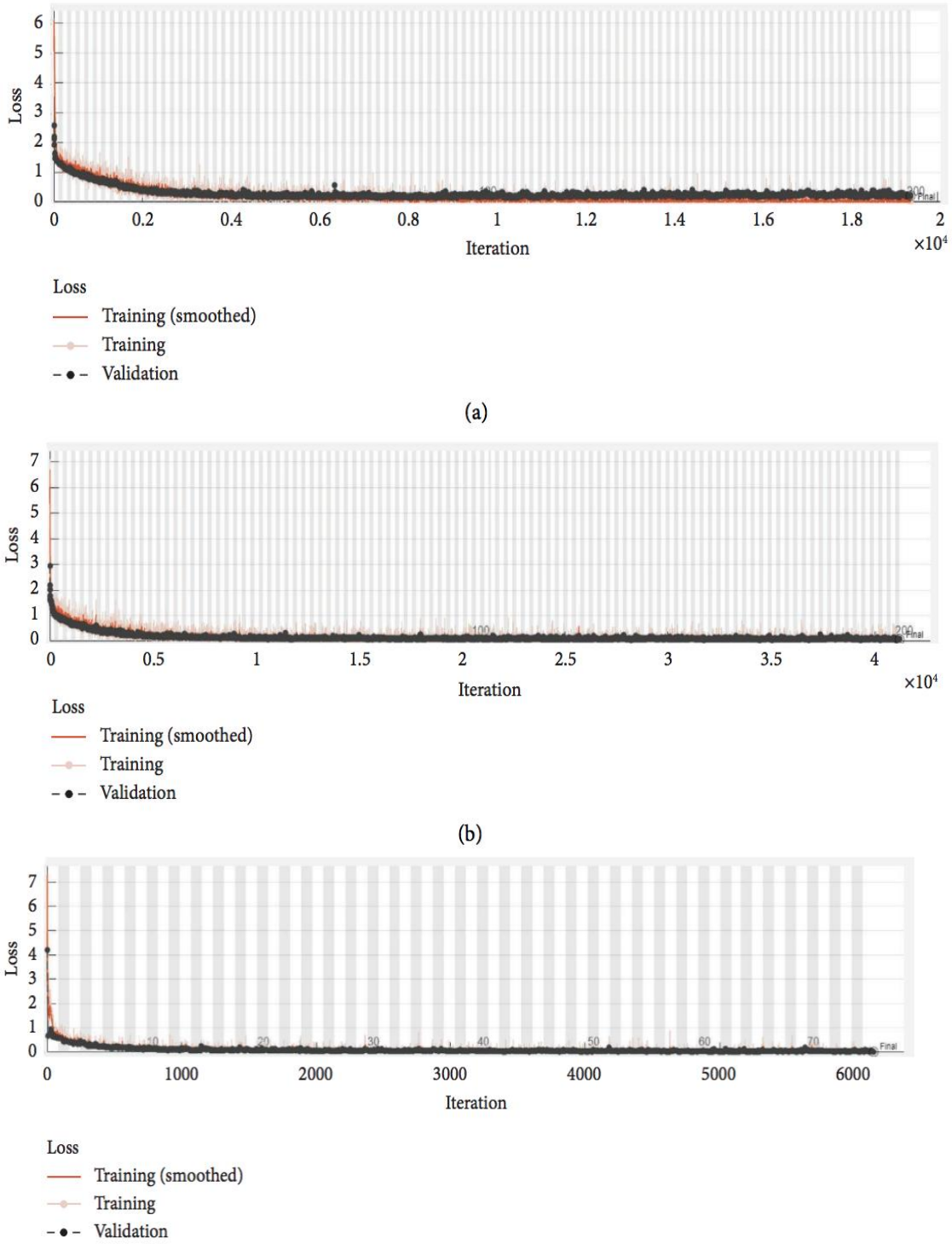


Figure 4. 9 Loss function of 1-CNN Model: (a) balanced datset-1, (b) unbalanced datsatet-1,(c) dataset-2

According to the experimental results, it is obvious that the proposed 1-CNN model functioned very well and it has lower error and loss function. Thus, the proposed 1-CNN model seems to be a suitable and robust deployed to become a practical application on mobile devices for cucumber leaf disease and pest diagnosis. Furthermore, the model demonstrates outstanding performance in terms of recognition outcomes, not only when applied to a large dataset containing numerous classes with a large number of images but also when applied on datasets with a few number of classes. Moreover, the Inception-V3, ResNet-50, and AlexNet models have pre-trained on extensive amounts of data.

Within this research study, the obtained results demonstrate that the proposed 1-CNN model achieved remarkable recognition accuracy with datasets characterized by a considerable number of samples and classes. Additionally, these findings highlight the model's versatility and its potential for real-world application, particularly when integrated with mobile devices.

4.6.3. Statistical Performance Analysis

The methods were presented in the past sections, and experiments were conducted for five different models based on deep learning (1-CNN, 2-CNN, Inception-V3, ResNet-50, and Alex Net) on two different datasets (Datasets-1 and 2). As explained in sections 4.6.1 and 4.6.2, the proposed model attained a superior accurate recognition result. To demonstrate the efficacy of the proposed 1-CNN algorithm performance, experiments on the models were conducted statistically using the chi-square test. It is a statistical test that is used to compare observed and expected results based on the formula, as defined in equation (4.4) (Sampath Kumar Gajawada, 2019).

$$X^2 = \sum \frac{(O_i - E_i)^2}{E_i} \qquad (4.4)$$

where O_i is observed value and E_i is expected value.

The predicted test samples based on the adopted models have been compared using the chi-square test. A standard threshold $\alpha = 0.05$ was used to show the significance differences of the proposed model compared with others models. Based on the results shown in Table 4.14, the values of the test statistics of 1-CNN with Inception-V3, ResNet-50, 2-CNN, and AlexNet were 4545.520, 4389.209, 4418.237, and 4273.575, respectively. The p-value of the proposed 1-CNN model was found to be less than the chosen significance level α = 0.05 when compared with all other models. We can state that there was a significant difference between 1-CNN with other models on unbalanced dataset-1 that included 973 test image samples.

Table 4. 14. All model statistical analysis on unbalanced dataset-1

Models	Test statistic value	p-value
1-CNN * Inception-V3	4545.520	0.001
1-CNN * Resnet	4389.209	0.003
1-CNN * 2-CNN	4418.237	0.0025
1-CNN * AlexNet	4273.575	0.004

In addition, 1660 test image samples were statistically experimented. From Table 4.15, we can see that the test statistical values were 8019.351, 7901.902, 7808.641, and 7722.563, for 1-CNN model with other models which are Inception-V3, ResNet-50, 2-CNN, and AlexNet, respectively. The results indicate that there was a significant difference between proposed 1-CNN and other models on balanced dataset-1. The p-value of the proposed 1-CNN model, when compared with all other models, was determined to be less than the chosen significance level α = 0.05.

Table 4. 15. All model statistical analysis on balanced dataset-1

Models	Test statistic value	p-value
1-CNN * Inception-V3	8019.351	0.001
1-CNN * Resnet	7901.902	0.003
1-CNN * 2-CNN	7808.641	0.005
1-CNN * AlexNet	7722.563	0.007

On the other hand, a statistical evaluation of all models on dataset-2, which consisted of 138 test image samples, was conducted. Based on the results shown in Table 4.16, it is indicated that there was a significant difference between 1-CNN and other models on dataset-2. The *p* value of 1-CNN with other models was less than the significance level $\alpha = 0.05$.

Table 4. 16. All model statistical analysis on dataset-2

Models	Test statistic value	p-value
1-CNN * Inception-V3	134.056	0.00053
1-CNN * Resnet	130.221	0.0036
1-CNN * 2-CNN	134.056	0.00053
1-CNN * AlexNet	122.557	0.017

The statistical analysis, as well as the evaluation of recognition outcomes, demonstrated that the proposed 1-CNN model exhibited superior performance in comparison to the other models. The statistical assessment, relying on calculated p-values, provided compelling evidence of the statistically significant differences in performance among the models. Furthermore, the assessment of recognition results, using various evaluation metrics, consistently preferred the proposed 1-CNN model. These results clearly

indicate the superior ability of the proposed model in accurately recognizing cucumber leaf diseases and pests. These findings highlight the robustness and effectiveness of the proposed 1-CNN model, emphasizing its potential for various applications in diagnosing cucumber diseases and healthy leaves.

4.7. Summary

In the domain of agriculture, diagnosing and classifying cucumber leaf diseases and pests is a critical task. In this chapter, an automated cucumber leaf disease and pest recognition system was proposed. Two different datasets, locally created dataset was used including five cucumber diseases and pests, i.e. spider, leaf miner, downy mildew, powdery mildew, CYSDV, and healthy leaf classes, and another publicly available dataset was used. Then they enlarged using data augmentation techniques to reduce overfitting. Quantitative experiments verified that the proposed 1-CNN model achieved superior recognition results. Additionally, comparison test results indicated that the proposed 1-CNN algorithm performed better than the 2-CNN, AlexNet, Inception-V3, and Resnet-50 models. The proposed 1-CNN algorithm yielded the best result as the number of sample images increased and the number of classes decreased.

CHAPTER FIVE

5. CUCUMBER LEAF DISEASE AND PEST DETECTION BASED ON LIGHTWEIGHT IMPROVED YOLOV5 MODEL

5.1. Introduction

An improved and proposed CNN model procedure from scratch for cucumber leaf disease and pest recognition system was presented in Chapter 4. The detection procedure of cucumber leaf disease and pest symptoms bases on deep learning models and research activities in this chapter is regarded as the second part within the overall research plan outlined in Chapter 3. This chapter focuses on addressing the problem of detecting symptoms in early stages, and detecting multiple disease symptoms on one cucumber leaf surface. The main objective of this chapter is to improve and develop YOLOv5 model based on deep CNN algorithm model for detecting cucumber leaf disease and pest symptoms.

The chapter starts with an overview of the procedures investigated in this part. It then proceeds to explain the dataset that was collected and prepared for the model by labeling image samples, followed by the improved YOLOv5 algorithm for detecting diseased cucumber leaf symptoms. The performance metric criteria are presented in relation to the accuracy of detection, as evaluated through the utilization of a confusion matrix. Furthermore, deep learning models in the YOLO family versions were applied for detection and comparing them with the improved yolov5 model. Finally, the chapter concludes with a summary of the investigations conducted.

5.2. Overview of the Improved Algorithm

The investigation in this chapter can be divided into three main parts, namely data preparation, data labelling and finally improving the network model. Figure 5.1 illustrates an overview of the experimental detecting procedure for cucumber leaf disease and pest. The first objective of this chapter is to improve YOLOv5 model-based deep learning algorithms suitable for the accurate

detection of small cucumber leaf disease and pest symptoms under natural light conditions. The system takes an image as input, which includes symptoms of cucumber leaf diseases and pests, based on a constructed leaf image dataset. Another objective in this chapter is, the improved YOLOv5 model from scratch with YOLOv5 family pre-trained models such as YOLOv5s, YOLOv5m, YOLOv5l, YOLOv5n, and YOLOv5x have been examined. The purpose of the developed system can help farmers detect the symptoms of these issues in timely stage before they cause significant damage to crops. This can help farmers take timely action and reduce their losses.

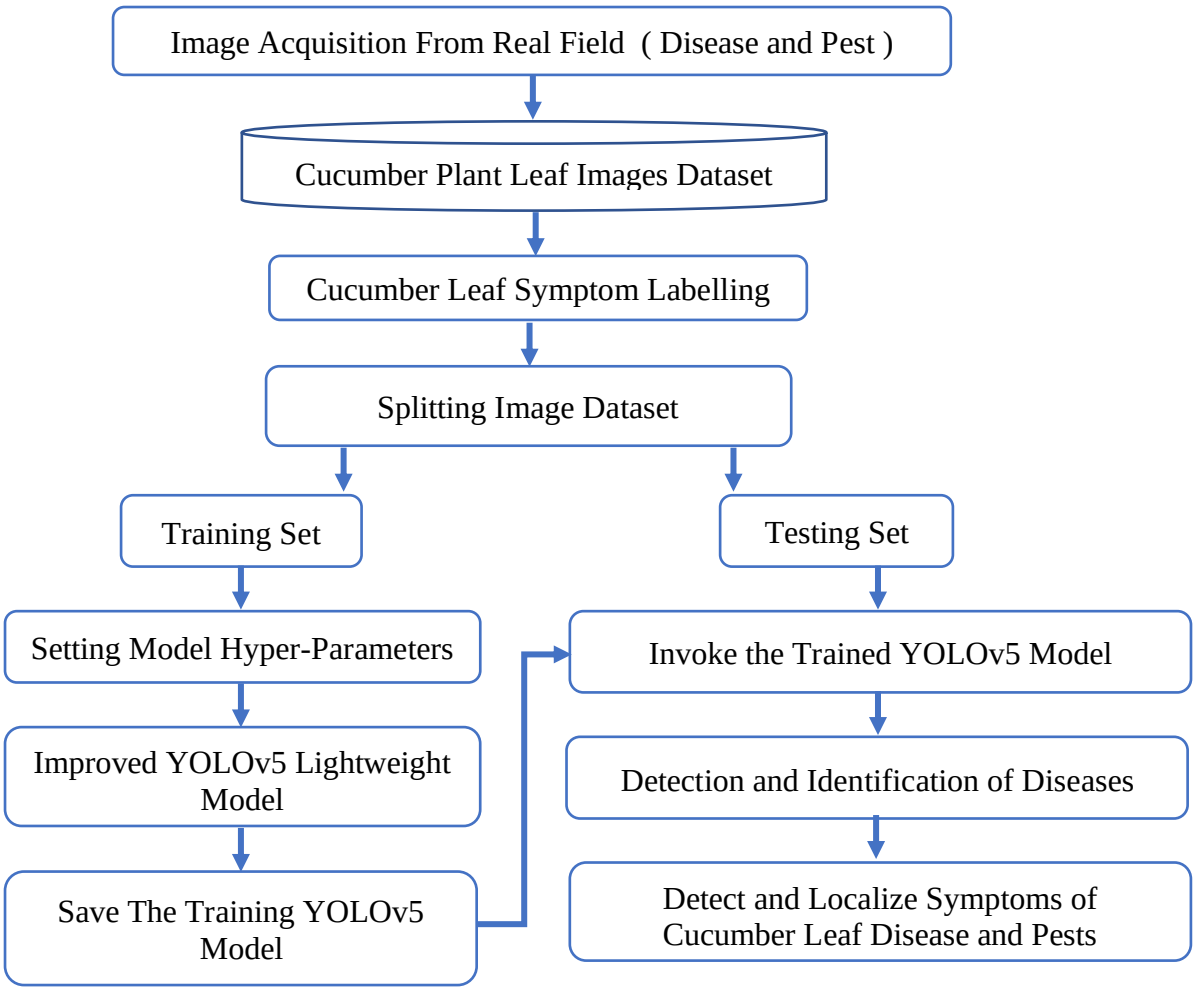


Figure 5. 1 An overview the procedure of detecting cucumber leaf diseases and pests

5.3. Data Preparation

In this chapter, the locally created dataset was used to conduct the model experiments. The dataset used in this chapter includes four various cucumber leaf diseases and pests. Figure 5.2 shows sample images of the dataset. It contains four cucumber disease classes including two pest diseases (spider and leaf miner), two fungal diseases (downy mildew and powdery mildew). Total images of the dataset are 3057 images, each class having a sample image number in the range of 313–1379**.** Class diseases names and sample numbers were described in Table 5.1.

Spider **Leaf Miner** **Powdery Mildew** **Downy Mildew**

Figure 5. 2 Samples of Images of the dataset

Images of cucumber leaf diseases and pests were involving single and multi-disease infections. However, cucumber leaf symptom diseases in the images were not labeled. The region symptoms in each image in the dataset should be labeled and annotated in order to be fit into the model as an input.

Table 5. 1 Dataset class image disease name and number

Class No.	Class Name	Original dataset	Training images	Testing images
1	Spider	555	455	100
2	Leaf Miner	810	709	101
3	Downy mildew	313	273	40
4	Powdery mildew	1379	1199	180
	Total	3057	2636	421

5.4. Data Pre-Processing (Data Labelling)

In this study, we prepared a dataset of 3057 images for cucumber leaf disease and pests to train YOLOv5 model in our experiments. To make the images suitable as input for the model, it's necessary to mark and annotate the symptoms in the regions of cucumber leaf diseases and pests in each image in the dataset. A labelling tool named (Image Labeler software) was used. We labeled and annotated the regions of interest for cucumber leaf diseases and pest symptoms with rectangular boxes in each image in the dataset using a custom algorithm, which was explained in detail in Section 3.5. Images which were used from the collected dataset for the training and validation sets. The names and numbers of images in each class are shown in Table 5.1. Furthermore, all image size dimensions are fixed at a resolution of 640 × 640. This step aimed to minimize computational complexity and enhance the model's efficiency. The labeling test image procedure resulted in a number of instances, which are presented in Table 5.2. Some test images included more than one symptom of the same disease or pest type, while other images included symptoms of different diseases or pests.

Table 5. 2. The labelling test image procedure resulted in a number of instances

Class name	Test Image Sample Number	Test Instances
Downy Mildew	40	497
Leaf Miner	101	452
Spider	100	191
Powdery Mildew	180	1765
Total	421	2905

5.5. Cucumber leaf Disease and Pest Detection Model

With the rapid development of the agricultural sector, particularly in leaf disease detection, numerous research studies and implementations have employed deep learning techniques to identify leaf diseases and pests. Inspired by the recent advancements in deep learning models, our research proposed an approach for detecting cucumber leaf diseases and pests by improving the YOLOv5 model. Among the comprehensive algorithms, the YOLOv5 algorithm stands out as the most notable within the YOLO series. In this study, the YOLOv5l is improved as the base algorithm. Consequently, addressing the issues of precisely detecting leaf symptom regions and making improvements to the original YOLOv5 algorithm are considered of utmost importance. Therefore, YOLOv5l has been improved and enhanced in order to detect small cucumber leaf diseases and pest symptoms accurately. The enhancements made to the YOLOv5l algorithm seek to enhance its capability to accurately detect and classify different types of diseases and pests affecting cucumber leaves. The vital goal is to provide a robust and effective solution that contributes to improved agricultural practices and facilitates timely intervention for disease management in cucumber crop leaves.

5.5.1. Improved YOLOv5 Network Structure

The cucumber leaf disease spots and pest symptoms are small objects compared with the whole plant leaf images, especially in cucumber plant diseases; therefore, the standard YOLOv5 model still has an issue in generalization and domain adaptation, despite its widespread popularity and recognized efficiency (Li, Ahmed, *et al.*, 2022). In this circumstance, to establish a lightweight deep learning model and the most intuitive purpose to fit the embedded system, we improved the YOLOv5 model to focus on adapting the network to specific cucumber leaf disease and pest detection tasks. The improved YOLOv5 model architecture includes four main parts: input, backbone, neck, and head. In this study, the YOLOv5l model was developed based on the model size that was used with a large number of data samples.

Our proposed model begins by inputting the image into the network, with the image size set to 640 × 640 × 3. Subsequently, a series of middle layers is employed. These include focus layer, convolutional layers with batch normalization and the SiLU activation function, along with up-sampling layers, concatenation layers, and finally, detection layers. The convolutional layers play a crucial role in feature mapping and extraction, utilizing filters to perform convolutions that capture diverse levels of detail from the input image. In the backbone section, filter sizes of (64, 128, 148, 224, and 256) are employed with 3x3 window size, while in the neck section, the filter sizes are specified as (512, 256, 128, and 128). The filter size determines the dimensions of the regions in which neurons establish connections with the input, influencing the receptive field and information integration within the network. The improved YOLOv5 model architecture is shown in Figure 5.3.

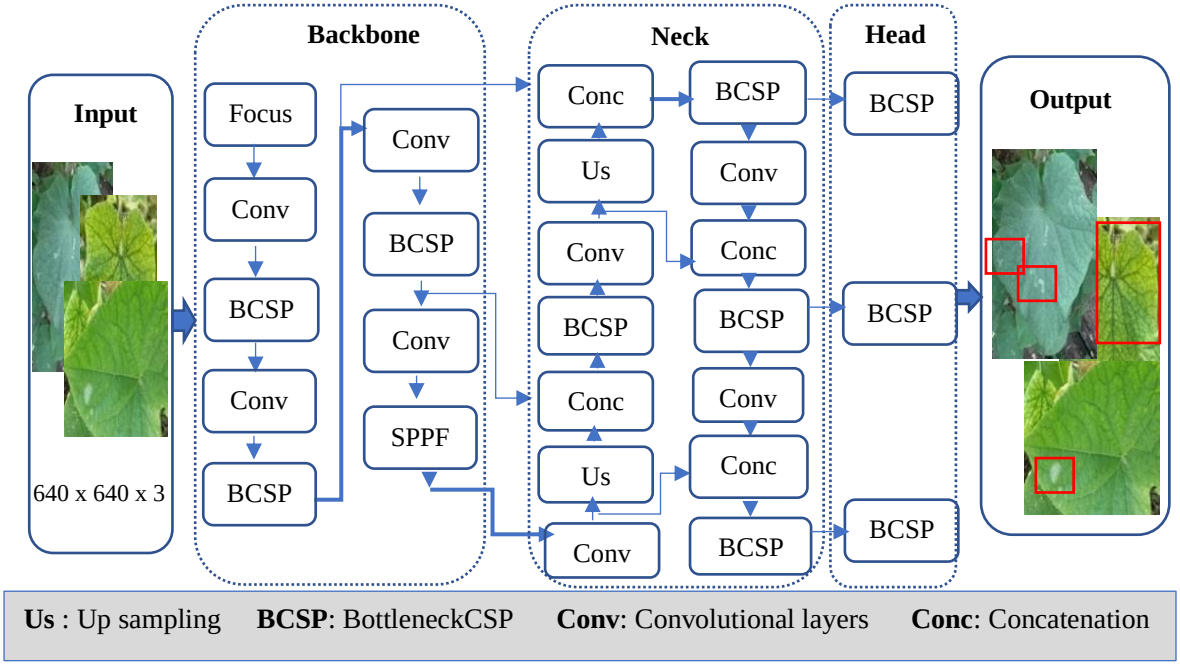


Figure 5. 3 An improved YOLOv5l model structure

Furthermore, several modifications were implemented on the original YOLOv5 model to address the aforementioned concerns and challenges effectively. The major modifications between the standard and the improved YOLOv5 are particularly applied in the backbone section. Firstly, the first convolutional layer was replaced with a focus layer, which involves partitioning the input image to acquire a downscaled feature map that contains twice the amount of information, to reduce the parameters, layers, CUDA memory, and FLOPS, as shown in Figure 5.4.

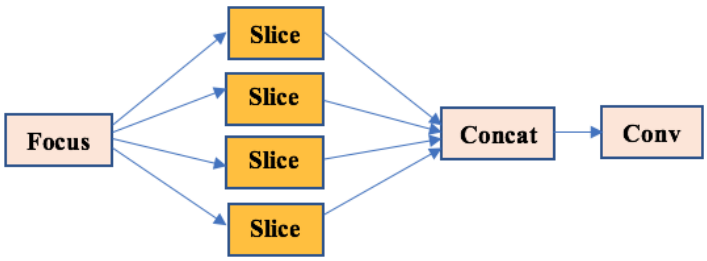


Figure 5. 4 Focus layer architecture (He and Wei, 2023)

Secondly, the Bottleneck cross-stage partial (CSP) is added and used instead of C3 module to enhance the representation of features. BottleneckCSP was used with different filter, kernel, and stride sizes, which improved the accuracy result. Additionally, the utilization of BottleneckCSP serves to tackle the concern of excessive computational requirements during inference, with a focus on optimizing the network structure design. Moreover, the incorporation of BottleneckCSP significantly enhances the network's capacity for effective learning. The BottleneckCSP consists of four convolutional modules, batch normalization, and Bottleneck. The Bottleneck is a residual block known for its accelerated computational speed. Additionally, it allows for the creation of deeper network architectures while minimizing computational parameters (Zhou *et al.*, 2023). The modifications were shown in Figure 5.5. Finally, the last C3 module layer was removed with the objective of decreasing the count of layers and parameters and enhancing the capability of feature extraction.

Neck section of YOLOv5, comprises supplementary convolutional layers that are responsible for integrating and combining features extracted from different spatial scales or feature maps. The model's capacity to effectively handle objects with diverse sizes and aspect ratios is also enhanced. The entire feature hierarchy with precise localization signals in the lower layers through bottom-up path augmentation were improved, which shortens the information path between the lower layers and the topmost feature. In addition, YOLOv5 uses a path aggregation network (PANet) to improve information flow. In the neck section, all C3 modules were replaced with the BottleneckCSP module. These modifications are expected to enhance the models ability and efficacy in accurately detecting cucumber leaf disease and pest symptoms region.

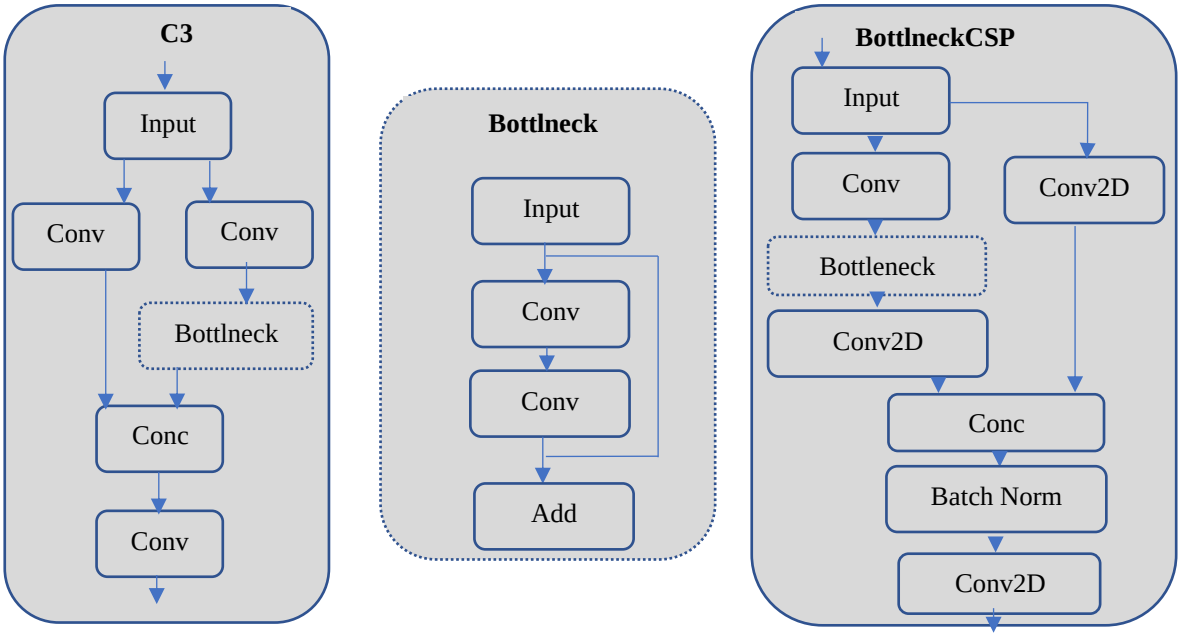


Figure 5. 5. Comparison of YOLOv5 model (Left figure is the C3 layer, Right figure is the way C3 is replaced by BottleneckCSP)

5.5.2. Hyper-parameter setting

The improved YOLOv5 model was trained using a python. During training and processing, different model hyperparameters, such as number of layers, filter numbers and sizes, were carefully adjusted to achieve the best detection accuracy outcome. Furthermore, three other factors to consider were the size of the input shape, the epoch number, and the size of the batch.

In this study, the training dataset images were resized into three different sizes: 380, 640, and 800. Among these sizes, the highest results were achieved when the images were resized to 640×640×3. Initially, the model was trained for 100 epochs. Subsequently, the number of epochs was fine-tuned and tested with values of 126, 150, 172, 185, and 210, ultimately determining that the best results were obtained with 100 epochs. To regulate the model's response to error, a learning rate parameter of 0.01 was employed during the training process. Moreover, the optimal batch size for training was determined to be 8,

while batch sizes of 16 and 32 were also tested. The specific training hyper-parameters for YOLOv5 can be seen in Table 5.3. Following improvements made to YOLOv5l, the resulting model comprised a total of 6,613,189 parameters and 214 layers.

Table 5. 3 Improved YOLOv5l model hyper-parameter settings

Parameters	Value
Image size	640 x 640 x 3
Batch size	8
Number of Epochs	100
Learning rate	0.01
Optimizer	SGD

5.5.3. Incorporating the Attention Mechanism

Effective feature extraction is essential for cucumber leaf disease and pest symptom detection using deep learning. However, this task becomes challenging when using common YOLOv5 models due to the small size of spot symptoms relative to the entire image. To address this issue and enhance the significance of spot symptom diseases, we introduce the convolutional block attention module (CBAM) into the backbone section of both the improved and original YOLOv5l models. CBAM combines channel and spatial attention, as illustrated in Figure 5.6, where channel attention emphasizes important features and suppress less relevant ones. Spatial attention focused on modelling the interconnections among various spatial locations within a feature map. This enables the model to concentrate on significant regions while suppressing irrelevant or less informative areas (Niu et al., 2021). The integration of these

attention mechanisms results in a superior feature representation. CBAM model is introduced after the convolutional layer of the backbone section of the YOLOv5l network model.

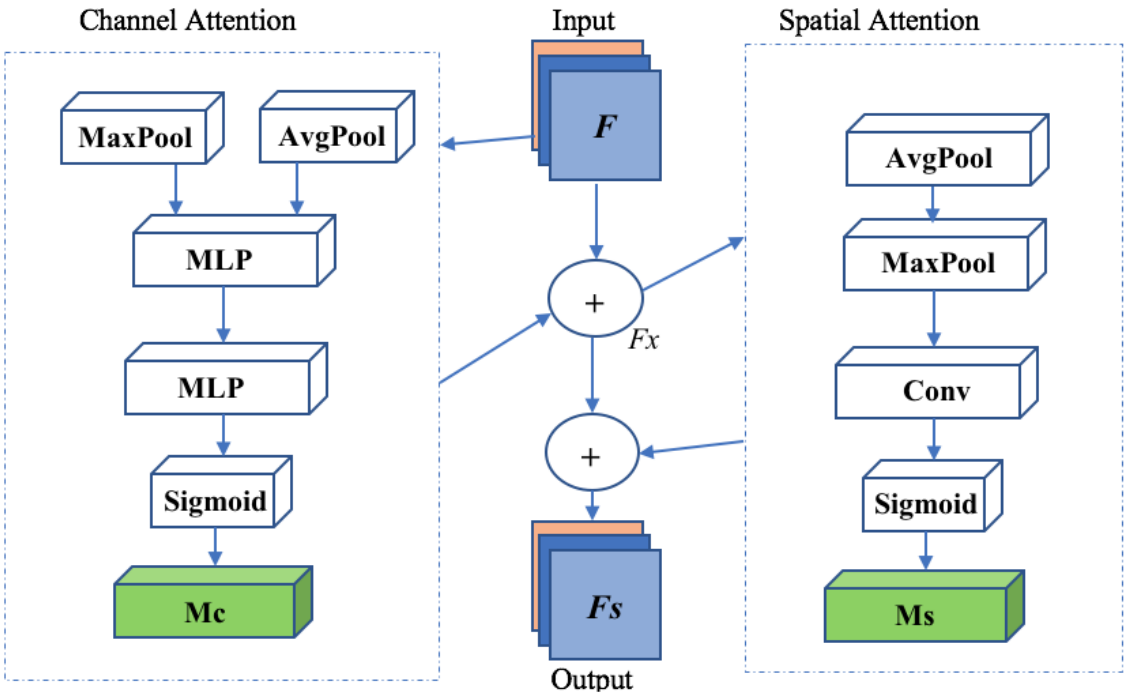


Figure 5. 6 Structure of CBAM (He and Wei, 2023)

In Figure 5.6, The CBAM incorporates both channel attention (Mc) and spatial attention (Ms). For a given feature map, $F \in R^{CxHxW}$, where C denotes the channel count and HxW signifies the feature map's dimensions, the CBAM module first processes F through the channel attention module. Simultaneously, it uses average and max pooling methods to gather information about each channel. These obtained parameters are then combined using a multilayer perceptron (MLP) and activated using the Sigmoid function, resulting in channel attention characteristics. Next, the output from Mc is fed into the spatial attention module, where channel information is gathered again

using average and max pooling, followed by a convolutional layer. The obtained parameters are then activated through the Sigmoid function, yielding spatial attention features.

In this study, the CBAM is incorporated into the original YOLOv5l network backbone section. The performance of cucumber leaf disease and pest symptom detection is improved by adding CBAM models after the convolutional layer of the backbone section of the YOLOv5l network model via enhancing feature extraction. Two CBAM blocks have been inserted after the last two convolutional layers in the backbone network section. The structure of the original YOLOv5l network with the added CBAM module is illustrated in Figure 5.7. The images were resized to 640×640×3; there was 100 epochs in the training procedure. During the training procedure, a learning rate value of 0.01 was used. Moreover, the optimal batch size for training was determined to be 8.

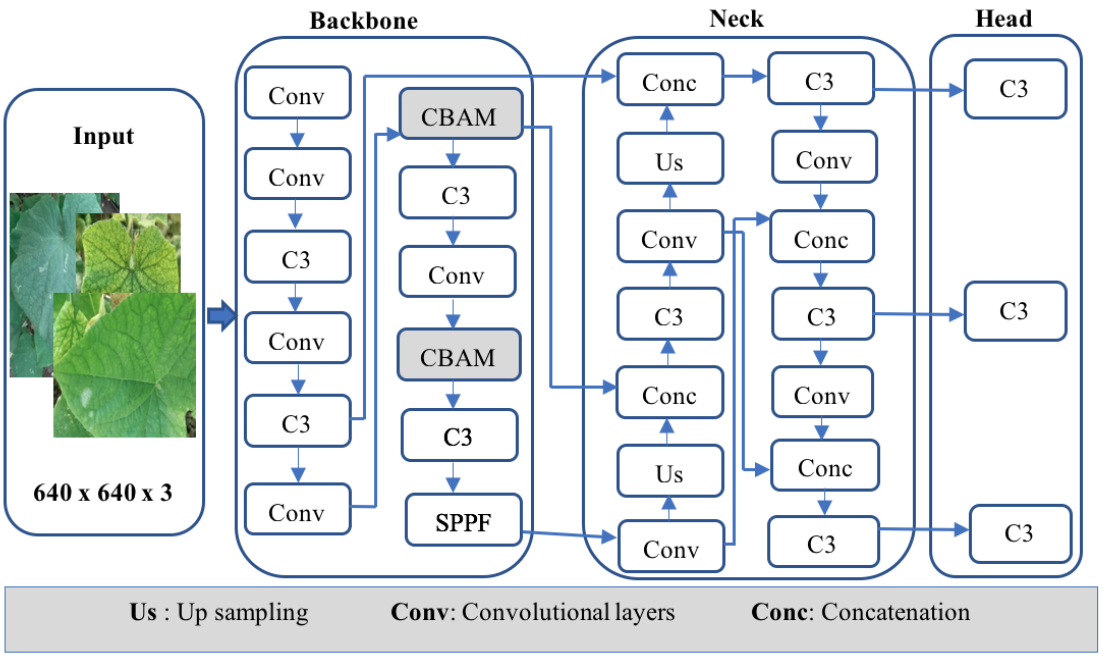


Figure 5. 7 Network architecture of the original YOLOv5l with incorporation of CBAM in the backbone section

Furthermore, the CBAM is incorporated into the improved YOLOv5l model. CBAM block has been added following all the convolutional layers in the backbone network section. The structure of the improved YOLOv5l network with the added CBAM module is depicted in Figure 5.8. The images were resized to 640×640×3, and the model was trained for 100 epochs. During the training process, a learning rate value of 0.01 was used. Moreover, the optimal batch size for training was determined to be 8.

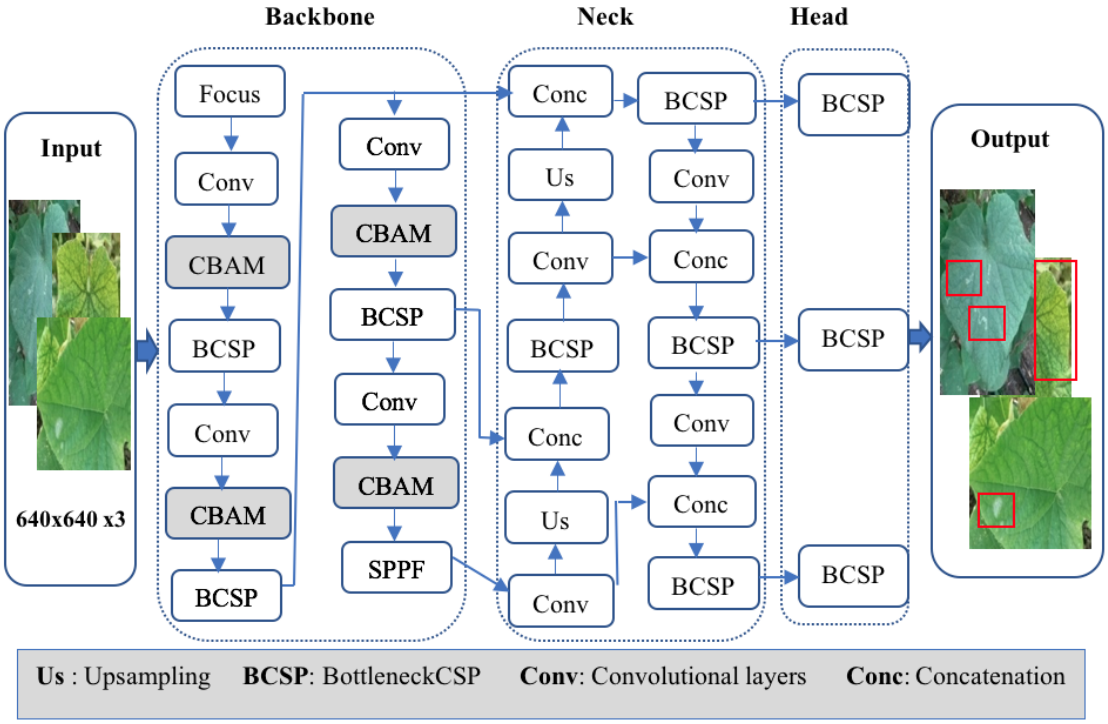


Figure 5. 8 Network architecture of the improved YOLOv5l with the incorporation of CBAM in the backbone section

5.6. Evaluation Indicators (Evaluation Model Metrics)

In this study, the improved YOLOv5 model's detection capability has been evaluated using three different metrics; the model applicability in real scenarios

was evaluated using precision, recall, and mAP @0.5. The calculation formulas of the recall, precision and mAP were defined and explained in Section 3.7.

5.7. Results and Discussion

This section provides a comprehensive analysis and interpretation of the results that have been empirically obtained from conducting different experiments on the improved YOLOv5 from scratch. This involves evaluating the accuracy and efficiency of these algorithms on a dataset of annotated images and their performance is compared to that of other state-of-the-art models. This study improves a lightweight YOLOv5l algorithm from scratch based on deep learning methods to precisely detect cucumber leaf diseases and pest spot symptoms. The detection system is done to identify and localize disease spots at timely stages and to minimize time consuming, hence reducing the network layers and increasing the accuracy rate. The performance of the detection system was measured based on percentage of correctly identified and localized disease and pest spots.

5.7.1. Cucumber Leaf Disease and Pest Detection Analysis

To assess the effectiveness of the proposed YOLOv5 models in accurately detecting cucumber leaf pests and disease symptoms, a series of experiments were performed. These experiments focused on various factors including the input image size, number of layers, parameters of the convolutional layers such as filter size and window size, activation function, batch size, and number of epochs. The study also involved in a comparative analysis of all YOLOv5 model versions with the improved YOLOv5l network specially designed for cucumber leaf disease and pest detection. The evaluation of model performance

and effectiveness involved the use of metrics including recall, precision, and mAP, which were calculated based on the confusion matrix.

The choice of selecting YOLOv5l as the base model for improvement and testing purposes was based on the correlation between the model size and the size of the constructed dataset. The improved YOLOv5l model was trained for 100 epochs using a dataset consisting of 3057 images for training, and validation purposes. The improved YOLOv5l network was trained from scratch specifically for this test. The outcomes indicated that the enhanced model outperformed the other YOLOv5 models, achieving an mAP@0.5 accuracy of 80.10%. Precision and recall, based on the confusion matrix, yielded values of 73.80% and 73.90%, respectively. The improved YOLOv5l model precision, recall, and mAP@0.5 metrics are provided which were presented in Table 5.4.

Table 5. 4 Precision, recall, and mAP results of the improved YOLOv5l

	Precision (%)	Recall (%)	mAP@0.5 (%)
All Classes	73.8	73.9	80.1

Furthermore, Figure 5.9 provides the P-R (precision-recall) curves obtained from the conducted experimental results. The algorithm's performance is considered superior when the P-R curve approaches the coordinate position (1, 1). The precision-recall curve (PRC) is a visual representation that illustrates the relationship between precision (also called positive predictive value) and recall (also known as sensitivity or true positive rate). The PRC is graphically displayed, with the x-axis representing recall and the y-axis representing precision.

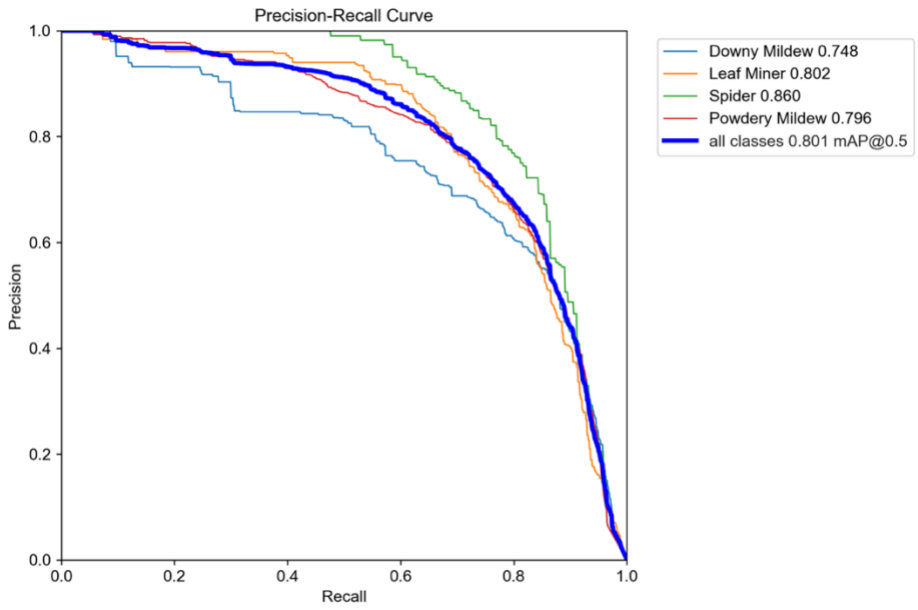


Figure 5. 9. PR curve: With the recall rate on the horizontal axis and the precision rate on the vertical axis

5.7.2. Comparison with Original YOLOv5l Algorithm

In order to indicate the superior performance of the improved algorithm, this study conducts a comparative analysis with the original YOLOv5l model in the context of detecting cucumber leaf disease and pest symptoms. The evaluation metrics employed in this comparison include precision, recall, and mAP@0.5. The improved YOLOv5l model and the original YOLOv5l version models were training for 100 epochs using a constructing dataset consisting of 3057 images for training, validation, and testing purposes; a test set of 421 randomly selected images were used in this dataset.

In this study, the improved YOLOv5l model trained from scratch was compared with the original YOLOv5l model, based on the best mAP@0.5 result. However, the original YOLOv5l model utilizes its own pre-trained weights obtained from the common object in context (COCO) dataset (Jung and Choi, 2022). The improved YOLOv5l model demonstrated superior

performance with an mAP@0.5 of 80.10% as compared to the original YOLOv5l model's mAP@0.5 of 79.00%. Experimental results presented that the improved model outperformed the original YOLOv5l model by 1.1% achieving with mAP@0.5. Comparative results for disease and pest detection are presented in Figure 5.10, where precision, recall, and mAP metrics are provided.

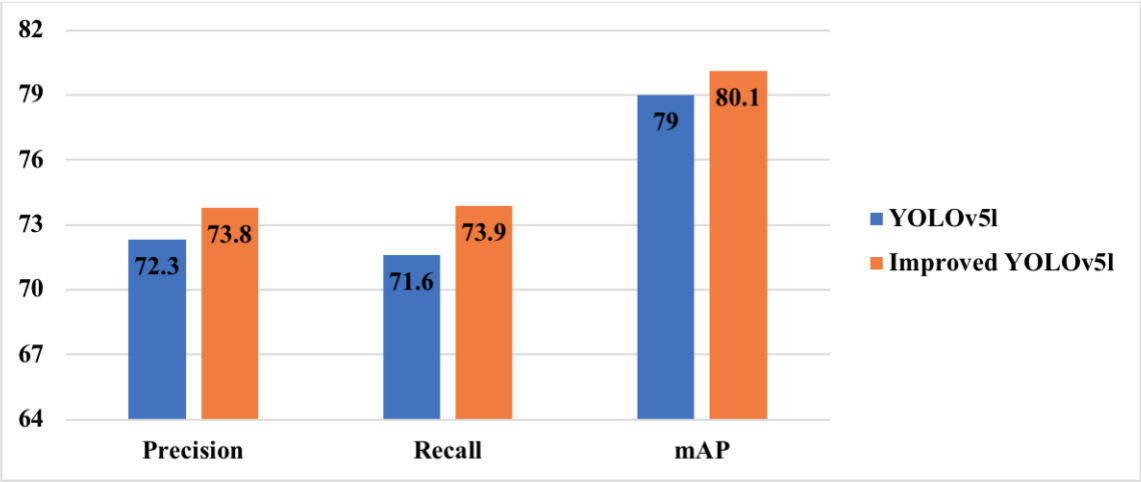


Figure 5. 10 Comparison of result values for original and improved YOLOv5l models

Based on the findings presented in Figure 5.10, it can be concluded that the improved YOLOv5l model outperformed the original YOLOv5l model exhibiting superior performance in terms of precision, recall, and mAP. Notably, the improved model was trained from scratch without utilizing pre-trained model weights, and it demonstrated the best performance compared to the weights obtained from the trained YOLOv5 model. As shown in Figure 5.10, the precision and recall for the improved YOLOv5l model increased by 1.5%, and 2.3%, respectively. In addition to that, the improved YOLOv5l was selected due to its optimal balance between accuracy and speed, along with its

ability to effectively detect small leaf disease spot symptoms, which were abundant in the dataset.

The training process of the improved YOLOv5l is shown in Figure 5.11. Based on metric curves, the results of the graphs indicate that the improved model demonstrates superior accuracy in detecting cucumber leaf diseases and pest symptoms compared to other models. Figure 5.11 displays the metric curves representing the progression of training and validation sets. The first three columns of the figure illustrate the loss components of the improved YOLOv5l model, specifically the box, object, and classification losses during the training processes. The second row's first three columns illustrate the validation process. Throughout the training, the experimental results indicate accurate identification of the four classes used for detection. Considering the change in the threshold value for the confidence level, the performance of the object detector was evaluated based on the PRC method. The confidence level refers to a value that indicates the user’s confidence in the algorithm’s detection. As a comparison, Figure 5.12 illustrates the original YOLOv5l training and validation process.

Furthermore, to the aforementioned points, the evaluation of the improved YOLOv5l model's detection capabilities necessitates conducting performance analysis in scenarios where small objects are present. This analysis takes into account the different sizes of diseases and pest symptoms that may occur during their various stages. Given the substantial size variations demonstrated by these objects, accurate detection and identification in small disease and pest symptom object scenarios becomes particularly important. This evaluation is essential for ensuring the model's reliability and applicability in a variety of real-world scenarios involving diseases and pest symptoms of different sizes.

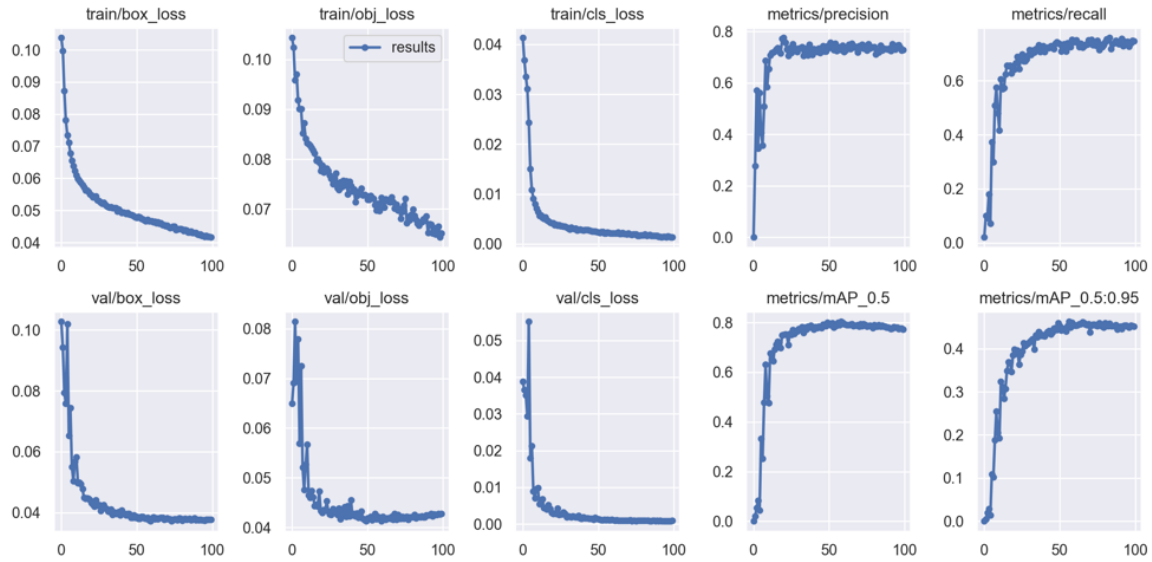


Figure 5. 11 Convergence the loss functions of training and validation sets for improved YOLOv5l

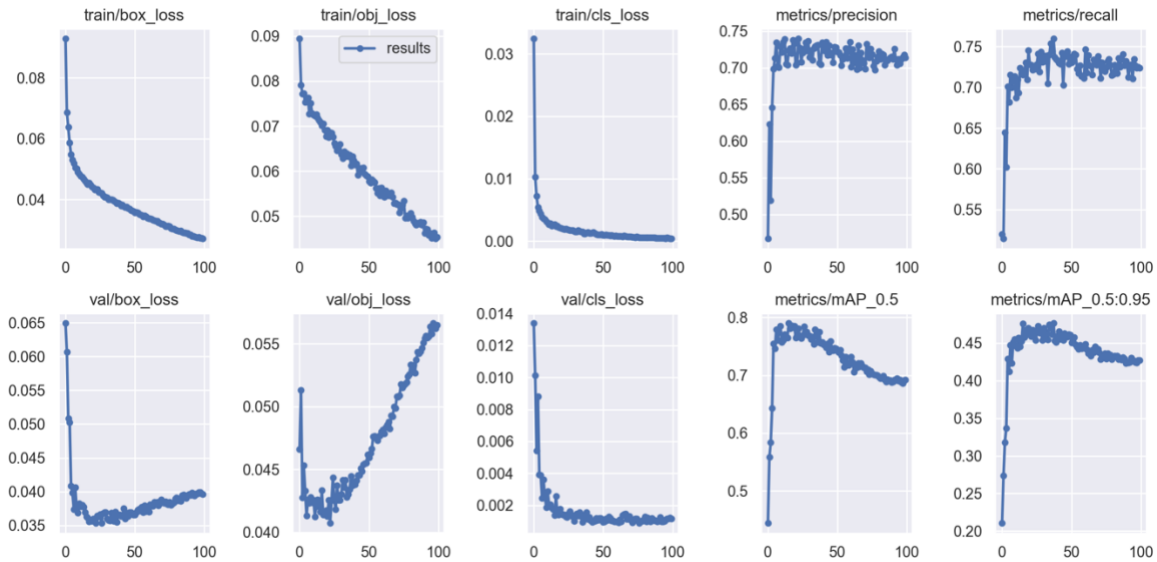


Figure 5. 12 Convergence the loss functions of training and validation sets for original YOLOv5l

Based on the findings presented in Table 5.5, it is concluded that the improved YOLOv5l model exhibits superior performance in detecting small disease spot instances and larger disease spot instances. The evaluation of each

class's performance highlights notable results, particularly in the spider class. Precisely, the spider detection achieved a precision of 83.4%, recall of 73.9%, and mAP@0.5 of 86.0%. In contrast, the leaf miner class presents even better performance with a recall of 79.2%. This study has significantly improved the detection of small pests and disease symptoms. Regarding other disease classes, such as powdery mildew and downy mildew, the mAP@0.5 values were 79.6% and 74.8% respectively. As the data presented in Table 5.5, the improved YOLOv5l model demonstrated superior results, especially in the spider class when compared to the other classes.

Table 5. 5 Key indicator values of the improved YOLOv5l

Parameter	Downy Mildew	Leaf Miner	Spider	Powdery Mildew	Total
Precision (%)	64.30	66.30	83.4	81.2	73.8
Recall (%)	76.7	79.2	73.9	65.7	73.9
mAP@.5 (%)	74.8	80.2	86	79.6	80.1

Furthermore, in this study, we present the robustness and detection effectiveness of the improved YOLOv5l model. We considered variables including the count of parameters, layers, and the duration of the training process. These aspects are illustrated in Table 5.6.

Table 5. 6. Experimental output parameters of improved YOLOv5l and YOLOv5l models

Models	No. Layers	Parameters	Weight Size	Training Time/ Hours
Improved YOLOv5l	**214**	**6613189**	**13.6 MB**	**2.58**
Yolov5l	267	46124433	92.8 MB	4.41

Based on the values presented in Table 5.6, we confirm that the improved model exhibits superior characteristics in terms of the training process time, number of layers, number of parameters, and weight sizes. Specifically, the improved model achieved a training process time of 2.58 hours, which demonstrates a faster training procedure, with a total of 214 layers. Furthermore, the number of parameters associated with the improved model is 6613189, indicating a more optimized and efficient design. In terms of weight sizes, the improved model recorded 13.6 MB, reflecting efficient utilization of storage resources. From Table 5.6, we provide evidence that the improved model outperformed the original YOLOv5l model in terms of space complexity and training time consumption. The improved model's faster training time allows for expedited model development and deployment, saving valuable computational resources. Additionally, the reduced space complexity of the improved model optimizes storage utilization, resulting in more efficient and scalable implementation. These findings demonstrate the significant advancements attained through the improvement made to the YOLOv5l model.

The multiscale training approach suggested in this research can improve the robustness of the model for the detection of images of various resolutions. This research separates the input image size into different distinct resolution sizes 380, 640, and 800, to examine the detection effect of the model on the input images of various resolutions. The best performance was achieved at a medium resolution of 640.

In addition, to evaluate how effectively the improved model performs, external test images were used for experimentation, as illustrated in Figure 5.13. As can be observed from Figure 5.13, the improved YOLOv5l can effectively address challenges related to false detection and missing detection issues. This capability remains reliable even in scenarios involving with a multiple of disease symptoms and diverse small-sized disease and pest spots within the image. Therefore, based on the experimental results, the improved

YOLOv5l model proves to be a capable solution for effectively accomplishing the detection task of cucumber leaf diseases and pests. Particularly, this deduction is derived by considering various factors such as detection accuracy, detection speed, minimal parameter number, the presence of small symptom disease spots, and the overall weight sizes.

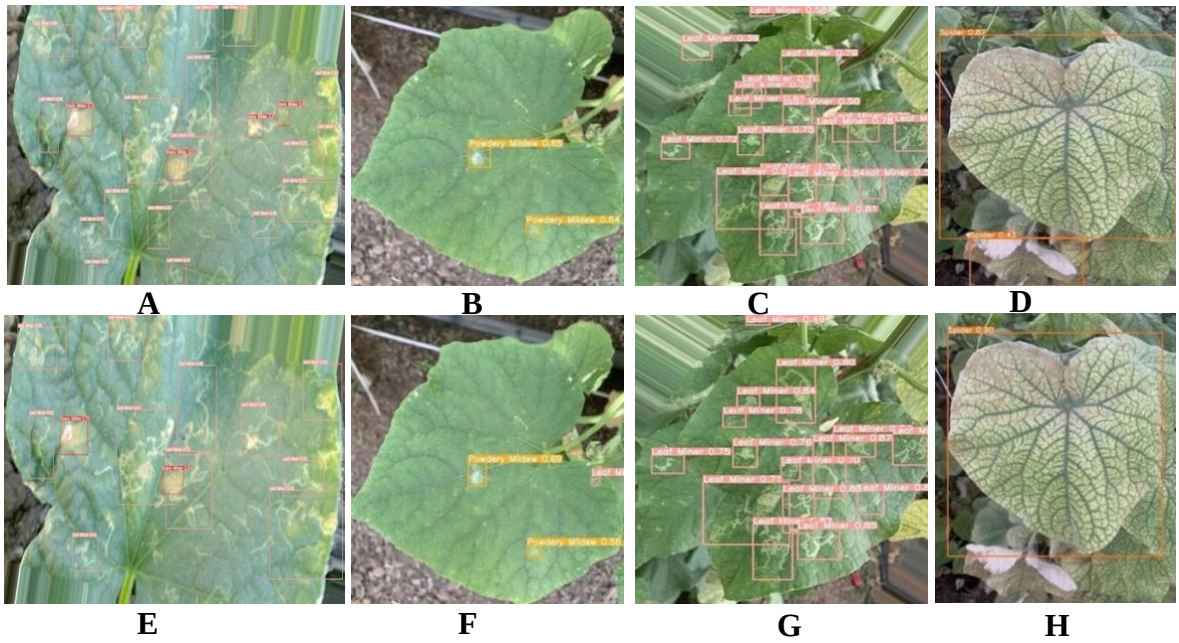


Figure 5. 13 Comparison of the detection effects of all classes. (a–d) Improved YOLOv5l ; (e–h) Original YOLOv5l.

5.7.3. Ablation Experiments

In this study, the algorithms conducted improvements on both the improved YOLOv5l and original YOLOv5l detection framework. Additionally, the CBAM was introduced into both the improved and original models. To verify the effectiveness of the presented methods for cucumber leaf disease and pest symptom detection, ablation experiments were conducted using two groups of

networks based on YOLOv5. The results of the evaluation and comparison of model improvement strategies were shown in Table 5.7.

Table 5. 7 Results of ablation experiments

Models	Evaluation Metric (%)			No. of Layer	Training Time / Hours	Weight size (MB)
	Precision	Recall	mAP@0.5			
YOLOv5l	72.3	71.6	79	267	4.41	92.8
YOLOv5l + CBAM	75.1	73.8	79.8	289	4.01	94.5
Improved YOLOv5l	**73.8**	**73.9**	**80.1**	**214**	**2.58**	**13.6**
Improved YOLOv5l + CBAM	**74.9**	**73.3**	**80.2**	**258**	**2.73**	**13.8**

The findings from Table 5.7 demonstrate that each model contributes to enhancing the model's overall performance to different degrees, as evaluated by precision, recall, and mAP@0.5 metrics. The initial detection outcome for the original YOLOv5l model is presented in the table's first row, achieving 79% mAP@0.5. When the CBAM layer is added into the backbone section of the original YOLOv5 after the last two convolutional layers, the detection results for precision, recall, and mAP@0.5 show an increase of 2.8%, 2.2%, and 0.8% respectively, as compared to the original YOLOv5 network. On the other hand, the detection result accuracy for precision, recall, and mAP@0.5 of the improved YOLOv5l increased by 1.5%, 2.3%, and 1.1% respectively, as compared to the original YOLOv5l network. Furthermore, when the CBAM layer is added into the backbone section of the improved YOLOv5 after all the convolutional layers, the detection results for precision, recall, and mAP@0.5 experienced an increase of 2.6%, 1.7%, and 1.2% respectively, as compared to the original YOLOv5 network.

Through the comparison of ablation experiments, it was observed that the performance improvement is achieved by the improved model. The improved YOLOv5l model demonstrates efficient detection capabilities for cucumber leaf diseases and pests while considering factors such as detection speed, a smaller number of parameters, fewer layers, and manageable weight sizes.

5.7.4. Comparison with All YOLOv5 Family Version Models

In this study, the YOLOv5 models including YOLOv5n, YOLOv5s, YOLOv5m, YOLOv5l, YOLOv5x, and the improved YOLOv5l from scratch were carried out to detect disease symptoms in cucumber leaves. Based on the confusion matrix, various calculations were performed, including recall, precision, and mAP@0.5. The performance of all the standard YOLOv5 family model versions was compared to accurately verify the effectiveness of the improved YOLOv5l model. Consequently, we conducted a series of experiments to test the evaluation of YOLOv5n, YOLOv5s, YOLOv5m, YOLOv5l, and YOLOv5x models using the constructed dataset. Each experiment was performed independently, and the precision, recall, and mAP@0.5 values were compared. The resulting data were summarized and presented in Table 5.9 and Figure 5.13 for ease of comparison. Upon analyzing the final values, our proposed model exhibited the best performance.

The findings resulting from the analysis of the experiments illustrated in Table 5.8 and Figure 5.14 provide clear evidence that the improved YOLOv5l model outperformed the other models and demonstrated a higher level of accuracy in the detection of small cucumber leaf disease and pest symptom. The improved YOLOv5l model demonstrated a superior performance in accurately identifying and localizing symptoms within a given image scene. Thus, founds its prominence as a robust and efficient solution for object detection tasks.

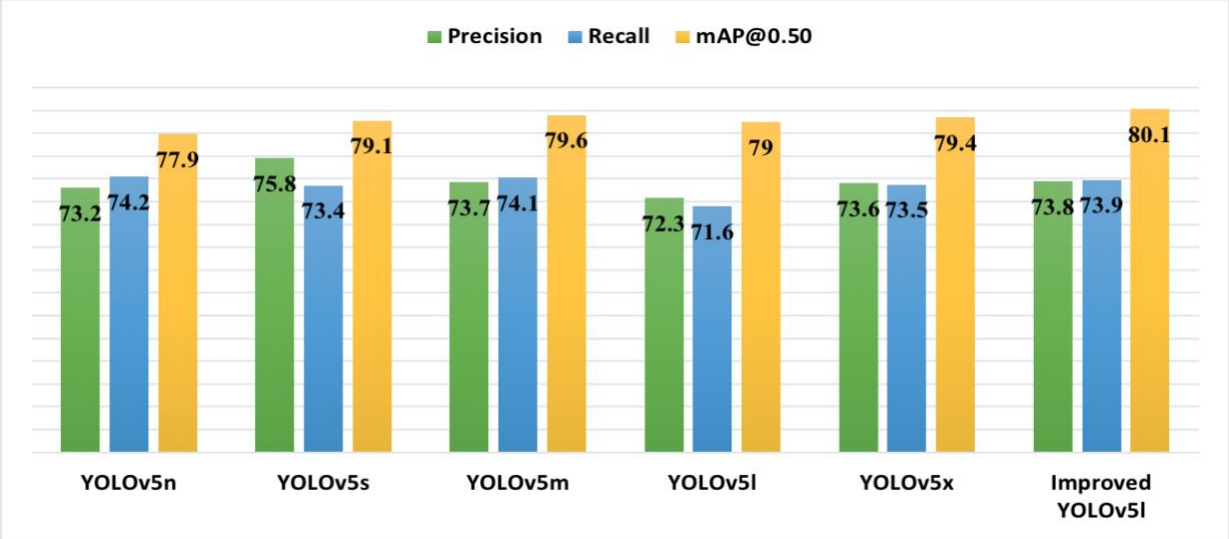


Figure 5. 14 Performance evaluation comparison of result values for YOLOv5 and our improved model

Table 5. 8. All YOLOv5 model versions performance comparison with improved YOLOv5l model.

	Evaluation metrics (%)		
Models	Precision	Recall	mAP@.50
YOLOv5n	73.2	74.2	77.9
YOLOv5s	75.8	73.4	79.1
YOLOv5m	73.7	74.1	79.6
YOLOv5l	72.3	71.6	79
YOLOv5x	73.6	73.5	79.4
Improved YOLOv5l	**73.8**	**73.9**	**80.1**

5.8. Summary

The inadequate adaptability of the model contributes to the obsolescence of the reference model, thus leading to poor accuracy in detecting small symptoms and a notable increase in the rate of missed detections. Identifying and localizing symptom regions associated with multiple diseases and pests posed a significant challenge in terms of detection. This chapter discusses the

enhancement of an adaptive model designed for detecting cucumber leaf diseases and pest symptoms. The model, based on the improved YOLOv5l, utilizes a custom dataset to support adaptive learning and the identification of regions with disease symptoms.

Based on the severity of changes in the YOLOv5l model, the improved YOLOv5l model experienced retraining, which included modifications in various aspects such as the input image size, number of layers, parameters of the convolutional layers including filter size and window size, batch size, and number of epochs. The experimental results obtained provide the improvement in detection accuracy, particularly for instances involving small disease and pest symptoms. Moreover, the findings of this chapter face a difficulty related to recognizing small object instances within image classes. In addition to addressing detection accuracy, the improved YOLOv5l model effectively considers a range of other factors, including detection speed, minimal number of parameters, the presence of small symptom disease spots, and the overall weight sizes.

CHAPTER SIX

6. CONCLUSIONS, RECOMMENDATIONS FOR FUTURE STUDIED

6.1. Conclusion

Disease and pests pose a significant obstacle to cucumber production and quality in agricultural farming. Recognizing various cucumber leaf diseases is of utmost importance for farmers worldwide. Unfortunately, the current methods for diagnosing cucumber leaf diseases and pests manually are often laborious, time-consuming, and subjective. The crucial challenges outlined above have prompted numerous researchers to dedicate their efforts towards improving the accuracy of diagnosing and detecting cucumber leaf diseases and pests to prevent the spread of such diseases and minimize crop damage. Therefore, there is a demanding need for an effective algorithm that enables the diagnosis, classification, and detection of cucumber leaf diseases and pests. This necessity arises from the desire to propose and improve an automated system capable of actively classifying cucumber diseases and health based on their leaves. The study aims to enhance a model for the effective diagnosis and detection of cucumber leaf diseases and pests. It focused on the development of an adaptive model tailored for use in agricultural farming, where the accurate diagnosis of these diseases is crucial. The three contributions have been divided into three phases.

In the first phase, a new structured dataset was created that includes healthy and infected cucumber leaves with single and multi-infections. The data are collected from natural scenes in Kurdistan region, Sulaymaniyah, Rania. It contains two pest diseases (spider, and leaf miner), two fungal diseases (powdery mildew and downy mildew), one viral disease CYSDV, and one healthy leaf. Total images of the dataset are 4868 images. It will also be available as a standard public dataset for a wide range of research community.

The objective of the second phase is to improve the accuracy of recognizing cucumber leaf diseases and pests and minimize the error classification rate caused by the number of image samples and classes in the dataset. This phase

focused on the development of fine-tuned CNN algorithm for cucumber leaf disease recognition system. It is necessary when farmers are unable to diagnose leaf diseases and distinguish healthy cucumber plants. The results showed a the improvement in correct classification accuracy and a reduction in error rates. Significant improvements were observed in cases where the dataset was balanced through data augmentation techniques to enlarge the size of dataset and a reduced number of classes were used.

Quantitative tests confirmed the CNN's superior recognition performance. The accuracy achieved by the proposed CNN model on the unbalanced dataset was 97.53%, while on the balanced dataset, it reached 98.19%. Additionally, the accuracy achieved on the publicly available dataset was 100%. The comparative test results showed that the proposed model outperformed other models, including Pre-trained models (AlexNet, Inception-V3, Resnet-50), and two combined CNN models developed from scratch. The system will be deployed as a real-time mobile application in agriculture farming. It would also be useful for farmers to detect and identify cucumber leaf diseases at early stages.

The third phase, corresponding to the third objective of the PhD dissertation, focused on enhancing the detection capabilities for timely identification of cucumber leaf disease and pest symptoms. The aim was to enable effective management to prevent their spread and minimize crop damage. This phase involved the development of a model for cucumber leaf disease detection, utilizing the improved YOLOv5l network. With the aim of reducing the model's size, modifications were applied to the model's hyper-parameters. Additionally, the BottleneckCSP module was used instead of the C3 module within both the backbone and neck network sections. As a result of reducing the number of parameters, layers, and computations, there was a significant improvement in the detection impact. The experimental results indicated that cucumber leaf disease and pest detection based on the improved YOLOv5l model obtained

80.10% mAP@0.5, 73.8% precision, and 73.9 recall. Furthermore, the training process time is only 2.58 hours. Additionally, the authenticity of the proposed model is demonstrated by incorporating the CBAM into both the improved and original YOLOv5l model. Furthermore, a comparison of the detection findings revealed that the improved YOLOv5l network outperformed the original YOLOv5l, YOLOv5n, YOLOv5s, YOLOv5m, and YOLOv5x networks. The improved model demonstrated better detection accuracy, particularly for large object symptoms and cases involving multiple disease symptoms on a single leaf surface. Additionally, it significantly reduced storage complexity and training time consumption. The system will be deployed as a real-time mobile application, operating in dynamic and immediate environments.

6.2. Recommendation for Future Studies

The results of this study investigation are expected to utilise a considerable influence on the future direction of researches in the domain of plant cucumber leaf disease and healthy recognition and detection systems. An early detection and identification is much considerable preliminary phase. This research presents a notable opportunity and direction for optimizing system efficiency, particularly in constructing lightweight systems. The following perspectives will guide for future studies:

a) Focusing on enlarging the constructed cucumber leaf dataset by increasing disease and pest types and sample image number.
b) Focusing on diagnosing cucumber leaf disease and healthy system that would be a reliable tool for farmers and plant pathologies to help humans save their efforts and reduce plant pesticide usage. Furthermore, the plan will be included expanding the dataset to encompass various disease types and variations, with the goal of early-stage disease detection.

c) Despite the encouraging outcomes obtained from utilizing the improved YOLOv5 network for cucumber leaf disease and pest detection, there is still a need for enhancing the accuracy of detection. The network model structure will be further adjusted in subsequent studies to enhance the network performance of the cucumber leaf disease and pest detection models.

REFERENCES

Agarwal, M., Gupta, S. and Biswas, K.K. (2021), "A new Conv2D model with modified ReLU activation function for identification of disease type and severity in cucumber plant", *Sustainable Computing: Informatics and Systems*, Vol. 30, available at:https://doi.org/10.1016/j.suscom.2020.100473.

Agrawal, S.S., Prasad, M.P.S. and Senthilrajan, D.A. (2021), "Leaf Features Extraction for Plant Classification using CNN", *Researchgate.Net*, Vol. 2 No. 2, available at:https://doi.org/10.48175/IJARSCT-807.

Amara, J., Bouaziz, B. and Algergawy, A. (2017), *A Deep Learning-Based Approach for Banana Leaf Diseases Classification*, *Dl.Gi.De*, available at: https://dl.gi.de/handle/20.500.12116/944 (accessed 5 July 2021).

Analytics vidhya. (2020), "Hyperparameter Tuning using Optuna", available at: https://www.analyticsvidhya.com/blog/2020/11/hyperparameter-tuning-using-optuna/ (accessed 5 July 2021).

Analytics Vidhya. (2020), "Have you heard of this? Machine Learning vs Deep Learning! | by Surya Govind | Analytics Vidhya | Medium", available at: https://medium.com/analytics-vidhya/have-you-heard-of-this-machine-learning-vs-deep-learning-cb7c5633d7d5 (accessed 5 July 2021).

Angelov, P. and Sperduti, A. (2016), "Challenges in deep learning", available at: https://eprints.lancs.ac.uk/id/eprint/134273/ (accessed 12 June 2021).

Arandjelovic, R., Gronat, P., Torii, A., Pajdla, T. and Sivic, J. (2018), "NetVLAD: CNN Architecture for Weakly Supervised Place Recognition", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 40 No. 6, pp. 1437–1451.

Arnal Barbedo, J.G. (2019), "Plant disease identification from individual lesions and spots using deep learning", *Biosystems Engineering*,

Academic Press, Vol. 180, pp. 96–107.

Arsenovic, M., Karanovic, M., Sladojevic, S., Anderla, A. and Stefanovic, D. (2019), “Solving current limitations of deep learning based approaches for plant disease detection”, *Symmetry*, MDPI AG, Vol. 11 No. 7, p. 939.

Ashqar, B.A.M., Abu-Nasser, B.S. and Abu-Naser, S.S. (2019), *Plant Seedlings Classification Using Deep Learning*, available at: www.ijeais.org/ijaisr (accessed 11 June 2021).

Bai, X., Li, X., Fu, Z., Lv, X. and Zhang, L. (2017), “A fuzzy clustering segmentation method based on neighborhood grayscale information for defining cucumber leaf spot disease images”, *Computers and Electronics in Agriculture*, Elsevier, Vol. 136, pp. 157–165.

Balodi, R., Bisht, S., Ghatak, A., Phytopathology, K.R.-I. and 2017, undefined. (2017), “Plant disease diagnosis: technological advancements and challenges”, *Researchgate.Net*, available at:https://doi.org/10.24838/ip.2017.v70.i3.72487.

Barbedo, J.G.A. (2018), “Factors influencing the use of deep learning for plant disease recognition”, *Biosystems Engineering*, Academic Press, Vol. 172, pp. 84–91.

Barbedo, J.G.A. (2019), “Detection of nutrition deficiencies in plants using proximal images and machine learning: A review”, *Computers and Electronics in Agriculture*, Elsevier, Vol. 162, pp. 482–492.

Bashish, D. Al, Braik, M., Technology, S.B.-A.-I. and 2011, U. (2011), “Detection and classification of leaf diseases using K-means-based segmentation and”, *Academia.Edu*, Vol. 10 No. 2, pp. 267–275.

Benuwa, B.B., Zhan, Y., Ghansah, B., Wornyo, D.K. and Kataka, F.B. (2016), “A review of deep machine learning”, *International Journal of Engineering Research in Africa*, Trans Tech Publications Ltd.

Berger, R.D. (1985), “A General Model for Disease Progress with Functions for Variable Latency and Lesion Expansion on Growing Host Plants”,

Phytopathology, Vol. 75 No. 7, p. 792.

Brahimi, M., Boukhalfa, K. and Moussaoui, A. (2017), "Deep Learning for Tomato Diseases: Classification and Symptoms Visualization", *Applied Artificial Intelligence*, Taylor and Francis Inc., Vol. 31 No. 4, pp. 299–315.

Brownlee, J. (2018), *Better Deep Learning: Train Faster, Reduce Overfitting, and Make Better Predictions*, available at: https://books.google.com/books?hl=en&lr=&id=T1-nDwAAQBAJ&oi=fnd&pg=PP1&dq=Better+Deep+Learning.+Train+Faster,+Reduce+Overfitting,+and+Make+Better+Predictions,+Machine+Learning+Mastery+With+Python&ots=tDPNdkl5EZ&sig=fVvGLaGFhLu5H7ktaeHCX8US49E (accessed 12 June 2021).

Cap, Q.H., Suwa, K., Fujita, E., Kagiwada, S., Uga, H. and Iyatomi, H. (2018), "An end-to-end practical plant disease diagnosis system for wide-angle cucumber images", *International Journal of Engineering and Technology(UAE)*, Vol. 7 No. 4, pp. 106–111.

Cen, H., Lu, R., Zhu, Q. and Mendoza, F. (2016), "Nondestructive detection of chilling injury in cucumber fruit using hyperspectral imaging with feature selection and supervised classification", *Postharvest Biology and Technology*, Elsevier, Vol. 111, pp. 352–361.

Chen, J., Chen, J., Zhang, D., Sun, Y. and Nanehkaran, Y.A. (2020), "Using deep transfer learning for image-based plant disease identification", *Computers and Electronics in Agriculture*, Elsevier B.V., Vol. 173, p. 105393.

Chen, Z., Wu, R., Lin, Y., Li, C., Chen, S., Yuan, Z., Chen, S., *et al.* (2022), "Plant Disease Recognition Model Based on Improved YOLOv5", *Agronomy 2022, Vol. 12, Page 365*, Multidisciplinary Digital Publishing Institute, Vol. 12 No. 2, p. 365.

Chlap, P., Min, H., Vandenberg, N., Dowling, J., Holloway, L. and Haworth,

A. (2021), “A review of medical image data augmentation techniques for deep learning applications”, *Journal of Medical Imaging and Radiation Oncology*, John Wiley and Sons Inc, Vol. 65 No. 5, pp. 545–563.

Cruz, A.C., Luvisi, A., De Bellis, L. and Ampatzidis, Y. (2017), “X-FIDO: An effective application for detecting olive quick decline syndrome with deep learning and data fusion”, *Frontiers in Plant Science*, Frontiers Media S.A., Vol. 8, p. 1741.

Dara, S. and Tumma, P. (2018), “Feature Extraction by Using Deep Learning: A Survey”, *Proceedings of the 2nd International Conference on Electronics, Communication and Aerospace Technology, ICECA 2018*, Institute of Electrical and Electronics Engineers Inc., pp. 1795–1801.

Durmus, H., Gunes, E.O. and Kirci, M. (2017), “Disease detection on the leaves of the tomato plants by using deep learning”, *2017 6th International Conference on Agro-Geoinformatics, Agro-Geoinformatics 2017*, Institute of Electrical and Electronics Engineers Inc., available at:https://doi.org/10.1109/Agro-Geoinformatics.2017.8047016.

El-Helly, M., Ahmed, R. and El-Gammal, S. (2003), “An Integrated Image Processing System for Leaf Disease Detection and Diagnosis”, *Proceedings of the 1st Indian International Conference on Artificial Intelligence*, pp. 1182–1195.

Faithpraise, Fina and Chatwin, C. (2013), “Automatic plant pest detection and recognition using k-means clustering algorithm and correspondence filters) Automatic plant pest detection and recognition using k-means clustering algorithm and correspondence filters”, *International Journal of Advanced Biotechnology and Research*, Vol. 4 No. 2, pp. 189–199.

FatihahSahidan, N., Juha, A.K., Mohammad, N. and Ibrahim, Z. (2019), “Flower and leaf recognition for plant identification using convolutional neural network”, *Indonesian Journal of Electrical Engineering and Computer Science*, Institute of Advanced Engineering and Science, Vol.

16 No. 2, pp. 737–743.

Ferentinos, K.P. (2018), “Deep learning models for plant disease detection and diagnosis”, *Computers and Electronics in Agriculture*, Elsevier B.V., Vol. 145, pp. 311–318.

Fuentes, A., Yoon, S., Kim, S.C. and Park, D.S. (2017), “A Robust Deep-Learning-Based Detector for Real-Time Tomato Plant Diseases and Pests Recognition”, *Sensors 2017, Vol. 17, Page 2022*, Multidisciplinary Digital Publishing Institute, Vol. 17 No. 9, p. 2022.

Fujita, E., Kawasaki, Y., Uga, H., Kagiwada, S. and Iyatomi, H. (2016), “Basic investigation on a robust and practical plant diagnostic system”, *Ieeexplore.Ieee.Org*, available at:https://doi.org/10.1109/ICMLA.2016.56.

Fujita, E., Uga, H., Kagiwada, S. and Iyatomi, H. (2018), “A Practical Plant Diagnosis System for Field Leaf Images and Feature Visualization”, *International Journal of Engineering & Technology*, Vol. 7 No. 11, pp. 49–54.

Ganatra, N. and Patel, A. (2020), “A Multiclass Plant Leaf Disease Detection using Image Processing and Machine Learning Techniques”, *International Journal on Emerging Technologies*, Vol. 11 No. 2, p. 388421.

Gu, J., Wang, Z., Kuen, J., Ma, L., Shahroudy, A., Shuai, B., Liu, T., *et al.* (2018), “Recent advances in convolutional neural networks”, *Pattern Recognition*, Elsevier Ltd, Vol. 77, pp. 354–377.

Hammad Saleem, M., Khanchi, S., Potgieter, J. and Mahmood Arif, K. (2020), “Image-Based Plant Disease Identification by Deep Learning Meta-Architectures”, *Plants 2020, Vol. 9, Page 1451*, Multidisciplinary Digital Publishing Institute, Vol. 9 No. 11, p. 1451.

He, D.C., Zhan, J.S. and Xie, L.H. (2016), “Problems, challenges and future of plant disease management: from an ecological point of view”, *Journal*

of Integrative Agriculture, Elsevier, Vol. 15 No. 4, pp. 705–715.

He, K., Zhang, X., Ren, S. and Sun, J. (2016), “Deep Residual Learning for Image Recognition”.

He, L. and Wei, H. (2023), “CBAM-YOLOv5: A Promising Network Model for Wear Particle Recognition”, *Wireless Communications and Mobile Computing*, Hindawi Limited, Vol. 2023, pp. 1–15.

Hernández, S. and López, J.L. (2020), “Uncertainty quantification for plant disease detection using Bayesian deep learning”, *Applied Soft Computing Journal*, Elsevier Ltd, Vol. 96, p. 106597.

Hutter, F., Lücke, J. and Schmidt-Thieme, L. (2015), “Beyond Manual Tuning of Hyperparameters”, *KI - Kunstliche Intelligenz*, Springer, Vol. 29 No. 4, pp. 329–337.

Ibrahim, Z., Sabri, N. and Isa, D. (2018), “Multi-maxpooling Convolutional Neural Network for Medicinal Herb Leaf Recognition”, *Pdfs.Semanticscholar.Org*, available at:https://doi.org/10.12792/icisip2018.060.

Ioffe, S. and Szegedy, C. (2015), “Batch normalization: Accelerating deep network training by reducing internal covariate shift”, *32nd International Conference on Machine Learning, ICML 2015*, Vol. 1, pp. 448–456.

Iqbal, M.S., Ahmad, I., Bin, L., Khan, S. and Rodrigues, J.J.P.C. (2021), “Deep learning recognition of diseased and normal cell representation”, *Transactions on Emerging Telecommunications Technologies*, Vol. 32 No. 7, pp. 1–13.

Jiang, D., Li, G., Sun, Y., Hu, J., Yun, J. and Liu, Y. (2021), “Manipulator grabbing position detection with information fusion of color image and depth image using deep learning”, *Journal of Ambient Intelligence and Humanized Computing 2021 12:12*, Springer, Vol. 12 No. 12, pp. 10809–10822.

Jiao, L., Zhang, F., Liu, F., Yang, S., Li, L., Feng, Z. and Qu, R. (2019), “A

survey of deep learning-based object detection", *IEEE Access*, Institute of Electrical and Electronics Engineers Inc., Vol. 7, pp. 128837–128868.

Jocher, G., Stoken, A., Borovec, J., NanoCode012, ChristopherSTAN, Changyu, L., Laughing, *et al.* (2021), "ultralytics/yolov5: v4.0 - nn.SiLU() activations, Weights & Biases logging, PyTorch Hub integration", available at:https://doi.org/10.5281/ZENODO.4418161.

Jung, H.K. and Choi, G.S. (2022), "Improved YOLOv5: Efficient Object Detection Using Drone Images under Various Conditions", *Applied Sciences 2022, Vol. 12, Page 7255*, Multidisciplinary Digital Publishing Institute, Vol. 12 No. 14, p. 7255.

Kamilaris, A. and Prenafeta-Boldú, F.X. (2018), "Deep learning in agriculture: A survey", *Computers and Electronics in Agriculture*, Elsevier B.V., 1 April.

Karthikayani, K. and Arunachalam, A.R. (2020), "A survey on deep learning feature extraction techniques", *AIP Conference Proceedings*, Vol. 2282, American Institute of Physics Inc., available at:https://doi.org/10.1063/5.0028564.

Kawasaki, Y., Uga, H., Kagiwada, S. and Iyatomi, H. (2015), "Basic study of automated diagnosis of viral plant diseases using convolutional neural networks", *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Springer Verlag, Vol. 9475, pp. 638–645.

Khan, A., Sohail, A., Zahoora, U. and Qureshi, A.S. (2020), "A survey of the recent architectures of deep convolutional neural networks", *Artificial Intelligence Review*, Springer Science and Business Media B.V., Vol. 53 No. 8, pp. 5455–5516.

Khan, M.A., Akram, T., Sharif, M., Javed, K., Raza, M. and Saba, T. (2020), "An automated system for cucumber leaf diseased spot detection and classification using improved saliency method and deep features

selection", *Multimedia Tools and Applications*, Springer, Vol. 79 No. 25–26, pp. 18627–18656.

Kianat, J., Khan, M.A., Sharif, M., Akram, T., Rehman, A. and Saba, T. (2021), "A joint framework of feature reduction and robust feature selection for cucumber leaf diseases recognition", *Optik*, Urban & Fischer, Vol. 240, p. 166566.

Kotsiantis, S., Kanellopoulos, D., Of, P.P.-I.J. and 2006, U. (2006), *Data Preprocessing for Supervised Leaning, Citeseer*, available at: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.104.8413&rep=rep1&type=pdf (accessed 11 June 2021).

Krishnakumar, A. and Narayanan, A. (2019), "A system for plant disease classification and severity estimation using machine learning techniques", *Lecture Notes in Computational Vision and Biomechanics*, Vol. 30, pp. 447–457.

Krizhevsky, A., Sutskever, I. and Hinton, G.E. (2012), "ImageNet Classification with Deep Convolutional Neural Networks", *Advances in Neural Information Processing Systems*, Vol. 25.

Krizhevsky, A., Sutskever, I. and Hinton, G.E. (2017), "ImageNet classification with deep convolutional neural networks", *Communications of the ACM*, ACM PUB27 New York, NY, USA, Vol. 60 No. 6, pp. 84–90.

Layton, O.W. (2019), "CS343: Neural Networks Convolution and Max Pooling".

LeCun, Y., Boser, B., Denker, J.S., Henderson, D., Howard, R.E., Hubbard, W. and Jackel, L.D. (1989), "Backpropagation Applied to Handwritten Zip Code Recognition", *Neural Computation*, MIT Press - Journals, Vol. 1 No. 4, pp. 541–551.

Li, D., Ahmed, F., Wu, N., Plants, A.S.- and 2022, undefined. (2022), "Yolo-JD: A Deep Learning Network for jute diseases and pests detection from

images", *Mdpi.Com*, available at:https://doi.org/10.3390/plants11070937.

Li, X., Yue, Z., Su, J., Wang, S., Hua, J. and Duan, F. (2022), "Application of Lightweight Object Detection Network in Cucumber Leaf Detection", *2022 IEEE International Conference on Mechatronics and Automation, ICMA 2022*, Institute of Electrical and Electronics Engineers Inc., pp. 686–691.

Li, Y. and Fang, J. (2023), "Detection of Surface Defects of Magnetic Tiles Based on Improved YOLOv5", edited by Tang, Y.*Journal of Sensors*, Vol. 2023, pp. 1–12.

Lin, K., Gong, L., Huang, Y., Liu, C. and Pan, J. (2019), "Deep learning-based segmentation and quantification of cucumber powdery mildew using convolutional neural network", *Frontiers in Plant Science*, Frontiers Media S.A., Vol. 10, available at:https://doi.org/10.3389/FPLS.2019.00155/FULL.

Liu, B., Id, †, Zhang, Y., He, D. and Li, Y. (2017), "Identification of Apple Leaf Diseases Based on Deep Convolutional Neural Networks", *Mdpi.Com*, available at:https://doi.org/10.3390/sym10010011.

Liu, C., Zhu, H., Guo, W., Han, X., Chen, C. and Wu, H. (2021), "EFDet: An efficient detection method for cucumber disease under natural complex environments", *Computers and Electronics in Agriculture*, Elsevier, Vol. 189, p. 106378.

Liu, J. and Wang, X. (2020), "Tomato Diseases and Pests Detection Based on Improved Yolo V3 Convolutional Neural Network", *Frontiers in Plant Science*, Frontiers Media S.A., Vol. 11, p. 898.

Lou, Y., Hu, Z., Li, M., Li, H., Yang, X., Liu, X. and Liu, F. (2021), "Real-Time Detection of Cucumber Leaf Diseases Based on Convolution Neural Network", *IEEE Information Technology, Networking, Electronic and Automation Control Conference, ITNEC 2021*, Institute of Electrical and Electronics Engineers Inc., pp. 1040–1046.

Lu, Y., Yi, S., Zeng, N., Liu, Y. and Zhang, Y. (2017), “Identification of rice diseases using deep convolutional neural networks”, *Neurocomputing*, Elsevier B.V., Vol. 267, pp. 378–384.

Ma, J., Du, K., Zheng, F., Zhang, L., Gong, Z. and Sun, Z. (2018), “A recognition method for cucumber diseases using leaf symptom images based on deep convolutional neural network”, *Computers and Electronics in Agriculture*, Vol. 154, pp. 18–24.

Mathew, M.P. and Mahesh, T.Y. (2022), “Leaf-based disease detection in bell pepper plant using YOLO v5”, *Signal, Image and Video Processing*, Springer Science and Business Media Deutschland GmbH, Vol. 16 No. 3, pp. 841–847.

Mia, M., Maria, S., Taki, S., Engineering, A.B.-B. of E. and 2021, undefined. (2021), “Cucumber disease recognition using machine learning and transfer learning”, *Beei.Org*, Vol. 10 No. 6, pp. 3432–3433.

Mikołajczyk, A. and Grochowski, M. (2018), “Data augmentation for improving deep learning in image classification problem”, *2018 International Interdisciplinary PhD Workshop, IIPhDW 2018*, Institute of Electrical and Electronics Engineers Inc., pp. 117–122.

Mohanty, S.P., Hughes, D.P. and Salathé, M. (2016), “Using deep learning for image-based plant disease detection”, *Frontiers in Plant Science*, Frontiers Research Foundation, Vol. 7 No. September, p. 1419.

Mubasir, M. (2020), “Don’t Overfit! II — How to avoid Overfitting in your Machine Learning and Deep Learning Models | by Md. Mubasir | Towards Data Science”, available at: https://towardsdatascience.com/dont-overfit-ii-how-to-avoid-overfitting-in-your-machine-learning-and-deep-learning-models-2ff903f4b36a (accessed 5 July 2021).

Nagasubramanian, K., Jones, S., Singh, A.K., Sarkar, S., Singh, A. and Ganapathysubramanian, B. (2019), “Plant disease identification using

explainable 3D deep learning on hyperspectral images", *Plant Methods*, BioMed Central Ltd., Vol. 15 No. 1, pp. 1–10.

Negm, K. (2020), "Cucumber plant diseases dataset | Kaggle", available at: https://www.kaggle.com/kareem3egm/cucumber-plant-diseases-dataset (accessed 22 March 2022).

Niu, Z., Zhong, G. and Yu, H. (2021), "A review on the attention mechanism of deep learning", *Neurocomputing*, Elsevier, Vol. 452, pp. 48–62.

Omer, S.M., Ghafoor, K.Z. and Askar, S.K. (2022), "An Intelligent System for Cucumber Leaf Disease Diagnosis Based on the Tuned Convolutional Neural Network Algorithm", edited by Podda, S.*Mobile Information Systems*, Vol. 2022, pp. 1–16.

Omer, S.M., Ghafoor, K.Z. and Askar, S.K. (2023), "Plant Disease Diagnosing Based on Deep Learning Techniques", *ARO-THE SCIENTIFIC JOURNAL OF KOYA UNIVERSITY*, Vol. 11 No. 1, pp. 38–47.

Omer, S.M., Hasan, R.M. and Anwer, B.N. (2020), "An Image Dataset Construction for Flower Recognition Using Convolutional Neural Network", *Science Journal of University of Zakho*, Vol. 8 No. 3, pp. 112–117.

Oo, Y.M. and Htun, N.C. (2018), "/licenses/by/4.0 | | Plant Leaf Disease Detection and Classification using Image Processing", *International Journal of Research and Engineering*, Vol. 5 No. 9, pp. 516–523.

Panigrahi, K.P., Das, H., Sahoo, A.K. and Moharana, S.C. (2020), "Maize Leaf Disease Detection and Classification Using Machine Learning Algorithms", *Advances in Intelligent Systems and Computing*, Vol. 1119, Springer, pp. 659–669.

Pawar, P., Turkar, V. and Patil, P. (2016), "Cucumber disease detection using artificial neural network", *Proceedings of the International Conference on Inventive Computation Technologies, ICICT 2016*, Vol. 2016,

available at:https://doi.org/10.1109/INVENTIVE.2016.7830151.

Perez, L. and Wang, J. (2017), “The Effectiveness of Data Augmentation in Image Classification using Deep Learning”, available at: https://arxiv.org/abs/1712.04621v1 (accessed 18 June 2023).

Prabhu. (2018), “Understanding of Convolutional Neural Network (CNN) — Deep Learning”, available at: https://medium.com/@RaghavPrabhu/understanding-of-convolutional-neural-network-cnn-deep-learning-99760835f148 (accessed 12 June 2021).

Pujara, A. (2020), “Concept of AlexNet:- Convolutional Neural Network | by Abhijeet Pujara | Analytics Vidhya | Medium”, available at: https://medium.com/analytics-vidhya/concept-of-alexnet-convolutional-neural-network-6e73b4f9ee30 (accessed 22 October 2023).

Ramachandran, P., Zoph, B. and Le, Q. V. (2017), “Searching for Activation Functions”, *6th International Conference on Learning Representations, ICLR 2018 - Workshop Track Proceedings*, International Conference on Learning Representations, ICLR, available at: http://arxiv.org/abs/1710.05941 (accessed 11 June 2021).

Ramcharan, A., Baranowski, K., McCloskey, P., Ahmed, B., Legg, J. and Hughes, D.P. (2017), “Deep learning for image-based cassava disease detection”, *Frontiers in Plant Science*, Frontiers Media S.A., Vol. 8, p. 1852.

Reyes, A.K., Caicedo, J.C. and Camargo, J.E. (2015), “Fine-tuning Deep Convolutional Networks for Plant Recognition”, *Ceur-Ws.Org*, Vol. 1391, pp. 467–475.

Rice, L., Wong, E. and Kolter, J.Z. (2020), *Overfitting in Adversarially Robust Deep Learning*, *Proceedings.Mlr.Press*, available at: https://github.com/ (accessed 12 June 2021).

Sampath Kumar Gajawada. (2019), “Chi-Square Test for Feature Selection in

Machine learning | by sampath kumar gajawada | Towards Data Science", *Towardsdatascience.Com*, available at: https://towardsdatascience.com/chi-square-test-for-feature-selection-in-machine-learning-206b1f0b8223 (accessed 24 October 2023).

Sannakki, S., Nargund, V., Kulkarni, P., Sannakki, S.S., Rajpurohit, V.S. and Nargund, V.B. (2013), "Diagnosis and classification of grape leaf diseases using neural networks", *Ieeexplore.Ieee.Org*, available at:https://doi.org/10.1109/ICCCNT.2013.6726616.

Saufi, M.M., Zamanhuri, M.A., Mohammad, N. and Ibrahim, Z. (2018), "Deep learning for roman handwritten character recognition", *Indonesian Journal of Electrical Engineering and Computer Science*, Institute of Advanced Engineering and Science, Vol. 12 No. 2, pp. 455–460.

Shen, W., Wu, Y., Chen, Z. and Wei, H. (2008), "Grading method of leaf spot disease based on image processing", *Proceedings - International Conference on Computer Science and Software Engineering, CSSE 2008*, Vol. 6, pp. 491–494.

Shorten, C. and Khoshgoftaar, T.M. (2019), "A survey on Image Data Augmentation for Deep Learning", *Journal of Big Data*, Vol. 6 No. 1, available at:https://doi.org/10.1186/s40537-019-0197-0.

Shruthi, U., Nagaveni, V. and Raghavendra, B.K. (2019), "A Review on Machine Learning Classification Techniques for Plant Disease Detection", *2019 5th International Conference on Advanced Computing and Communication Systems, ICACCS 2019*, Institute of Electrical and Electronics Engineers Inc., pp. 281–284.

Sladojevic, S., Arsenovic, M., Anderla, A., Culibrk, D. and Stefanovic, D. (2016), "Deep Neural Networks Based Recognition of Plant Diseases by Leaf Image Classification", *Computational Intelligence and Neuroscience*, Hindawi Publishing Corporation, Vol. 2016, available at:https://doi.org/10.1155/2016/3289801.

Song, L., Liu, M., Liu, S., Wang, H. and Luo, J. (2023), “Pest species identification algorithm based on improved YOLOv4 network”, *Signal, Image and Video Processing*, Springer Science and Business Media Deutschland GmbH, pp. 1–8.

Srivastava, N., Hinton, G., Krizhevsky, A. and Salakhutdinov, R. (2014), *Dropout: A Simple Way to Prevent Neural Networks from Overfitting*, *Journal of Machine Learning Research*, Vol. 15, available at: https://www.jmlr.org/papers/volume15/srivastava14a/srivastava14a.pdf?utm_campaign=buffer&utm_content=buffer79b43&utm_medium=social&utm_source=twitter.com (accessed 12 June 2021).

Sun, G., Jia, X. and Geng, T. (2018), “Plant Diseases Recognition Based on Image Processing Technology”, *Journal of Electrical and Computer Engineering*, Hindawi Limited, Vol. 2018, available at:https://doi.org/10.1155/2018/6070129.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J. and Wojna, Z. (2016), “Rethinking the Inception Architecture for Computer Vision”, *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, IEEE Computer Society, Vol. 2016-December, pp. 2818–2826.

Tani, H., Kotani, R., Kagiwada, S., Uga, H. and Iyatomi, H. (2018), “Diagnosis of Multiple Cucumber Infections with Convolutional Neural Networks”, *Proceedings - Applied Imagery Pattern Recognition Workshop*, Vol. 2018-Octob, available at:https://doi.org/10.1109/AIPR.2018.8707385.

Tian, H., Wang, T., Liu, Y., Qiao, X. and Li, Y. (2020), “Computer vision technology in agricultural automation —A review”, *Information Processing in Agriculture*, Elsevier, Vol. 7 No. 1, pp. 1–19.

Tian, Y., Yang, G., Wang, Z., Wang, H., Li, E. and Liang, Z. (2019), “Apple detection during different growth stages in orchards using the improved

YOLO-V3 model", *Computers and Electronics in Agriculture*, Elsevier, Vol. 157, pp. 417–426.

Too, E.C., Yujian, L., Njuki, S. and Yingchun, L. (2019), "A comparative study of fine-tuning deep learning models for plant disease identification", *Computers and Electronics in Agriculture*, Vol. 161, pp. 272–279.

Uçar, A., Demir, Y. and Güzeliş, C. (2017), "Object recognition and detection with deep learning for autonomous driving applications", *Simulation*, SAGE Publications Ltd, Vol. 93 No. 9, pp. 759–769.

Umer, S., Rout, R.K., Pero, C. and Nappi, M. (2022), "Facial expression recognition with trade-offs between data augmentation and deep learning features", *Journal of Ambient Intelligence and Humanized Computing*, Springer Science and Business Media Deutschland GmbH, Vol. 13 No. 2, pp. 721–735.

Victoria, A.H. and Maragatham, G. (2021), "Automatic tuning of hyperparameters using Bayesian optimization", *Evolving Systems*, Springer Science and Business Media Deutschland GmbH, Vol. 12 No. 1, pp. 217–223.

Wang, H., Shang, S., Wang, D., He, X., Feng, K. and Zhu, H. (2022), "Plant Disease Detection and Classification Method Based on the Optimized Lightweight YOLOv5 Model", *Agriculture 2022, Vol. 12, Page 931*, Multidisciplinary Digital Publishing Institute, Vol. 12 No. 7, p. 931.

Wang, Q., Cheng, M., Huang, S., Cai, Z., Zhang, J. and Yuan, H. (2022), "A deep learning approach incorporating YOLO v5 and attention mechanisms for field real-time detection of the invasive weed Solanum rostratum Dunal seedlings", *Computers and Electronics in Agriculture*, Elsevier, Vol. 199, p. 107194.

Wu, S., Zhong, S. and Liu, Y. (2018), "Deep residual learning for image steganalysis", *Multimedia Tools and Applications*, Springer, Vol. 77 No.

9, pp. 10437–10453.

Yang, R., Hu, Y., Yao, Y., Gao, M. and Liu, R. (2022), “Fruit Target Detection Based on BCo-YOLOv5 Model”, *Mobile Information Systems*, Hindawi Limited, Vol. 2022, available at:https://doi.org/10.1155/2022/8457173.

YueJu, X., Ning, H., ShuQin, T., Liang, M., AQing, Y., XunMu, Z., XiaoFan, Y., *et al.* (2018), “Immature mango detection based on improved YOLOv2.”, *Transactions of the Chinese Society of Agricultural Engineering*, Chinese Society of Agricultural Engineering, Vol. 34 No. 7, pp. 173–179.

Zbakh, A., Mdaghri, Z.A., Benyoussef, A., El Kenz, A. and El Yadari, M. (2019), “Spectral classification of a set of hyperspectral images using the convolutional neural network, in a single training”, *International Journal of Advanced Computer Science and Applications*, Vol. 10 No. 6, pp. 245–250.

Zhang, J., Rao, Y., Man, C., Jiang, Z. and Li, S. (2021), “Identification of cucumber leaf diseases using deep learning and small sample size for agricultural Internet of Things”:, *Https://Doi.Org/10.1177/15501477211007407*, SAGE PublicationsSage UK: London, England, Vol. 17 No. 4, available at:https://doi.org/10.1177/15501477211007407.

Zhang, J. and Zhang, W. (2010), “Support vector machine for recognition of cucumber leaf diseases”, *Proceedings - 2nd IEEE International Conference on Advanced Computer Control, ICACC 2010*, Vol. 5, pp. 264–266.

Zhang, S. and Wang, Z. (2016), “Cucumber disease recognition based on Global-Local Singular value decomposition”, *Neurocomputing*, Vol. 205, pp. 341–348.

Zhang, S., Wu, X., You, Z. and Zhang, L. (2017), “Leaf image based

cucumber disease recognition using sparse representation classification", *Computers and Electronics in Agriculture*, Elsevier, Vol. 134, pp. 135–141.

Zhou, F., Deng, H., Xu, Q. and Lan, X. (2023), "CNTR-YOLO: Improved YOLOv5 Based on ConvNext and Transformer for Aircraft Detection in Remote Sensing Images", *Electronics 2023, Vol. 12, Page 2671*, Multidisciplinary Digital Publishing Institute, Vol. 12 No. 12, p. 2671.

پوختە

لە کشتوکاڵدا، مێروەکان و نەخۆشییە ڕووەکییەکانی تر گرنگترین فاکتەرن کە کاریگەریان لەسەر گەڵای خەیار هەیە کاتێک گەشە دەکەن. نەبوونی دەستنیشانکردنی خیرا و کۆنترۆڵ نەکردنیان لەکاتی پیویستدا لەوانەیە ببێتە هۆی کەمبوونەوەی بەرهەم یان لەناوچونی ڕووەکەکە. لە ئێستادا جوتیاران لە سەرانسەری جیهاندا ڕووبەڕووی کێشە بونەتەوە لە ناسینەوەی نەخۆشییە جۆراوجۆرەکانی گەڵای خەیار. بەداخەوە، تەکنیکەکانی ئێستا بۆ دەستنیشانکردن و دۆزینەوەی نەخۆشییەکانی گەڵای خەیار بڕێکی زۆر لە سەرچاوە مرۆییەکان بەکاردەهێنێت وە کاتیکی زۆریش بەفیڕۆدەدات. لەبەر ئەوە، پێویستییەکی داواکراو هەیە بۆ پەرەپیدانی ئەلگۆریتمێکی کاریگەر کە توانای دەستنیشانکردن و پۆلێنکردن و دۆزینەوەی نەخۆشییەکانی گەڵای خەیاری هەبیت بە شیوەیەکی ئۆتۆماتیکی. مەبەستی ئەم تێزە پەرەپیدان و باشترکردنی مۆدێلێکی ئۆتۆماتیکیە، لەسەر بنەمای بەکارهێنانی تەکنیکەکانی فێربوونی قووڵ بۆ دەستنیشانکردن و پۆلینکردنی نەخۆشیەکانی گەڵای خەیار. لە لێکۆڵینەوەکەدا، چەندین کێشە و گرفت دۆزراونەتەوە کە پێویستیان بە چارەسەر هەیە. لە هەمووی گرنگتر، نەبوونی داتای گشتی باوەڕپێکراوە بۆ وێنەی نەخۆشی گەڵای خەیارکە بەردەست بیت بۆ بەکارهینان لەلایەن توێژەرانەوە. دواتر، پەرەپیدانی ئەلگۆریتمێکی ئۆتۆماتیکی کاریگەرپیویستە لەسەربنەمای فیربونی قوڵی (CNN) بۆ پۆلێنکردنی نەخۆشییەکانی گەڵای خەیاربە شیوەیەکی وردودروست. سێیەم، کیشەیەکی ترتیبینی کراوە سەبارەت بە مۆدێلی (YOLOv5) لە دۆزینەوەی نیشانە بچووکەکانی نەخۆشی گەڵای خەیار، بە فیرۆدانی کات بۆ پۆلین کردن وە کیشەی قەبارەی مۆدیلەکە بۆ داگیرکردنی بیرگە ،وە خراپی ئەنجامی ناسینەوەی جۆری نەخۆشیەکان.

بۆ چارەسەری ئەم کیشانە چەند هەنگاویك ئەنجام دراوە لەم تیزەدا، لەوانە: یەکەم، بۆ زاڵبوون بەسەر نەبوونی داتایەکی گشتیدا، داتاسیتیکی نوی دروست کراوە کە کۆمەڵێك وینەی نوێ لە نەخۆشیە جۆراوجۆرەکانی گەڵای خەیار لەخۆ دەگرێت کە پیکهاتوە لە دوو جۆری میروەکان (spider, leaf miner)، دوو جۆری نەخۆشی کەڕوو (downy mildew, powdery mildew)، یەک جۆری نەخۆشی ڤایرۆسی وە یەك پۆلی گەڵای تەندروست، کە لە کوردستان وینەکان کۆکراونەوە، وە کۆی گشتی 4868 وێنەی گەڵای خەیار لەخۆدەگرێت. لەگەڵ ئەوەشدا، ئەم تیزی دکتۆرایە جەخت دەکاتەوە لەسەر پەرەپیدانی ئەلگۆریتمێک لەسەربنەمای فیربونی قوڵی (CNN) لەگەڵ ڕێکخستنی هایپەرپارامیتەرەکان بۆ باشترکردنی ئەدای مۆدێلەکە لە ناسینەوەی پێنج جۆری نەخۆشی وگەڵای تەندروست ، کە باشترکردنی وێنەکان و دەرهێنانی تایبەتمەندی وینەکان و پۆلێنکردنیان لەخۆ دەگرێت. زیادکردنی داتا وەک هەنگاوێک پێش پرۆسیس کردن بەکارهێنراوە

بۆمەبەستی زیادکردنی ژمارەی وینەکان لە داتاسێتەکە. بە شێوەیەکی ئۆتۆماتیکی تایبەتمەندییەکان دەرهێنراون بە بەکارهێنانی چینەکانی CNN. پاشـــان پێنج نەخۆشـــی گەڵای خەیار و یەک گەڵای تەندروست پۆلێن دەکریت. سەرەرای ئەوەش، بۆ دۆزینەوەی نیشانەکانی نەخۆشی سەرگەڵاکان بە وردی، مۆدێلیکی دۆزینەوەی نیشـانەکان پەرەپیپدراوە لە سـەرەتاوە لەسـەر بنەمای مۆدێلی فیربونی قـوڵـی (YOLOv5). بـۆ کەمـکـردنەوەی قەبـارەی مـۆدێـلەکە، مـۆدیـولـی BottleneckCSP بەکارهێنراوە لەجیاتی C3 لە پیکهاتەی بەشەکانی مۆدیلەکە وەك (Backbone and Neck) لەگەڵ گۆرانکـاری لـە هـایپەرپـارامیتـەرەکـانی مۆدێلیـەکـە. بەهۆی کەمبوونەوەی ژمـارەی پـارامیتەرەکـان، ژمارەی چینەکان، کاریگەری دۆزینەوەکە بە شێوەیەکی بەرچاو باشتر بوو؛ لەگەڵ ئەوەشدا، مۆدێلی باشـــترکراو توانای ئەوەی هەیە نیشـــانەی بچوکی گەڵای نەخۆشـــی بدۆزێتەوە. وە بۆ باشـــترکردنی پرۆســـــەی دەرهینـانی تـایبـەتمـەنـدی وینـەکـان (CBAM) مۆدیول زیـاد کرا بۆ هەردوو مۆدێلی پەرەپیدراوو ڕەسەنی (YOLOv5).

بۆ هەڵسـەنگاندنی ئەنجامەکانی مۆدیلی (CNN)ی پەرەپیدراو، تاقیکردنەوەی بەراوردکاری ئەنجامدرا بە بەکارهینانی جەند مۆدیلیکی جیاوازوەك (AlexNet،Inception-V3 ResNet-50) کە پیشـــتر ڕاهینراون لەســـەرزانیاری تر.ئەنجامە تاقیکردنەوەییەکانی تیزەکە ئەوەی پشـــتڕاســـت کردەوە کە ئەلگۆریتمی پێشنیارکراوی CNN کاریگەر بووە بۆ ناسینەوەی نەخۆشی گەڵای خەیار بە بەراورد لەگەڵ ئەلگۆریتمەکانی تر، وە لەســـەر بنەمای هەردوو داتاســـیتەکە بەبێ زیادکردنی داتا وە دوای زیادکردنی داتا. مۆدیلی CNN ی پەرەپیدراو ڕێژەی ناسینەوەی ٪٩٨.١٩ ی بەدەست هیناوە. هەروەها، ئەنجامە تاقیکردنەوەییەکانی مۆدێلی پەرەپیدراوی دۆزینەوەی وردی (YOLOv5) نیشانی دا، کە تێکڕای ناســـینەوەی نەخۆشـــی گەڵای خەیار لەســـەر بنەمای پیوەری (mAP) بە ریژەی ٪٨٠.١٠ بوو، لە کاتێکدا پیوەرەکانی وەك (precision, recall) بە ڕێژەی ٪٧٣.٨ ، ٪٧٣.٩ بوو بەدوای یەکدا. لەگەڵ ئەوەشـــدا ، قەبارەی مۆدێلی پەرەپیدراوکەمتر بۆتەوە بە بەراورد بە قەبارەی مۆدیلەکانی تر لە ٩٢.٨ مێگابایت بۆ 13.6 مێگابایت کە وای کردوە شـوینیکی کەمتر لە بیرگە داگیر بکات . ســـەرەڕای ئەمەش کاتی کەمترکردۆتەوە لە 4 کاتژمێر و 41 خولەک بۆ تەنها 2 کاتژمێر و 58 خولەک لە پرۆسەی فیربون بەراورد بە مۆدیلەکانی تر.