Ministry of Higher Education and
Scientific Research
University of Diyala
College of Science
Department of Computer Science

Plant Leafs Diseases Detection and Classification Based on Deep Learning Techniques

A Thesis
Submitted to the Department of Computer Science\ College of Sciences\ University of Diyala in a Partial Fulfillment of the Requirements for the Degree of Master in Computer Science

By

Marwan Adnan Jasim

Supervised By

Asst. Prof. Dr. Jamal Mustafa Al-Tuwaijari

2020 A.D. **Iraq** **1441 A.H.**

بسم الله الرحمن الرحيم

﴿ وَآيَةٌ لَّهُمُ الأَرْضُ المَيْتَةُ أَحيَينَاهَا وَأَخْرَجْنَا مِنْهَا حَبّاً فَمِنْهُ يَأْكُلُوْنَ * وَجَعَلْنَا فِيْهَا جَنَّاتٍ مِّنْ نَّخيْلٍ وأَعْنابٍ وفَجَّرْنَا فِيهَا مِنَ الْعُيُونِ * لِيأْكُلُوا مِنْ ثَمَرِهِ وَمَا عَمِلَتْهُ أَيْدِيْهِم أَفَلا يَشْكُرُونَ * سُبْحَانَ الَّذِى خَلَقَ الأَزْوَاجَ كُلَّهَا مِمَّا تُنْبِتُ الأَرْضُ وَمِنْ أَنْفُسِهِمْ وَمِمَّا لَا يَعْلَمُونْ ﴾

صدق الله العظيم

سورة يس/ (33 - 36)

Acknowledgment

First of all, praise is to Allah, the lord of the entire creation, on all the blessing was the aid in achieving this work to its end. I would like to express my appreciation to my supervisor, Dr Jamal Mustafa AL-Tuwaijari for his faithful guidance, valuable instructions, and constructive comments which have made the completion of this work possible.

Also, I would like to express my gratitude and thanks to all the teaching staffs who have taught me. Special thanks are extended to all my friends for their help. Special thanks to the members of the evaluation committee for discussion my thesis. My final words go to my family. I want to thank my parents, my wife and my daughter for their love and guidance which helped me in achieving my goals.

Marwan Adnan –AL-Ahbabi

Dedication

To those who brought people out of the darkness of ignorance to the light of guidance is the beloved of our hearts, Messenger of God), may God bless him

....

To heaven under her feet ... My kind, compassionate mother.

....

To the most precious people ... My dear father

....

To my life partner... My beloved wife.

....

To my pride in my life ... All my brothers and friends.

....

To the bud of my life ... My daughter (Myaar).

Supervisors' Certification

I certify that this thesis entitled ***"Plant Leafs Diseases Detection and Classification Based on Deep Learning Techniques "*** was prepared by ***"Marwan Adnan Jasim"*** under my supervisions at the University of Diyala Faculty of Science Department of Computer Science, as a partial fulfillment of the requirements needed to award the degree of Master of Science in Computer Science.

(Supervisor)
Name: Asst. Prof. Dr. Jamal Mustafa AL-Tuwaijari

Signature:
Date: / / 2020

Approved by University of Diyala\ College of Science\ Department of Computer Science.

Signature:
Name: Asst. Prof. Dr. Taha Mohammed Hasan
Date:
Head of Computer Science Department

Scientific Amendment

I certify that the thesis entitled ***"Plant Leafs Diseases Detection and Classification Based on Deep Learning Techniques "*** was prepared by ***"Marwan Adnan Jasim"*** *has been evaluated scientifically; therefore, it is suitable for debate by examining committee.*

Signature:
Name :
Date : **/ / 2020**

Linguistic Certification

This is to certify that this thesis ***"Plant Leafs Diseases Detection and Classification Based on Deep Learning Techniques "***was prepared by ***"Marwan Adnan Jasim"*** at the University of Diyala/ Department of Computer Science, is reviewed linguistically. Its language was amended to meet the style of the English language.

Signature:
Name :
Date : / / 2020

Abstract

Agricultural products are one of the basic needs of people's lives in any country and backbone of its economy. So, plants diseases impacts the country's agricultural production and its economic resources; these are such reasons which encourage and prompted us for working in this field. Analyzing the image of the leaf of plants and extracting the most important characteristics of these diseases and relying on image analysis and processing techniques to distinguish between different diseases based on the symptoms of each disease.

This work adopts two proposed approaches for detecting plant leaf diseases using image processing and deep learning techniques with makes a comparison between these two approaches. The dataset used in this work was obtained from the (Plant Village website), which is certified and benchmark dataset. This work was planned to some an important and common group of plants, including tomatoes, pepper, and potatoes. These types of plants are the most common types in the world and particularly in Iraq. The dataset contains 20636 images of infected plants in these aforementioned types that were adopted in the proposed approach. In the first proposed approach, where several steps are used in the form of stages, which are include, the image acquisition stage, image pre-processing, image segmentation, image post-processing, extraction the features, and the classification stage. Multi class support vector machine (MSVM) algorithm was used to perform the classification process.

In the second proposed approach, the convolution neural network (CNN) was used through which the leaves of diseased plants are classified according to a special structure of this algorithm consisting of several layers. In these two proposed approaches, 15 classes were classified, including 12 classes for diseases of different plants that were detected, such as bacteria, fungi, viruses, etc., and 3 classes for health leaves of the three used types.

The obtained results from the comparison between the two proposed approaches in terms of performance and accuracy showed the preference of the second approach which adopted the deep learning and using the CNN algorithm, over the first approach, because the overall accuracy rate that obtained from the second proposed approach was (98,029%) in the testing stage and (98.29%) in the training stage. While the overall accuracy rate that obtained from the first proposed approach was (61.79%) in the testing stage and (69.33%) for the training stage. So, the second proposed approach is more accurate and powerful in the process of detecting and classifying plants leaf diseases.

Contents

List of Abbreviations

Abbreviation	Description
1VR	One-Versus-Rest
2D	Two Dimension
ANN	Artificial Neural Network
ASM	Angular Second Moment
CNN	Convolution Neural Network
ECOC	Error-Correcting Output Codes
FE	Feature Extraction
FNN	Feed-forward Neural Network
GLCM	Gray Level Co-occurrence Matrix
HSV	Hue, Saturation, Value
IDM	Inverse Difference Moment
ML	Machine learning
MLP	Multi-layer Perception
MSVM	Multi-class Support Vector Machine
RBF	Radial Basis Function
RBM	Restricted Boltzmann Machines
ReLU	Rectified Linear Units
RGB	Red, Green, Blue
RMS	Root Mean Square
RNN	Recurrent Neural Networks
SGDM	Stochastic Gradient Descent with Momentum
VGG	Visual Geometry Group
μ	Mean
σ, SD	Standard Deviation

List of Figures

List of Tables

Chapter One

General Introduction

Chapter one

General introduction

1.1 Overview

Plants are one of the essential elements of life and living on earth and play a significant role in our world. People in this world depend mainly on plants for food, either directly or by using them as feed for different pets [1]. It played essential roles in the production of various vegetables and fruits, as well as medicinal plants that contain organic substances for the manufacture of pharmaceutical compounds and forests, which are the primary source of wood [2].

The extinction of plants is one of the main problems that the world fears these days because of the large number and variety of serious diseases afflicting it. Moreover, the diseases and insect pests are the major problems in agribusiness. These require careful diagnosis and timely opportune dealing with to protect the crops from heavy losses in farms. The diseases can be found in various parts of plants such as leaves, stems and other parts, but the leaf remain the most important part of the plants, which covers most of its parts [3].

To reduce diseases that occurs in plants there are a variety of strategies in place to detect infections in plants in the initial stages. A traditional strategy for detect of plant leaves diseases is naked-eye observation techniques and it isn't powerful especially for an enormous harvest. By using digital image processing and deep learning a technique, disease plant detection becomes more efficient, accurate, and fewer efforts and time spending [4].

In this regard, many research and studies show the effectiveness of image processing and analysis by using multiple techniques and also using deep learning techniques to obtain results and high accuracy in detecting various diseases of plants. This encourages and drives research to improve reliability, correctness, and accuracy of image analysis for plant disease detection over the past twenty years [5]. The basic concepts and standards remain the same, but the development lies in increasing accuracy and using computer vision and the principle of analysis in this field. Using these techniques, the input image of the particular plant is analyzed so that the things inside it are separated from the background of this image. The image analysis process is utilized to separate the objects from the background, thereby detach quantifiable data which is utilized in various control operations of decision making [5].

1.2 Introduction to Image Processing

Image processing expresses a form of signal processing, in which the images are entered for processing, and in the output it is in the form of either features or parameters specific to that image. It is used in many sections of the most important preprocessing in the classification and detection of objects in images. Moreover, it is a computer imaging where the application involves a human being in the visual loop. In other words, the images are to be examined and are acted upon by people. The categories for image processing are many of including improving images, compressing, segmentation, analyzing them, and other different categories [6].

Detection of diseases through image processing are done by using an integrated approach working methods of processing images of plants entering

it and distinguishes this approach if the plants are healthy or sick of one of the diseases that affect them. It was also mentioned previously that manually detecting plant-specific diseases is an inaccurate and ineffective work, but using modern techniques for image processing. Analyzing the image of the leaf of plants and extracting the most important characteristics of these diseases and relying on image analysis and processing techniques to distinguish between different diseases based on the symptoms of each disease. Using these methods is especially useful when observing large plantations [7].

1.3 Related Work

In this section, the study reviews some of the several approaches and methods using image processing technique and deep learning that is used for plants leaves diseases discovery and classification are presented:

In 2015, Kawasaki, et al. [8], were proposed the CNNs algorithm that is used to differentiate nutritious cucumbers from diseased cucumbers by utilizing images of leaves. They were diagnosed two dangerous viral infections with CNN in this survey, "MYSV" (Melon Yellow Spot Virus), and "ZYMV" (Zucchini Yellow Mosaic Virus). They utilized dataset comprises of 800 images of leaves of cucumbers (200 with ZYMV, 300 with MYSV, or 300 with healthy leaves). The dataset was enlarged using rotational transformations on images. Researchers suggested CNN structure consist of 3 layers used are (convolutional layers, pooling layers, and normalization layers). The activation method applied in the approach is the function "Rectified Linear Unit" (ReLU), the precision obtained in this analysis was (94.9%).

In 2015, Mokhtar, Usama, et al. [9], were introduced a system that classifies and detects two classes of tomato diseases, these diseases are viral (Tomato yellow leaf curl virus (TYLCV)) and (Tomato yellow leaf curl disease (TYLCD)). The system is divided into 4 main stages: preprocessing stage, image segmentation, feature extraction, and a classification stage. The classification was done using the support vector machine algorithm (SVM) on a set of data and includes 200 infected images. The system was trained and tested to perform the classification process for diseases. The classification accuracy ranges between (90%) and (92%) for classifying tomato plant leaf diseases.

In 2016, Prajakta Mitkal, et al. [10], they were investigated image processing procedures for recognizing the illness of sugarcane leaves. The researchers chose six types of disease to be explored: ("Brown spot", "Downy mould", Red Stripe", "Sugarcane Mosaic", "Red Decay" and "Downy Fungal"). For images acquired, images were captured in higher quality format targets, e.g. JPEG, TIF, and BMP for image investigative process. In preprocessing step the RGB images were transformed into grayscale, and unintended information was ejected from the images. Segmentation showed a normal region of the image that includes green pixels and a feasibly infected region. Three algorithms, in particular (Linear SVM, Non-linear SVM and Multi-Class SVM), were utilized to extract features and genetic algorithm was used for detect infection place and classify the illnesses.

In 2017, Amara Jihen, et al. [11], were suggested a methodology where they detect and classify by apply deep neural networks to determine two famous banana diseases which are “Banana Speckle", and "Banana Sigatoka". The model has been trained under difficult conditions, represented by the

lighting and conditions in the images, as well as the background of images, the resolution of the images, their size and direction. The success of the methodology followed and the attempt to use simple mathematical operations has emerged.

In 2017, Atabay, Habibollah Agh [12], were proposed work depends on deep residual learning strategy. Convolution Neural Networks (CNNs) have been used to identify tomato plant leaves pictures that are relying on the apparent impact of illnesses. In comparison to transfer learning as a convincing technique. To get that, the design of CNN is proposed and applied to a subset of the Plant Village dataset, including tomato plant leaf picture. The outcomes viewed the superiority of the proposed design VGG models, pre-trained on the image dataset, in both exactness and the time required for re-training; and it tends to be utilized with a standard PC with no additional equipment required. A common feature visualization and verification are likewise applied to the outcomes and further talks are made to infer the significance of background pixels encompassing the leaves.

In 2018, Tm, Prajwala, et al. [13], aimed at recognition and identify ten various classes of diseases in the tomato plant. In this system, the convolutional neural network (CNN) was applied to classify the illness of the tomato leaf gained from the (plant village dataset). The convolution neural network was used and was built with the lowest number of layers to facilitate classification, though it were classified ten categories of different illnesses. This model can be utilized as a choice apparatus to assist and encourage farmers in recognizing the diseases that can be found in the tomato product with a percentage of (94-95%) of the accuracy, this methodology can perform disease detection using easy and uncomplicated calculations.

In 2018, P. Goncharov1, et al. [14], they utilized the Plant Village open database during this research; the synthetic nature of the sets can dangerously influence the accuracy of the neural model while processing real-life images. They gathered a special database of the grape leaves comprising of four arrangements of images deep “Siamese convolution network” was developed to determine the problem of the poor image databases. Accuracy over 90% arrived in the detection of the Esca, Black rot and Chlorosis diseases on the grape leaves. Comparative results of different models and plants utilizing are introduced.

In 2019, Ozguven, Mehmet Metin, and Kemal Adem[15], were developed a system to classify diseases of the sugar beet plant using 155 different images of diseases of this plant. They were developed a structure for the CNN algorithm using faster parameters for automation using the R-CNN structure. Also in this structure, the activation function Rectified Linear Unit (ReLU) was used. The rate overall correct accuracy reached 95.48%, and they believed that this method would reduce time and effort in detecting diseases of sugar beet plants on large farms.

In 2019, Francis, Mercelin, and C. Deisy,[16], they were created a deep learning model using the CNN algorithm. This model consists of four convolutional layers, and each layer is followed by pooling layer Also, the fully connected dense layers and also the sigmoid function was utilized for detecting the likelihood of disease occurring or not. This model classified the sick and healthy leaves of tomato and apple plants using the dataset consisting of 3663 pictures with the classification accuracy reached 87%.

In 2020, Sambasivam, G., and Geoffrey Duncan Opiyo., [17], they took the dataset by conducting a regular survey on Uganda farms taking 10,000 images of cassava leaf. Their study aimed to classify 5 different categories of cassava diseases. Deep learning and the "Convolution Neural network" algorithm were used to construct and structure their own classification, and they mentioned several difficulties that occurred in data balance for the used dataset. Classification accuracy was achieved higher than 93% for the classification of cassava leaf diseases.

1.4 Problem Definition

There are several problems that are illustrated as follows:

1. The traditional method for plant diseases detection is a natural eye view strategy which is not effective for a large crop and takes a lot of time and effort. With the existence of new techniques to design a approach that can detect and classify the diseases of plant leaves, the machine learning and deep learning technologies have been emerged as one of the powerful and effective approach for detecting and classifying the plant leaves diseases. These techniques with the support of computer facility result in easy access and fast manner that have a high degree of accuracy in diagnosis and classification of diseases
2. Image processing techniques and using support vector machine (SVM) algorithm for classification were ineffective when using large data and detecting a large number of diseases, unlike deep learning techniques, they are effective despite the large data volume and the number of multiple diseases.

1.5 Aim of the Thesis

The main aim of this thesis is to design and implement an effective and efficient approach for detection of plant leaf diseases based on image processing and machine learning techniques to achieve a high degree of accuracy, in addition, making a comparison between these techniques to determine the best ones between them.

1.6 Outline of the Thesis

This work for plant leaf diseases detection is structured in five chapters; here is a brief description of their contents is given:

Chapter 2: This chapter presents theoretical backgrounds and an overview of image processing techniques, plant leaf with their diseases and machine learning with its sections. Also, it describes the plant leaf diseases approach, with an explanation of all the algorithms used in the proposed approach with all examples and detailed equations.

Chapter 3: This chapter produces the details of the proposed approach plant leaf diseases detection and the implementation of it.

Chapter 4: This chapter gives the experimental results obtained from the implementation of plant leaf diseases detection approaches.

Chapter 5: This chapter includes conclusions and future work for the development plant leaf diseases detection approaches with lists a number of suggestions for future studies.

Chapter Two

Theoretical Background

Chapter Two

Theoretical Background

2.1 Introduction

Agriculture is one of the most important fields of life in all countries of the world and many countries economy depend entirely on agricultural products. Therefore, protection of agricultural crops and their products from diseases must be at a high level of importance. In addition, attention must be paid to developing new methods and techniques to increase their productivity [18].

Diseases and defects that infect plants and their product greatly affect agriculture and lead to significant economic losses [19]. If the diseases are not diagnosed at their beginnings, it will be damaging the productivity of the plant. Using image processing and deep learning techniques to find out specific diseases have a great impact and help to detect the disease and provide restrictions to prevent specific diseases from developing and spreading.

2.2 Plants Leaf Diseases

Each plant disease has various symptoms known by specialists and botany experts, but the symptoms that appear on the leaves are of high important in this work, because most of the symptoms appear on the leaves, which represent the very important part of plants. These diseases are clear leaves symptoms and vary from one disease to another including the form and the colour of leaves as shown in figure (2.1) which represents an example of different phenotypes of tomato plants. There are many known diseases that hit fruits and vegetables such as Bacteria, Fungi, Virus, Mold, and Mite diseases.

These diseases can be defined along with their symptoms and images so that they can be identified and classified accordingly.

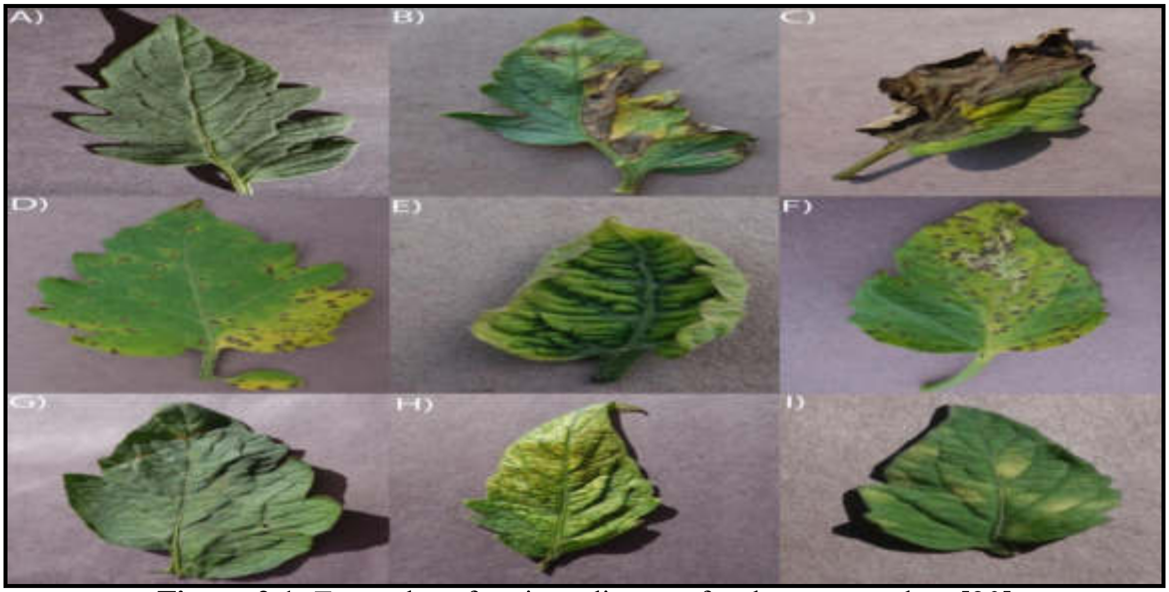


Figure 2.1: Examples of various diseases for the tomato plant [20].

In the above figure 2.1, different phenotypes of tomato plants are shown as following: A represent normal leaf), (B represent Early Blight),(C represent Late Blight),(D represent Septoria Leaf Spot),(E represent Yellow Leaf Curl Virus),(F represent Bacterial Spot,(G represent Target Spot) and (H represent Spider Mite)[20].There are a lot of plants diseases types, especially those appear on the leaves; the common and famous types will be discussed in this work with their symptoms which are used to identify each disease, these types are:

1- Bacterial Diseases

Bacteria are single-celled organisms that are microscopic; by simple cell division bacteria rapidly divide and absorb nutrients from their immediate surroundings. It has many types, but a very famous disease in a lot of plants is the bacterial spot [21]:

• Bacterial Spot (*Xanthomonas euvesicatoria*)

Bacterial spot is a popular disease affected tomato and pepper, etc plants, where levels of rainfall and moisture promote the development of the disease. Through defoliating and spotting fruit bacterial spot decreases the production and activity of plants. Through biological openings or injuries, bacteria enter plants. Leaf spots are dark brown, greasy when the leaves are wet and rarely exceed 1/8 inch in thickness as seen in figure (2.2)[21].

Figure 2.2: Bacterial spot (leaf spots) [21].

2-Fungi Diseases

The fungus infections are the most common group of plants infection. Usually microscopic are these multicellular plants, fungi ' shell ' is made up of filament-like fibers' called hyphae. Paroles of hyphae are called mycelia such masses can be seen without a microscope when they are big enough. Popular fungal symptoms include spots of the leaf wilts, blights, tumors, rots of the fruit, and dieback. Fungi cause a broad range of diseases such as Septoria leaf spot, early blight (Alternaria-solani), target Spot (cassiicola), and tomato leaf mold (Fulvia-fulva) [22].

A. Septoria Leaf Spot (Cercospora capsici)

Septoria is a fungus that causes various leaf spot diseases on field crops, forages and many vegetables, including tomatoes, and is responsible for large crop losses. The symptoms involve chlorotic spots as the following figure (2.3) that show leaf spot disease the leaf turns brown and necrotic [23].

Figure 2.3: Leaf spot caused by Septoria (Cercospora capsici) [24].

B. Early Blight (Alternaria-Solani)

Alternaria solani is a fungal pathogen that causes an illness in tomato and potato crops known as early blight. Early blight signs appear mostly on lower senescent buds, which are excessively chlorotic and abscise as in figure (2.4) [25].

Figure 2.4: Early blight leaf spot on a potato leaf [25].

C. Target Spot (Corynespora Cassiicola)

Corynespora cassiicola is a well-popular pathogenic fungus species. It is a shell fungus in the Corynesporascaceae group. It is the Corynespora genus type classes [26]. The disease is detected by leaf exposure with light centers and dark borders in the style of target-shaped spots. Target spot symptoms on the leaf of tomato plants is seen in (figure 2.5) [27].

Figure 2.5: Target spots symptoms on the leaves of tomato [27]

D. Tomato Leaf Mold (Fulvia-Fulva)

It's a non-compulsory disease that leads to tomato disease named as the tomato leaf mould (P. fulva) [28]. (P. fulva) mainly affects tomato plants, particularly leaves, and is a common illness in greenhouses, but may too occur in the fields. The symptoms of this infection usually occur on the leaves and on the adaxial and abaxial surfaces. It transpires on both sides of the leaf as seen in figure (2.6). The more adult leaves are first affected and then the disease spreads to new leaves [29].

Figure 2.6: Symptoms on Tomato Leaf mold fungus (Fulvia-fulva) [29].

E. Mold

Moulds are a wide and taxonomically complex number of fungal organisms in which hyphae growth contributes to discoloration and appearance, particularly on the food [30]. There are several types of mould, the most famous of which is Late blight (Phytophthora Infestans).

- **Late Blight (Phytophthora Infestans)**

Phytophthora infestans is a water mould that causes severe potato and tomato infection called late blight or potato blight. Early blight, made by Alternaria solani, the potato mark colour is white. On the outside of potato stems and leaves, persons will find Phytophthora infestans develop sporangia and sporangiophores. These sporangia and sporangiophores often occur on the lower surface of the plants, are shown in figure (2.7) [31].

Figure 2.7: Symptom of late blight on the underside of a potato leaf [31].

3-Virus

Viruses are objects smaller than a single cell that can’t be seen by a light microscope. Some viruses transmit by humans, but some are mechanically transmitted by exposure to infected sap from plant injuries. Sampling and feeding behaviors of insects such as aphids, mealy bugs and leafhoppers bearing viruses infect plants in insect transmission [21]. There are many diseases of viral plants, thus will review two important types of them in the tomato plant, namely tomato yellow leaf curl virus, and tomato mosaic virus.

A- Tomato Yellow Leaf Curl Virus

Tomato yellow leaf curl virus (TYLCV) is a begomovirus family and Geminiviridae type DNA virus. TYLCV causes the most damaging tomato disease and severe economic losses can be seen in tropical and subtropical areas. This virus is spread by an insect vector of the Aleyrodidae family and controlled by the whitefly Bemisia tabaci, Hemiptera, commonly known as the silver leaf whitefly. Symptoms of TYLCV disease include extreme stunting, decreased leaf size, outward curling of leaves, chlorosis of leaves and flowers [32], is shown in figure (2.8).

Figure 2.8: Tomato Yellow Leaf Curl Virus [20].

B-Tomato Mosaic Virus

Tomato mosaic virus (ToMV) is a pathogenic virus in plants. It is impacts tomatoes and several other plants, the leaves of infected tomato plants display mottling, with contrasting yellowish and dark green regions, the latter sometimes being deeper and higher, giving a blister-like appearance. The leaves lead to appear fern-like with pointed hints and maybe bent with younger leaves. The virus decreases the collection of fruit in young plants and can lead to imperfections. The whole plant can be eclipsed and the flowers decoloured, shown in figure (2.9) [33].

Figure 2.9: Tomato Mosaic Virus [20].

4-Mites

Mites are microscopic arthropods that relate to the Arachnida class and the Acari subclass [34]. There are several types of mites, the most famous of which is spider mites (Tetranychus urticae).

- **Spider Mites (Tetranychus Urticae)**

Tetranychus urticae (popular title involve two-spotted spider mite and red spider mite) is a plant-fed mite genus that is generally regarded as a pestilence. It is the most common Tetranychidae family member or spider mites. The spider mite is notably small, barely visible as reddish or slightly green spots on stems and leaves with the human eye; grown-up females are about 0.4 mm long. The red mite of spider, its seen greenhouses temperate areas and tropic spinning a little web above and under leaves [35], are shown in the figure (2.10).

Figure 2.10: Tomato Spider mites. [20]

2.3 Digital Image Processing

The image might be characterized as a function of two-dimensional f (x, y), where y and x are locative (plane) organizes, the plentifulness (f) at each couple of directions (x, y) is known as the grey level or intensity image at that point. When (x, y) the intensity portions estimations for the most part limited discrete amounts, it considers the image a digital image. The field of digital images processing alludes to processing digital images by methods for a digital computer. Note that an advanced image is made out of a limited number of contents, every one of which has a specific area f (x, y) one image is worth more than ten thousand information unknown and value. These components are called image elements, panels, and pixels. Pixel is the term utilized most broadly to mean the elements of a digital image [36].

2.4 Typical Structure of a Plants Leaf Diseases Detection Approach

The general structure of plants leaves diseases detection consists of the following main stages:

1. **Image acquisition**: In this stage, images were taken from dataset for plant leaves collected and captured by digital cameras [37].
2. **Image pre-processing**: The primary goal of image preprocessing is an improvement of the image data that contains undesired distortions or enhance some image feature important for additional processing.
3. **Image segmentation**: This technique is used for the conversion of the digital image into several segments having some similarity. Image segmentation assists in the detection of objects and the boundary line of the image [37].

4. **Image post-processing:** It's called this name because it is the process that it performs after one of the important operations. This process and its host arrange and configure data for entry into the next process. The procedures performed by this process vary according to the approach it needs [37].
5. **Feature extraction**: The goal of the feature extraction process is to extract the most important features in the image after the image post-processing and these features are used to determine the specific sample used in the next stage [37].
6. **Classification:** It is considered from the important steps because, from this step, it can classify the diseases according to features that get it from plant leaves image, for classification step used many algorithms like machine learning algorithms. Figure (2.11) views the general structure for detection of plant leaves diseases and classification [37].

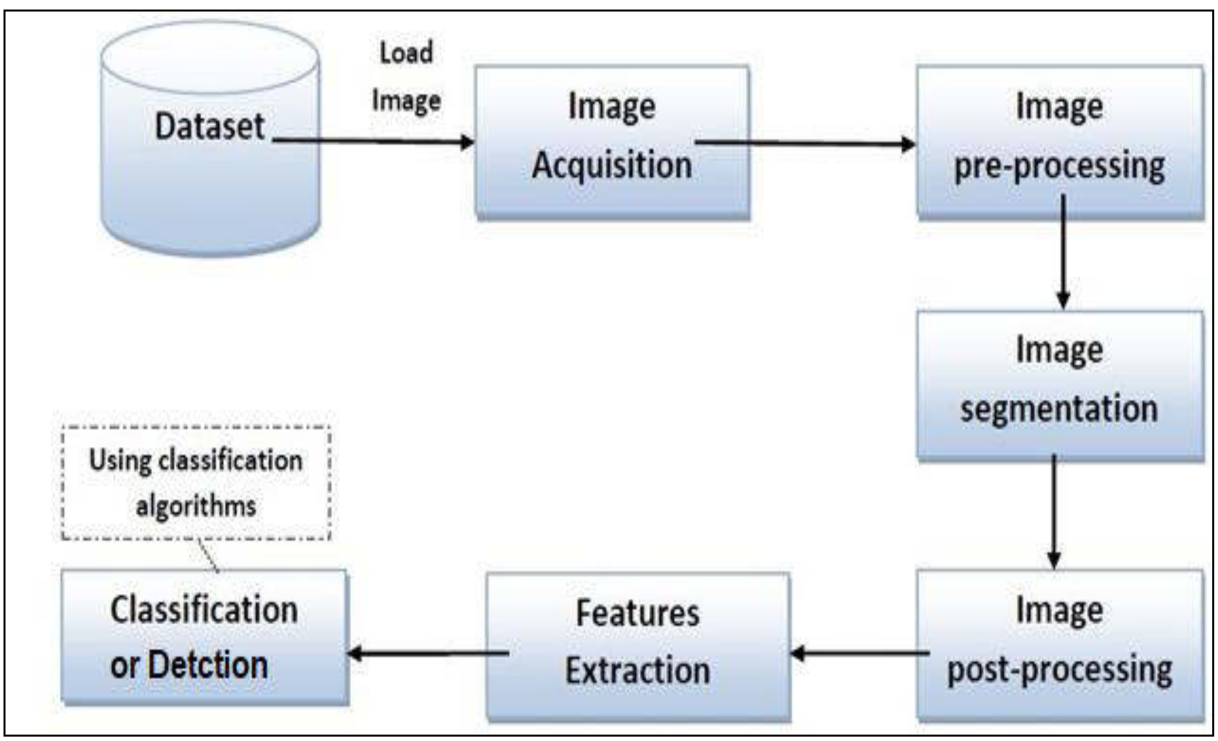


Figure 2.11: General structure of plant leaf diseases detection [37].

2.5 Image Acquisition

In image processing, it is characterized as the activity of retrieving an image from some source, generally a hardware-based for processing. It is the initial stage in the work process grouping, because, without an image, no processing is conceivable; the image that is gained is natural [37].

2.6 Image Pre-Processing

The image pre-processing is one of the basic stages in image processing. It improves image information by noise, the background removing and furthermore stifling of unwanted deformations. The improvement of image characteristic for investigation and processing also may image are resized from size to another [38]. An example of pre-processing is the process of converting from one colour space to another.

2.6.1 Conversion from RGB to L*A*B* Colour Space

This conversion is needed in order to show the images more, especially the things that depend on the colour illumination. The three L*A*B* dimensions reflect the highlight of the colour (L*=100 and L*=0 yields black gives speard white; specular stay higher at white), its location between green and red/magenta (a *, contrary estimate show the positive of green while amounts get magenta) and that location between blue and yellow (b *, negative amount signify blue and positive amount signify yellow correlative ranges from 0 to 100 are shown in figure (2.12).

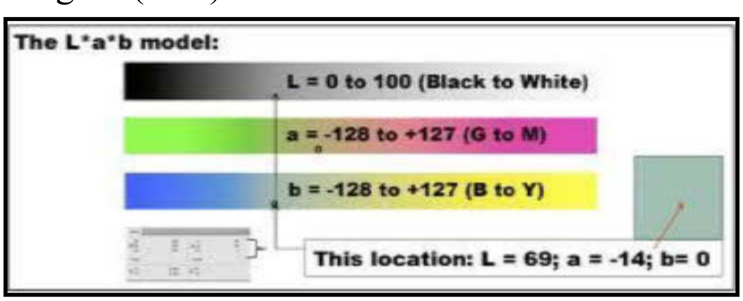


Figure 2.12: The L*a*b* design [39]

The disparity in the L*a*b* colour space between the two points is like the human visual system, because the model L*a*b* is a three-dimensional pattern that fully designed in the space three-dimensional. The following equations appear the solution to transform digital images from RGB space to L*a*b* colour space [39].

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} 0.608 & 0.174 & 0.201 \\ 0.299 & 0.587 & 0.114 \\ 0.000 & 0.066 & 1.117 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix} \quad \ldots\ldots\ldots\ldots\ldots \quad (2.1)$$

CIEXYZ's colour space coordinates are X, Y, Z, Xn, Yn, and Zn. The approach from the RGB space to the CIEXYZ colour space to transform digital images is as in the following equations with the colour boundaries for x [39]:

$$f(x) = \begin{pmatrix} X^{\frac{1}{3}} & x > 0.008856 \\ 7.787x + 16/116 & x <= 0.008856 \end{pmatrix} \ldots\ldots\ldots\ldots \quad (2.2)$$

$$\begin{bmatrix} L* = 116\, f(Y/Yn) - 16 \\ a* = 500[f(X/Xn) - f(Y/Yn)] \\ b* = 200[f(Y/Yn) - f(Z/Zn)] \end{bmatrix} \ldots\ldots\ldots\ldots \quad \ldots\ldots\ldots\ldots (2.3)$$

Where Xn, Yn, and Zn are respectively corresponding to the white value of the parameter. Conversion of colour space is the transfer of colour reproduction from one basis to another. It typically occurs by transferring an image to another colour space that is depicted in one colour space. The effects after the transformation of colour space are seen in the figures (13, 14) [39].

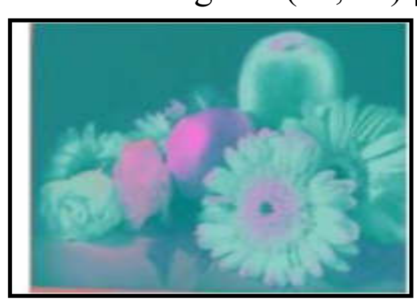


Figure2.13: RGB image colour space **Figure2.14**: The image after conversion to L*A*B*

2.7 Image Segmentation

According to the various processing image techniques, the segmentation image assumes an indispensable job in stage to do the given image will be analyzed. The segmentation image is the essential stage to images dissects and the information extract from it. This project bargains about essential standards of the techniques by the segment of the image. The segmentation has gotten an unmistakable goal in computer vision and image analysis. The segmentation methods of images like edge detection thresholding, clustering and region growing are account for this investigation. The segmentation algorithms depend on two characteristic discontinuity and similarity. Here must highlight on the technique that is broadly utilized to image segment [40].

2.7.1 Segmentation by Feature Based Clustering

Clustering is a step of division the gatherings according to their attributes. A cluster as a base includes a gathering of matching pixels that has a sit with a particular region and not the same as different areas. The term information clustering as equivalent words like cluster analysis, automatic classification, numerical taxonomy, and typological analysis. Images are gathered according to their substance. The techniques of clustering are generally sorted as partitional algorithms and hierarchical algorithm. In content-based hierarchical, the gathering is done dependent on the acquired features of the pixels example texture, shape etc. There are various clustering techniques use the most generally by the K-means algorithm [40]

2.7.2 K-Means Clustering Algorithm

K-Means algorithm is an unsupervised learning algorithm. It's one of the prevalent techniques. The k-means clustering, it parcels data an assortment

into group's data of k number [41]. It arranges given information of arrangement into k number of cluster disjoint. K-means algorithm involves two separate parts.

In the first step, is account the k centroid and in the second step, it get every point to the cluster that has the closest centroid from the separate point data. That's have various techniques to define closest centroid is the separated and one of the most utilized techniques is Euclidean separation, when the grouping is finished it recalculates the new centroid of any cluster and up on that centroid. Another Euclidean space is determined between any middle and each data point and the points a signs in the cluster that have little Euclidean space. Every cluster in the parcel is defined use the objects part and its centroid are seen in figure (2.15) [42].

The point of centroid for any cluster that the aggregate distance from every object that make the minimized of cluster. So K-means is an algorithm iterative, which minimizes the sum of the distance from any item to its centroid cluster, total clusters. Give us a chance to think an image with a resolution of x * y and the image should be cluster into cluster of k number. Let p(x, y) be Ck be the cluster centers and an information pixel to be cluster. The k-means clustering algorithm is summarized in the following steps [42]:

1. Begin a number of cluster center and k.
2. The image of all pixel, account the Euclidean distance (ED), between the center and every the image of pixel utilizing the equation given below. Center and every pixel of an image utilizing the equation (2.4) [42] given below.

$$ED = \sqrt{(x_1 - c_1)^2 + (x_2 - c_2)^2 \dots \dots + (x_n - c_k)^2} \dots\dots\dots\dots\dots\dots \quad (2.4)$$

3. Allocate all the pixels to the nearest center depended on ED.

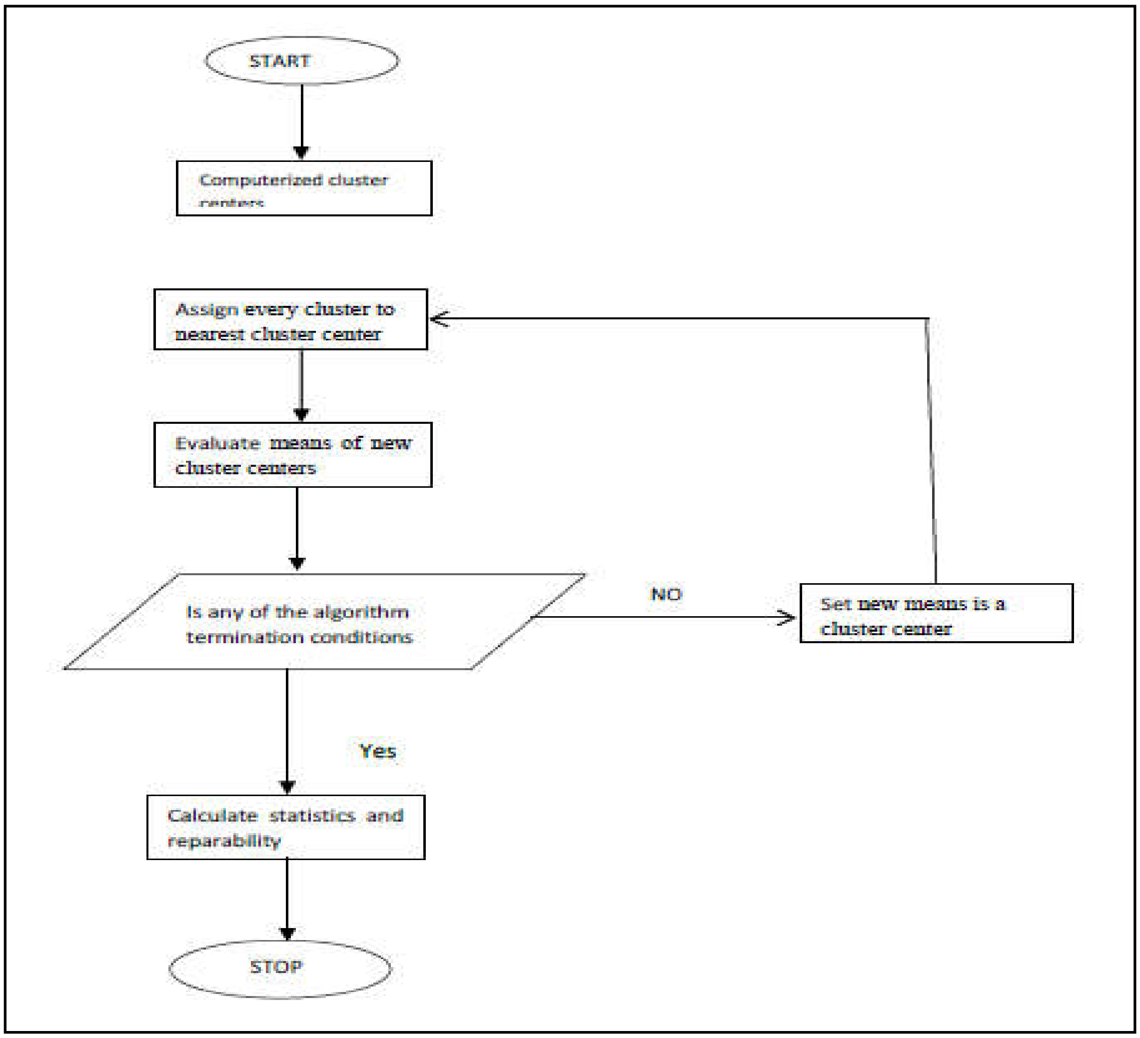
4. After all, pixels have been assigned; recalculate the new position of the center by calculate the mean utilizing the equation (2.5) [42] given below.

$$Ck = \frac{1}{k} \sum_{y \in Ck} \sum_{x \in Ck} P(x, y) \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ (2.5)$$

Where Ck represented the cluster centers

5. Repeat the process unto it satisfies the tolerance or error value.

6. Reshape the cluster pixels into the image.



To facilitate the K-Means algorithm here will show it in the following flowchart in figure (2.15) [43].

Figure 2.15: Scheme of K means clustering algorithm [43]

Example:

In this example an image it has several operations to segmentation. In first Labeled image using cluster index the then establishes three cluster areas of the image is shown in the following figures:

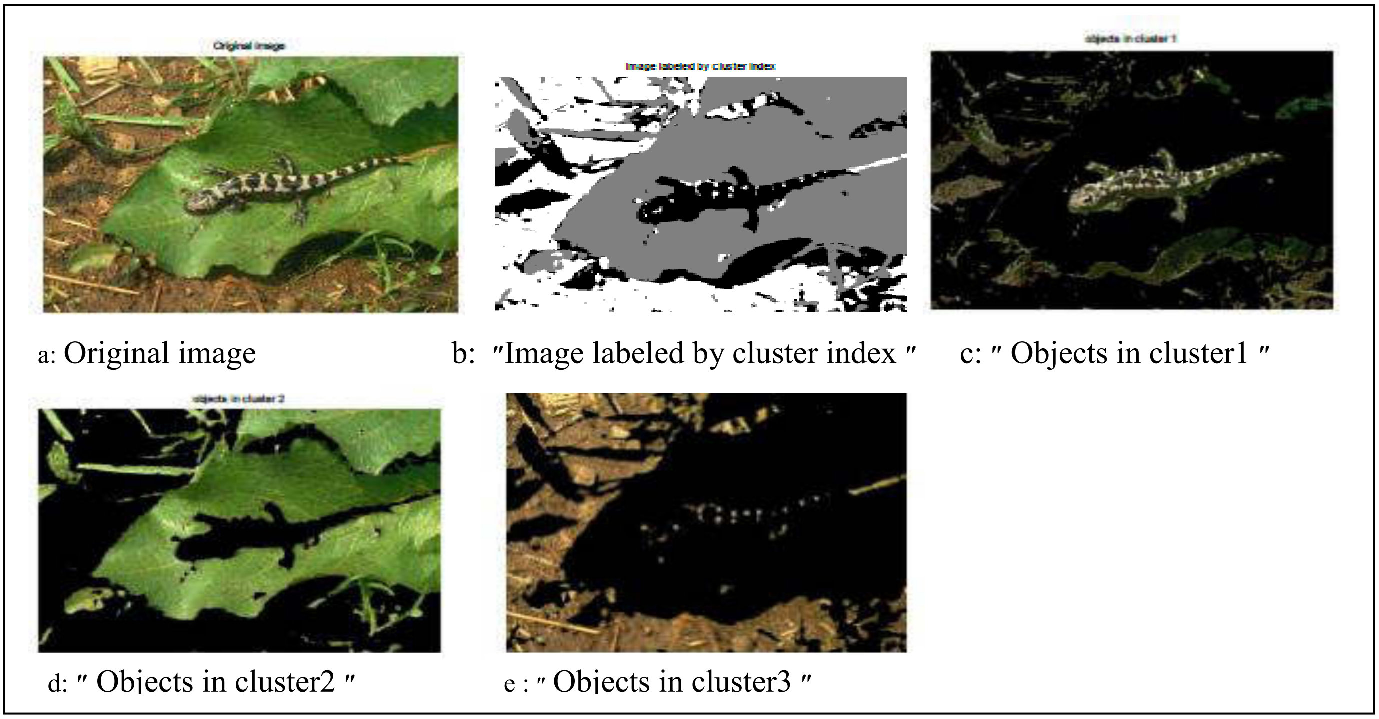


Figure 2.16: The segmentation process by K-mean clustering algorithm [44].

In figure (2.16) **a** represents the original image, **b** shows labeled image using cluster index (**c**, **d**, and **e**) displays the cluster objects one , two and three respectively [44] .

Although k-means has the advantage of being easy to implement, it has a few downsides. The quality of the last clustering outcomes relies upon the subjective choice of the initial centroid. So if the initial centroid is arbitrarily picked, it will get a diverse outcome for various initial centers. The initial centers will be intentionally picked with the goal that gets need segmentation.

Furthermore, computational complexity nature is another term which has to reflect while structuring the K-means clustering. It depends on the number of information components, amount of cluster and number of iteration [43]. Anyway segmentation image is field advance for realizing and image analyzes that is a usually way to gave semantic high level [45].

2.8 Image Post-Processing

This process always enhances or modifies or smoothes images to make them look better and prepare them for the next stage. Some of the possible post-processing operations are converting the image from RGB to gray-scale [36].

2.8.1 Conversion from RGB to Gray-Scale Image

Most of digital image file formats (DICOM [Digital Imaging and Communications in Medicine] digital cameras, slide scanners, or flatbed scanners) spare images as RGB (red, green, blue) records. The corresponding images are comprised of three different channels, per channel including data on the images' red, green or blue sections. Converting RGB pictures to grayscale eliminates colour from the image without content reduction, which can be used to extract colour emitted from a digitally scanned or photographed image. The RGB colour values are expressed by the properties of lightness, chrome, and hue in three dimensions XYZ. The output of a colour image based on the number of bits that could be assisted by the digital device [46].

The main advantage of translating an image from colour to grayscale domain is that to receive fewer data because the grayscale domain has a single channel preferably than three RGB domain channels, per colour pixel is

represented by a triple (RGB) of red, green, and blue depths transformed to grayscale is a set of shades of gray without visible colour. Each pixel value in the gray image is a representation of 256 values; more precisely, the black colour is marked by the value 0, where white colour is marked by the value 1 and the residual values are standard black and white shades as in the following equation [47]:

$$\text{Gray Scale}\,(i,j) = \big((0.3 * R) + (0.59 * G) + (0.11 * B)\big) \ldots\ldots (2.6)$$

The R represents red, G= green and B=blue.

2.9 Feature Extraction

Features extraction usually is done after the pre-processing stage and segmentation stage. The essential assignment of model recognition is to take an information design and effectively dole out it as one of the conceivable output classes. This procedure can be isolated into two general stages: Feature selection and classification. Feature selection is basic to the entire procedure since the classifier won't have the option to perceive from inadequately selected features. Extraction feature is a significant progress in the evaluation of each model classification of the viable data that portrays of every class. The procedure indicates that important features are released from objects to vectors of feature. These vectors of feature are utilized to see the classifiers information unit with the objective yield unit. It gets a classifier simpler to series between various classes by pick the features gander. Feature extraction is a procedure retrieves the ultimate significant information from the original data. The finding arrangement of the element when characterizes state of an element furthermore, remarkably [48].

The objective significant of the extraction feature is to evolve a lot of features, which expands the rate of recognition with the least sum components and create a similar feature set for a variety of instance for an assortment of an example of a similar image. The generally utilized feature extraction strategies are Gradient feature, Template matching, Graph description, Deformable templates, Zoning, Histograms, Fourier descriptors, Contour profiles, Unitary Image transforms, Geometric minute invariants, and Gabor features etc [48]. This introduction summarizes that feature extraction is a data minimization procedure to discover a subset of useful image dependent variables that are used and relied on in the data classification process. That will mention here some types of features and some methods used to extract them. Images are normally represented as shape, colour, and texture features. The most famous are colour and texture features.

2.9.1 Colour Feature

In consideration of its robustness, usefulness and procedural naturalness, colour is one of the most commonly used tools of image retrieval. The image colour is depicted by a few colour models. The regularly utilized colour models are RGB (Red, Green, Blue), HSV (Hue, Saturation, Value) and Y, Cb, Cr (luminance and chrominance) thus for any colour picture the colour content is described by 3-channels from some colour model. The colour features can be depicted by the colour histogram, colour correlogram and the colour moment [49, 50].

2.9.2 Texture Feature

The texture is an important depiction of an immense arrangement of images. It is a helpful representation of the image's wide scale. It has been acknowledged, for the most part, that systems of human visual by texture for

interpretation. In most ways, a pixel property is coloured whereas texture must assume from a pixels group. Countless methods to extract the texture features were proposed in light of the domain when the texture feature is extracted, it's extensively grouped into spatial extraction strategies of texture feature and extraction techniques of spectral texture feature. Texture feature is taken away using registering the statics pixel or searching the nearby pixel construction in genuine image space, though last transform image into the domain frequency and afterwards computes feature from the image is transformed. The texture is represented by two kinds of essential methodologies that are utilized in image processing which it spectral methodology and statistical methodology. A statistical methodology for texture portrayal texture represented in this methodology depends on the distribution of pixels and the connection between image pixels [36].

2.9.3 Statistical Based Features

The statistical in this project are classified into the first and second-order statistic [48].

2.10 First-Order Features

These features usually depended on the colour, some of these features are:

1. **Mean**

Describes this feature the medium value, providing information for the total image pixels, it expresses in the equation (2.7) below [51]:

$$\mu = \frac{1}{N}\sum_{j=1}^{N} P_{ij} \quad \text{……….………………….………… (2.7)}$$

While N represents the number of pixels in the image and (P_{ij}) is the value of the j-th image pixel in the i-th colour channel.

2. **Standard deviation**

The standard of deviation is additionally named the contrast square root, viewing the image contrast. It displays the spread of information the picture with a rising standard deviation and has an elevation contrast ratio. The standard deviation (SD) is described in equation (2.8) [51].

$$(\sigma)\ or\ (SD) = \sqrt{\left(\frac{1}{N}\sum_{j=1}^{N}(P_{ij} - \mu)^2\right)} \qquad \ldots\ldots\ldots\ldots (2.8)$$

3. Entropy

The entropy is a measure that discloses to what number of bits it needs to code the image information, it is represented by the following equation (2.9) [51].

$$\text{Entropy} = -\sum_{i=0}^{N-1} P_i(x) log_2\ [P_i(x)] \ \ldots\ldots\ldots\ldots\ldots (2.9)$$

Where P_i is the probability associated with image pixels.

4. Root Mean Square (RMS)

The Root Mean Square (RMS) calculates the amount of every column or row of the insert overall vectors of a predetermined dimension of the insert or of all insert. The Root Mean Square amount to j column of an M- by -N insert matrix u is obtained using the equation (2.10) below [52]:

$$RMS = \sqrt{\frac{\sum_{i=1}^{M}\left|u_{ij}\right|^2}{M}} \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (2.10)$$

Where M is the number of samples

5. Variance

Variance is the desire of the squared deviation of a random variable from its mean. Casually, it measures how far a lot of (random) numbers are spread, out from their average value. Variance is the square of standard deviation. The equation for obtaining the Variance is (2.11) [52]:

$$Variance = (SD)^2 \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (2.11)$$

(SD) represented the Standard Deviation

6. **Kurtosis**

Kurtosis depicts the state of probability dissemination, and similar to skewness, there are various methods for measuring it for a hypothetical distribution and comparing methods for evaluating it from a sample from a population. It is represented in equation (2.12) [51].

$$\text{Kurtosis} = \sqrt[4]{\frac{1}{N}\sum_{j=1}^{N}(P_{ij}-\mu)^4} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots \ (2.12)$$

7. Skewness

It aims to measure how asymmetrical the colour distribution is and thus gives details on the shape of the colour distribution. Its equation (2.13) represented below [51].

$$\text{Skewness} = \sqrt[3]{\frac{1}{N}\sum_{j=1}^{N}(P_{ij}-\mu)^3} \quad \ldots\ldots \ (2.13)$$

2.11 Second-Order Features

The first-order statistics of features produced obtain data introduced with the gray level appropriation of image. It's may that don't get any data concerning the overall positions of the diverse gray levels into the picture. These features shan't have the choice to calculate whether total minimum-value gray levels are placed together, or it is exchanged with the minimum-value gray levels. Some event of gray level design must be depicted using a matrix of frequencies relative Pθ, d (I_1, I_2). It how depicts frequently two pixels with gray levels I_1, I_2 appear in the window isolated by a distance d in direction θ [53]. There are more, methods for computing second-order features such as Gray-Level-Co-Occurrence-Matrix (GLCM).

2.12 Gray-Level-Co-Occurrence-Matrix (GLCM)

GLCM focuses on co-occurrence matrix gray level. It maps the pixel brightness of a picture which takes place. It was offered by Harlick in 1970. It is additionally called Gray tone Spatial Reliance Matrix (GSDM). GLCM

texture gets the connection between two pixels one after another, called the reference and the neighbor pixel. A matrix is developed at distance d=1 and at angles in degrees (0, 45, 90,135) as shown in figure (2.17). GLCM clarifies the distance and angular spatial relationship over a picture sub-region of a particular size. It is prepared from grayscale values are shown in figure (2.18) explained the GLCM generating process for angle 0 with degree 1. It is considered how regularly a pixel with gray level (gray scale intensity or gray tone) values come either vertically, horizontally and diagonally to leveled the pixels with the value j Harlick additionally offered various measures for example entropy, vitality, differentiates relationship and so on. These measurements at various angles and the figure (2.18) which explained an example the GLCM generating process for angle 0 with degree 1. GLCM direction are: Horizontal (0) Vertical (90) Diagonal a) bottom left to upper right (- 45) b) top left to base right (- 135) .They are declared as P0, P45, P90 and P135 respectively [54].

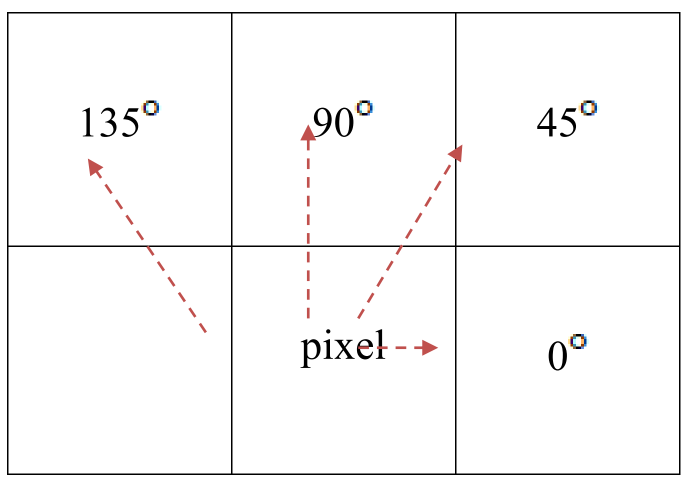


Figure 2.17: Angle direction in GLCM.

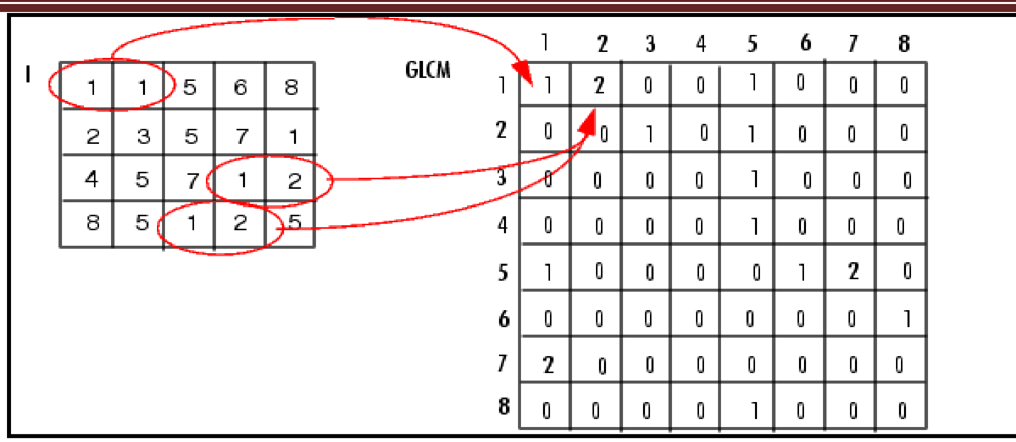


Figure 2.18: Process used to generating GLCM.

Harlick proposed many features textures are determined the probability matrix up on the matrix co-occurrence [55], some of these feature:

1. Contrast

The using this feature to estimate the contrast picture by the differences of the local measure, if the image has a high contrast of, the rate of contrast will be high as the equation viewed (2.14) [56].

$$Contrast = \sum_{i=0}^{N_g-1} \sum_{j=0}^{N_g-1} (i-j)^2 P(i,j) \quad \text{.......... ....} \quad (2.14)$$

Here(i , j) the-spatial assimilate co-ordinates of the function P (i, j) the GLCM matrics, where (N_g) represent the gray tone.

2. Correlation

Correlation is an important feature since they can show a predictive relationship that can be employed in practice. This indicates statistic the proportion of gray linear tone displays in the image, its equation (2.15) [57].

$$Correlation = \sum_{i=0}^{N_g-1} \sum_{j=0}^{N_g-1} \frac{(i-\mu_i)(j-\mu_j)}{\sigma i \sigma j} \quad \text{.................} \quad (2.15)$$

3. Energy

It has different names, such as uniformity or Angular Second Moment (ASM), which can be characterized as the total of the squares of entries in the

GLCM. It estimated homogeneity of the image. When the value is high, the image has high homogeneity. It represented by equation (2.16) [56]:

$$Energy = \sum_{i=0}^{N_g-1}\sum_{j=0}^{N_g-1}(P_{ij})^2 \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (2.16)$$

4. Homogeneity

Homogeneity or Inverse Difference Moment (IDM) is a feature measure the picture's homogeneity, its value being high at the point when the picture elements are similar. There is an inverse association with the contrast, implying that when the contrast value is high, its value is low, as in equation (2.17) [56]:

$$\text{Homogeneity} = \frac{\sum_{i=0}^{N_g-1}\sum_{j=0}^{N_g-1}P(i,j)}{1+(i-j)^2} \quad \ldots\ldots\ldots\ldots\ldots\ldots \quad (2.17)$$

2.13 Classification of Plant Leaf Diseases

Classification of image alludes to this images labelling into one of the numbers of previously classes. The type of image is most significant as a basic stage, for the processing high-level. The classifier is utilised for classifying images depending on their features. Classification is a significant module in plant diseases detection structures. plant leaves detection approach that recognize plant diseases utilizing an image, thus classification here is characterized as a procedure of categorizing plant leaf images dependent on distinguished diseases [58]. In the proposed approach, the machine learning algorithms used to classify and detect plant leaf diseases. Here will talk about machine learning, its types, and some algorithms that will be used.

2.14 Machine Learning(ML)

Machine learning is a subset of artificial intelligence, that by re-iterative procedures allows a computer system to derive from the environment and improve itself. Machine learning methods sort out the information gain from it gather insights and make predictions reliant on the data analyzed without the

need for additional programming that is definite. ML is utilized to teach machines how to deal with the information all the more efficiently. Some of the time after viewing the information, it can't interpret the pattern or concentrate data from the information. All things considered, it applies ML [59]. With the wealth of datasets accessible, the interest in machine learning is in the ascent. Numerous ventures from medication to the military to agriculture apply ML to separate significant data and classification for that data [60]. ML is divided into several common types as follows:

1. Supervised Learning:

In this type, the algorithm produces a function that plans the inputs to wanted outputs. The first standard definition of the learning task supervised is divided issue the learner is wanted to learn a function that vector maps change to one of the various types use looking at the function of manyinput_output cases [61].

2. Unsupervised Learning

It is used to draw conclusions from datasets comprising of input information without labelled reactions. The most well-known unsupervised learning strategy is cluster analysis, which is utilized for exploratory information investigation to discover hidden patterns or grouping in the information [61].

3. Semi-Supervised Learning

Its merges both labelled and unlabeled patterns to produce a suitable function or classifier.

4. Reinforcement Learning

In this type, the algorithm learns a pattern of acceptable behaviour given a perception of the environment. Each action has some effect on nature, and the environment gives input that aides the learning algorithm.

5. Transduction

Like supervised learning, however, doesn't unequivocally generate a function rather attempts to predict new outputs dependent on training inputs, training outputs and the new inputs.

6. Learning to Learn

In this type, by relying on the previous experience the algorithm learns it's own inductive [61].

The proposed approach used the first type of machine learning (supervised learning) that will clarify and some of its algorithms that have been used in detail.

2.15 Supervised Learning

The machine learning assignment of a function learning a maps an input to an output dependent on model output-input.[62] the function derives from training labelled information comprising of many examples of training. Every model is a couple comprising of an insert object (vector commonly) and the ideal output amount (additionally called the ″supervisory signal″). An algorithm analyzes of supervised learning down the training produces and information a function deduced that utilized for modern mapping models. Ideal situation will be consider algorithm to effectively decide the labels type for concealed instances. This wanted the algorithm learning of sum up from the training information concealed circumstances at "sensible" way [63].

Among the most important algorithms in the field of supervised learning is a support vector machine (SVM) algorithm and also within the deep learning algorithm that will be explained with its algorithms in the most important of which are the Artificial neural network(ANN) and the convolutional neural network(CNN), but before that the SVM algorithm will be detailed.

2.16 Support Vector Machine (SVM) Algorithm

The algorithm was invention in (1995), introduced this algorithm to resolve the kind of specific regression and classification problems. This method is based on the learning statistical method [64]. Its work to determine the optimal hyperplane that maximizes the classes' margin. SVM has become one of the classification's most popular methods. It is one of the supervised classification methods in which the inputs series with their labels is given. The structures defined by the vector features of these inputs. This algorithm produces a hyperplane distinguishing the two groups in order to achieve total separation between these classes. In figure (2.19) there are two vectors parallel to the classifier that passes via several stages. The distance between these two parallel vectors is called the (margin). The points on boundary vectors are called support vectors [65].

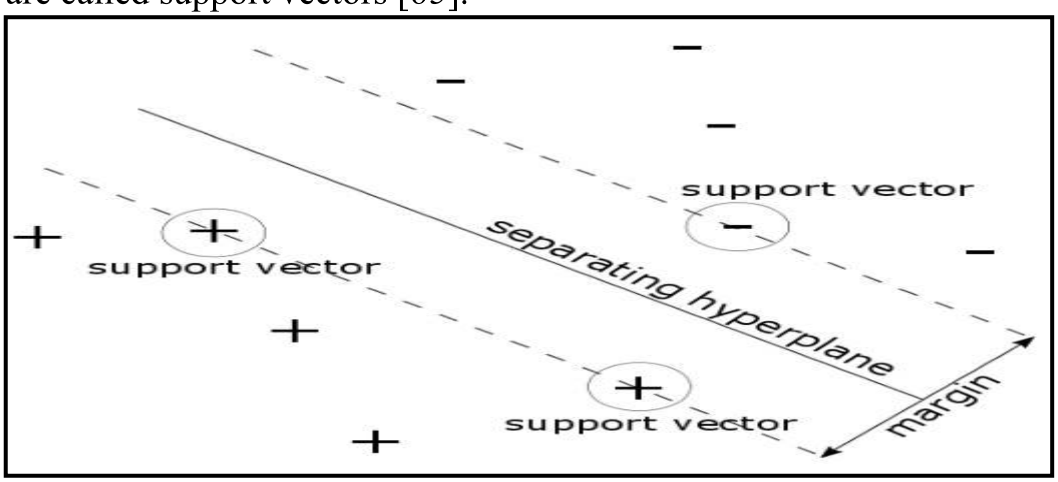


Figure 2.19: The SVM hyperplane between two classes [66]

The hyperplane is illustrated in figure (2.19) the separating data points from two groups as + and -. The vectors and margin of support are labeled.

SVM algorithm is divided into two kinds linear and non-linear SVM, the non linear SVM illustrate in the following section:

2.16.1 Non-Linear SVM

In most cases, the linear classification does not consider the appropriate classification approach for the non-linear classification used in such situations, where a non-linear kernel function will be used. Linear SVM are fast to train and implement, but with many training examples and not too many features they appear to underperform on complicated datasets [67]. In many applications, non-linear SVM can be more consistent for quality across different problems and the preferred choice, although they lack critical power [68].

A- Kernel Function

To move testing samples and training to feature space high-dimensional feature space, functions kernel are implemented. This section, functions kernel are described to replace functions of mapping because the kernel computation is most effective than the function of mapping, computation period usually saved to exchange functions of mapping with the used of kernels [65]. SVM implements the kernel function, K (x_n, x_i), which transforms the raw data space into a higher-dimensional new space. It process involves the dot product transformation function ϕ(x) (Equation (2.18). The target is the information that can be feasibly extracted, that has been converted into a higher dimension. It is possible to write the hyperplane function in formula (2.19) [69].

$$\mathrm{K}(\mathrm{x_n}, \mathrm{x_i}) = \phi(x_n)\phi(x_i) \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (2.18)$$

$$f(x_i) = \sum_{n=1}^{N} \alpha_n y_n \mathrm{K}(\mathrm{x_n}, \mathrm{x_i}) + b \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots.. \ (2.19)$$

Where α_n is a lagrange multiplie, x_n is support vector information, and y_n is a membership class label (+1, −1) with n = 1, 2, 3,……., N,[68]. In this analysis, the three commonly used functions of the kernel at the linear, polynomial, and (RBF) Radial Basis Function, it's calculating in the following equations [69]:

1. **Linear** $\mathrm{K}(\mathrm{x_n}, \mathrm{x_i}) = (\mathrm{x_n} \cdot \mathrm{x_i})$ …………… (2.20)
2. **Polynomial of degree d** $\mathrm{K}(\mathrm{x_n}, \mathrm{x_i}) = (\mathrm{x_n} \cdot \mathrm{x_i})^d$ ……….. (2.21)
3. **Radial Basis Function (RBF)** $\mathrm{K}(\mathrm{x_n}, \mathrm{x_i}) = exp \frac{-\|(\mathrm{x_n} - \mathrm{x_i})\|^2}{2\sigma^2}$ ... (2.22)

B- Multi-Class SVM

The main requirement of the support vector machine algorithm was prepared to distinguish two classes, but there are conditions in life that need to more than two classes to be classified. Multiclass classification issues (k > 2) are usually broken down into a set of binary issues so that generic support vector machine can be applied directly. One-versus-rest (1VR) [71] and one-versus-one (1V1) [70] approach are two representative organization schemes for multi-class SVM. Both (1VR) and (1V1) are individual cases of the Error-Correcting Output Codes (ECOC) which disintegrates the multi-class problem into a predefined set of binary issues.

1. One-Versus-Rest (1VR)

The (1VR) strategy for k-classification constructs separates binary classifiers. The binary classifier m-th is learned to utilize the m-th type information as examples confirm and negative examples are k − 1 the remaining types. The binary classifier, which gives the greatest output value, specifies the class label during the test. The unbalanced learning set is a main issue of the (1VR) strategy. Assuming that total types have an equivalent

number of examples training, in every individual classifier, the rate of plus to minus examples is 1k−1. This scenario, the initial problem's balance is missed [71].

2. One-Versus-one (1V1)

Another classical strategy to multi-type of classification is to decompose (1V1) or pair wise [70]. It tests all potential pair wise classifiers and therefore causes individual binary classifiers k (k − 1)/2. Introducing every classifier to an example of a test would give the having won class one vote, the best test is labeled with greatest votes for class. Each size of the classifiers generated by the (1V1) almost is greater than the against-one of rest strategy. The measure of QP is smaller in each classifier, however, making it possible to train quickly. However, the (1V1) strategy is more symmetrical compared to the one-versus-rest method [72].

2.16.2 Advantages VS Disadvantages of SVM Algorithm

A_ Advantage [73]

1. It is marked with its high accuracy especially when used to classify two classes.
2. In this algorithm, over fitting is less important and is considered strong to noise.

B_ Disadvantages of SVM Algorithm [56]

1. A time that consumes
2. It is designed to solve two classes problems if have more than two classes, the accuracy decreases sometimes, and in this case, that must use 1VR solutions or other methods
3. It is computationally hard and expensive.

2.17 Deep Learning

Deep learning (hierarchical learning or deep machine learning) is a department of machine learning based on specifics of algorithms that aim to model high-level data abstractions using multiple layers of complex processing with systems or otherwise consisting of multiple non-linear transformations [74]. An analysis (e.g. an image) can be depicted in many ways as a vector of intensity values for every pixel, or as a set of edges, areas of a particular shape, etc in a more abstract manner. Many characterizations make learning tasks easier (e.g. recognition of plants leaves diseases, recognition of the face, or recognition of facial expression) [75]. So, it is a term that includes a specific approach to neural network construction and learning. Since the 1950s, neural networks have existed and, such as nuclear fusion, they have been an extremely ambitious laboratory concept whose practical application has been constantly delayed.

They take a number matrix (which can represent pixels, word, or shapes of audio). On that set encompass a sequence of functions and output one or even more numbers as outputs. The outputs are typically a prediction of certain properties that you are trying to determine from the data, such as whether or not an image is a cat picture [76]. Deep learning techniques are dependent on distributed representations. The fundamental theory that underlies distributed representations is that experimental information is generated by layer-structured factor correlations. Deep learning incorporates the assumption that the levels of abstraction or structure apply to these component layers. Various layer numbers and sizes of layers can be used to provide various abstraction products [77].

Deep learning uses this concept of explanatory hierarchical variables in which higher-level, more abstract principles are learned from lower-level ones such architectures are often designed using a greedy model of layer-by-layer. Deep learning is attempting to isolate these concepts and to determine what features are useful for learning. Its techniques eliminate function engineering in supervised learning activities by transforming the data into compact intermediate formats close to main components and obtaining layered architectures that remove consistency in representation. Deep learning algorithms are many and the most famous of these algorithms are Artificial Neural Networks (ANNs) and Convolution Neural Network (CNN) [77].

2.18 Convolution Neural Network (CNN)

CNN is mean the Convolutional Neural Networks is the most popular approaches in deep learning in which have many layers are robustly trained [78]. It has been observed to be extremely effective and is additionally the generality by various applications of vision computer. The general structure of the convolutional neural networks system is viewed in figure (2.20) [79].

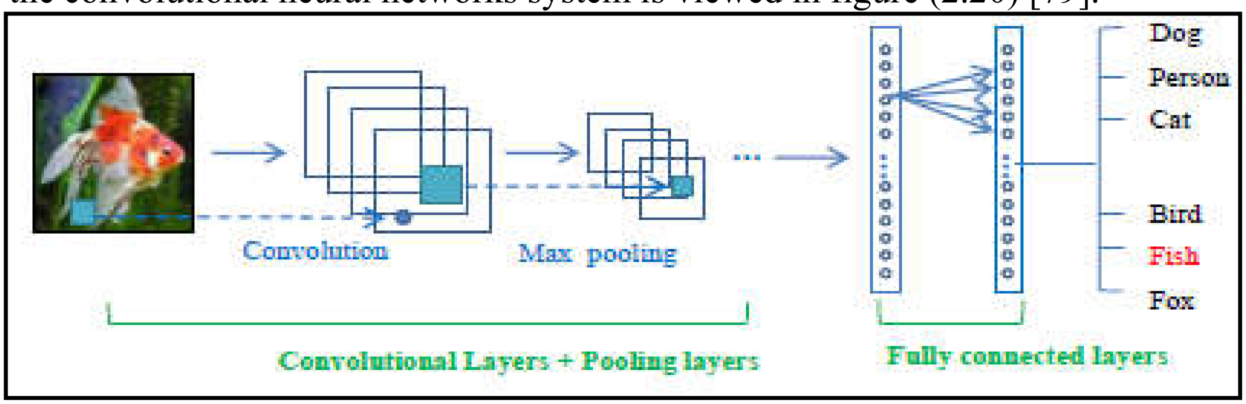


Figure 2.20: The general structure of the CNN system [80].

Commonly, CNN consisting of three basic neural layers, convolutional layers, pooling layers, and layers that are fully connected. Various types of slices take various roles. The figure 2.21 is showed a general convolutional

neural networks structure for classification of image is presented layer by layer [80]. For train the network, there are two stages: forward and backward. The forward stage's main objective is to specify the input image in every layer with the proposed parameters (weights and bias). Later the output of the prediction is used with the ground of the truth labels to calculate the estimate of loss. Second, relative to the cost of loss, the backward stage determines with chain rules the gradients of per parameter. All parameters are modified and prepared for the next forward calculation dependent on the gradients. After enough variations of the stages forward and backward, the learning network can be ended [79].

CNN and traditional Neural Networks (NNs) have major differences; the best is CNN, due to these differences, which are among the advantages of CNN:

1_ CNN input can handle either 3dimensions or 2D and 1D sizes focused on configurations and parameters, while NN input is just a 1D array

2_ CNN lists more layers than regular Neural Network.

2.18.1 Basic Structure of CNN

Convolution neural network is identical to artificial neural network, as both are composed of self-optimized neurons, introduced by inputs and executing non-linear transformation. Convolution neural network is commonly utilized in pattern recognition on objects relative to artificial neural networks, because it encodes image-specific features in system architecture. In the convolution neural network, there are five fundamental elements: the input layer, convolution layer, non-linear layer, pooling layer and fully connected layer [80].

1. Input Layer

The input layer contains the pixel values of the image that enter CNN.

2. Convolution Layers

A CNN uses different kernels in the convolutionary layers to convert the entire object as much as the optimal maps of feature, creating different feature maps [79].

In another meaning a flip and gathering over the kernel. This is extended to two dimensions [82]:

$$S(i,j) = \sum_m \sum_n I(m,n)F(i-m, j-n) \quad \text{..........} \quad (2.23)$$

In another meaning flip the filter (F) up-down and left-right and sum over all products are shown in an example about convolution layer in figure (2.23).

The convolution method has three key advantages [82]:

1) The weight distribution functions in a similar map of feature to decrease the number of parameters.

2) Spatial learns communication associations between nearby pixels.

3) Invariance to the object's position.

The primary idea is that the conventional replacement of convolution layer with a tiny multilayer softmax made up of completely many connected layers with functions nonlinear, substituting linear filtrate with neural networks nonlinear. This approach produces best results when classifying images.

An Example of Convolution Operation

Figure (2.21) shown a, b, c convolution example process with a kernel measurement of [3×3], with no padding and one stride, which the stride means the step scale to vertically and horizontally traverse. The kernel is extended around the insert tensor, and the component-wise function between each

object of the tensor input and kernel is measured at every point and averaged up to produce the output amount in the related direction of the tensor output called the map feature [83].

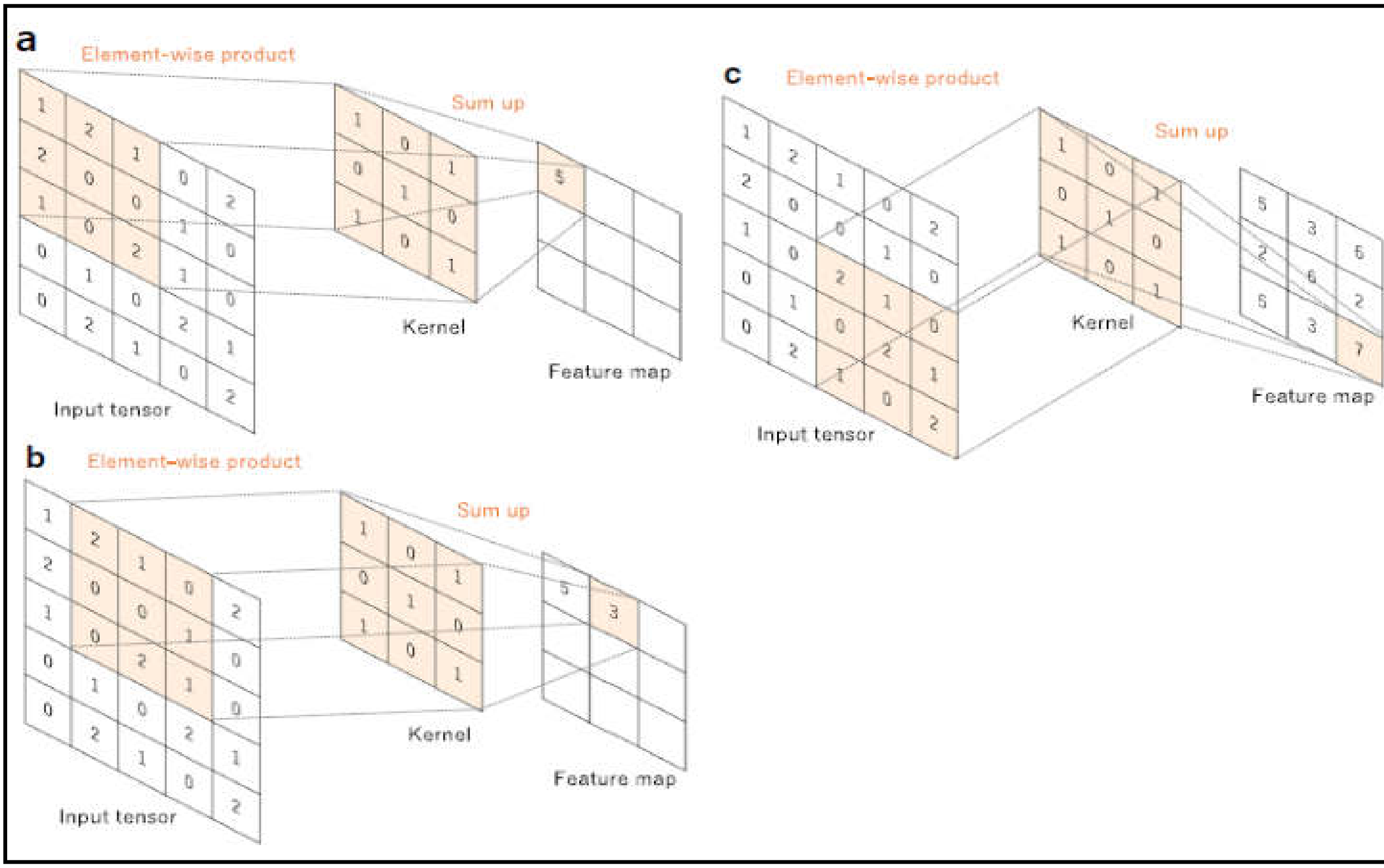


Figure 2.21: An example of convolution operation [83].

3. Non-linear Layer (Activation Function)

A non-linear transformation (or called activation function) also is applied to the input by the convolution neural network, the object of which is to classify the features within per hidden layer. The non-linear translation function in the artificial neural network is sigmoid or hyperbolic tangent. Though, if the sparsity of the data is greater, the outcome will be better for image processing. This layer uses many types of activation functions the most popular is Rectified Linear Units (ReLU).

A_ Rectified Linear Units (ReLU)

Rectified linear units are commonly used as non-linear transformation, for rectified linear unit, it applies a following equation [84].

$$y = max\ (x, 0) \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \ (2.24)$$

So that the output is in the same size as the input. Rectified linear unit raises the decision function's non-linear properties and has no negative impact on the convolution layer's sensitive fields. The training rate of the rectified linear model is much higher compared to other non-linear units. The figure (2.22) an example of the Rectified Linear Units (ReLU) [84].

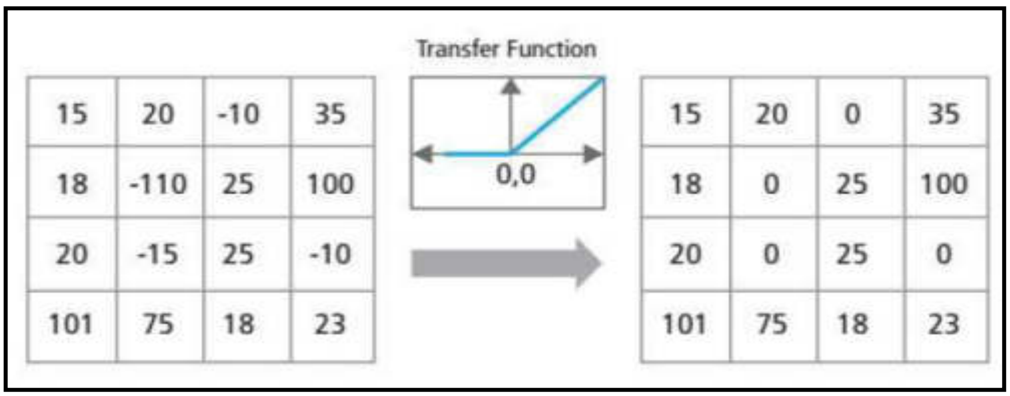


Figure 2.22: An Example of ReLU Transformation [84].

4. Pooling layers

It's usually follows the layer of convolutional and used to minimize parameters of the network and the dimensions of feature maps. Pooling layers are also invariant in translation, alike to convolutional layers because their calculations take into account neighboring pixels. The most widely used approaches are average pooling and max pooling. The next figure (2.23) describes the processes of both types of pooling.

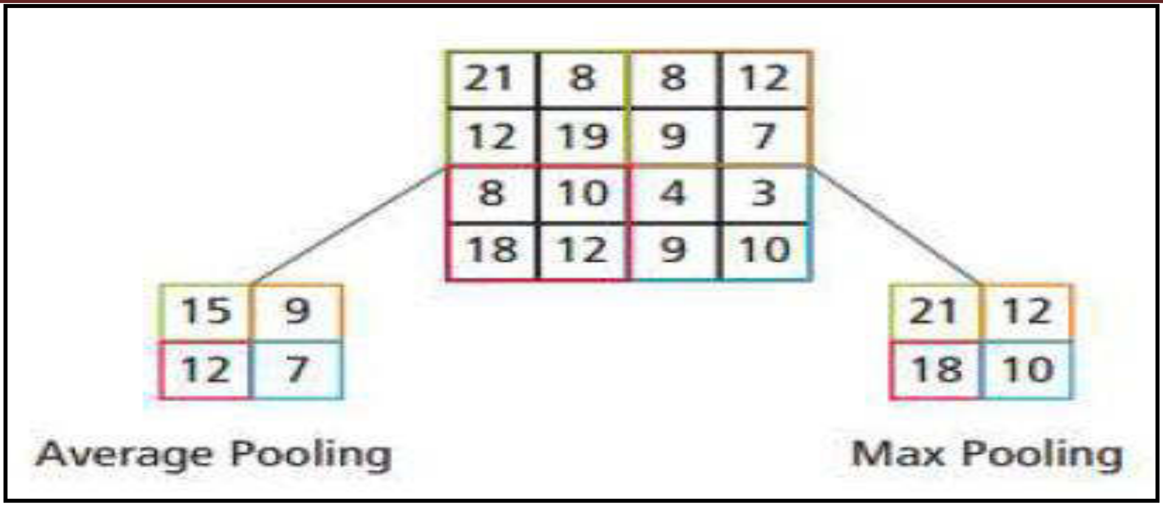


Figure 2.23: Two Classic Pooling Methods [84].

The pooling layer applies the max-pooling function with a 2 x 2 kernel and a stage of 2 along the spatial dimensions of the input, according to the reference cases. It is based on the care about the pooling layer's destructive functionality. Through this process, the feature map is reduced to 25 % of the previous size while keeping the depth capacity at its standard size. It deducts the layer to use the intervening region to move through the whole spatial dimensionality of the data. If the stage is adjusted to 3 with a kernel size set to 3, the model's output will be effectively reduced [85].

5. Fully Connected Layer

The data arrives at the final layer of the convolution neural network, which is the fully connected node, after many iterations of the prior layers. In the two neighboring layers, the neurons are connected directly to the neurons within the fully connected network. The objective from this layer is, to sum up, the weights of the features from the previous layers and show the likelihood of per class. For example, if there is a neural convolution network for gender classification and the result matrix is a probability of [0.8, 0.2], this means that there is 80 % probability of male gender and 20 % probability of

female gender [86]. Also, there are several other concepts related to the CNN structure such as MLP, normalization layer and softmax function.

6. Multi-Layer Perception (MLP)

Finishing a CNN with a multi-layer perception is popular. In practice, this is just one pixel for every filter for convolutional layers. Also used in the term Deep Neural Networks which is the product of adding layers that expanding the network size. The decision boundary is no large limited to a hyperplane when incorporating hidden layers but can lead to non-linear solutions as shown in figure (2.24) [87].

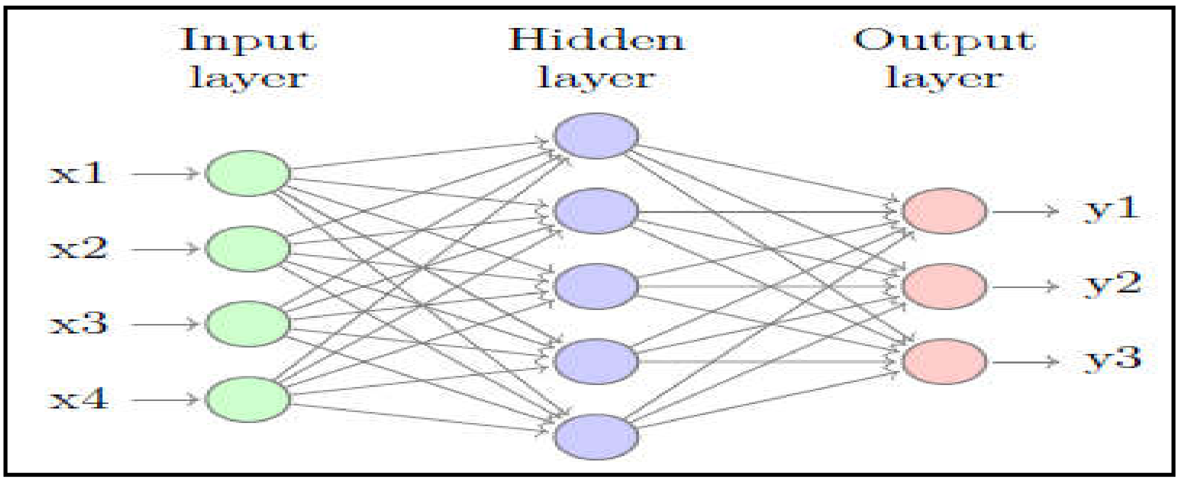


Figure 2.24: Schematic representation of an MLP [87].

7. Normalization Layer

The input layer is normalized by modifying and scaling the activations, for e.g., if it has features from 0 to 1 and also from 1 to 1000, also must normalize those features to accelerate learning. If it supports the input layer, why not do the alike for the values in the hidden layers that shift all the time and increase the training pace by 10 times or more. There are various types of normalization, the most famous of which is the batch normalization.

- **Batch normalization:** is a mechanism for enhancing the speed, execution, and stability of neural networks. In a neural network, batch normalization is achieved through a normalization step that fixes the means and variances of each layer's inputs, to compute the mini batch mean (μB) by the equation (2.25), m is the values of this activation in the mini-batch [88]:

$$\mu B = \frac{1}{m}\sum_{i=1}^{m} x_i \quad \text{…………….} (2.25)$$

While computing the mini batch variance $\sigma^2 B$ by the equation (2.26):

$$\sigma^2 B = \frac{1}{m}\sum_{i=1}^{m}(xi - \mu B)^2 \quad \text{…………………..…} (2.26)$$

At the last, normalize the layer inputs by using the prior calculated batch statistics as in the equation (2.32) [88]:

$$\overline{x_i} = \frac{xi - \mu B}{\sqrt{\sigma^2 B + \epsilon}} \text{....................................} (2.27)$$

8. Softmax

The network's performance can be difficult to interpret. It is normal to finish the CNN with a softmax function in classification issues. The following equation (2.28) expresses the Softmax [87]:

$$yi = \frac{e^{z_i}}{\sum_{j=1}^{N} e^{z_i}} \quad \text{……………………...} \quad (2.28)$$

Where the N is the number of the input Z_i is the input layer value, this normalization of the total amount of outputs to one can be defined as the likelihood of the input belonging to class i and therefore the softmax output y_i [87].

2.18.2 Training a Network

It's is a procedure of kernels obtaining in convolution weights and layers in completely connected layers that reduce various on a training dataset between specified ground truth labels and output predictions. The back propagation technique is the approach widely used to train neural networks where the concept of loss and the optimization of gradient descent algorithms play important tasks. A model efficiency under different weights and kernels is determined by absence function by forwarding propagation on learnable parameters and a training dataset, namely kernels and weights, are modified by the loss value by, among other items, an favor technique called stochastic gradient descent with momentum (SGDM) [83].

2.18.3 Stochastic Gradient Descent with Momentum (SGDM)

Stochastic gradient descent (frequently abbreviated SGD) is an iterative approach to optimize an actual function with appropriate smoothness (e.g. differentiable or sub differentiable) attributes. It can be considered as a stochastic estimate of gradient descent optimization as it substitutes the true gradient (obtained from the whole dataset) with an average of it (computed from a randomly picked data sub-set). Stochastic gradient descent in machine learning has become a significant method of optimization. The trouble of reducing a function objective in the form of average is known by both statistical analysis and machine learning [90]:

$$F(x) = \frac{1}{n} \sum_{i=1}^{n} F_i(x) \quad \text{…………… …....... (2.29)}$$

When the x parameter that decreases F (x) to be determined. Any function summand F (i) is generally joined in the i-th check in the set of data (used for training). A "batch" or standard tendency descent technique would carry the following equation [89].

$$x = x - \eta \nabla F(x) = x - \eta \sum_{i=1}^{n} \nabla F_i(x) / \text{n} \quad \ldots\ldots\ldots\ldots (2.30)$$

$\boldsymbol{\eta}$ represent a stage measure (sometimes named the rate of learning in machine learning) [89].

2.18.4 Loss Function

This function is known as an estimate function, calculates consistency through the network's output estimates by assigned ground truth labels and forward propagation. Cross entropy is the widely utilized loss function for multiclass grouping, while mean squared error is usually extended to continuous values for regression. One of the hyper parameters is a form of loss function and it requires to be decided according to the tasks provided [83].

2.18.5 Back-propagation

The back-propagation network is considered to be mostly used in the domain of neural network models. The core architecture of a multi-layer neural feed forward network based on a back-propagation training algorithm (build). The output sends out, and then the errors of the whole network are measured and re-propagated back to a certain layer. Throughout this case, the network receives the signal by the neurons in the input layer, and the production of the entire neurons is generated by the specific learning NN paradigms in the output layer of the network. The output can be repeated by one or more intermediary hidden layers. The idea of a back-propagation learning algorithm is generally utilized to fine-tune training parameters in which the errors should be decreased [90].

2.19 Evaluation Criteria

Performance evaluation of classification algorithms conducted in various methods from the confusion matrix of these methods involves information on predicted and observed classification produced by the classification method. The confusion matrix includes data from which output can be determined. The table below displays the confusion matrix of two classes [91]

Table 2.1: Confusion matrix Table

		Predicated	
		Positive	Negative
Actual	Positive	a	b
	Negative	c	d

a: describe true prediction is positive **b**: present false prediction is negative

c: present false prediction is positive **d**: present true prediction is negative

The following equation can be used to calculate accuracy [91]:

$$Accurcy = \frac{a+d}{a+b+c+d} \qquad \text{(2.31)}$$

Or use the equation below

$$Accurcy = \frac{\text{number of correctly classified images}}{\text{Total number of images}} \times 100\% \qquad \text{(2.32)}$$

Or represent the following in pixel form:

$$\text{Accurcy} = \frac{\text{Total number of correctly classifiy pixel}}{\text{Total classifiy pixel}} \times 100\% \qquad \text{(2.33)}$$

Chapter Three

Plant Leaf Diseases Detection

Chapter Three
Plant Leaf Diseases Detection

3.1 Introduction

Plants are the essential and necessary element in our life and we cannot live without it, because it is the world's first indispensable product. For these reasons, all countries of the world pay close attention to plants to protect them from diseases and strive to use the latest methods and techniques to develop and increase their productivity. This work tries to contribute to providing a system that helps in detecting and classifying diseases that affect plants and controlling them to avoid their negative and harmful effects on plants. This chapter included the proposed plant leaf diseases detection approach by using image processing and deep learning technique. It will start by introducing and discussing the general block diagram of the approach. Furthermore, this chapter will present the architecture of the approach in details with the proposed algorithms in the different stages of the approach.

3.2 The Proposed Approach Design

The proposed approach aims to identifying diseases in plants leaf; the work includes two proposed approaches. They are separated based; each one is different from the other.

According to figure (3.1) below, the approach will be divided into two parts depending on the detection method. The main steps of this approach are as below:

1. In the first proposed approach, the Multi-class support vector machine (MSVM) algorithm is applied and their stages will be done up to the MSVM detection algorithm for plant leaf diseases.
2. In the second proposed approach based on deep learning where convolutional neural network (CNN) algorithm is applied. It also has various stages depends on building this work.
3. This approach will detect diseases and take into account the most common diseases such as bacteria, fungi, viruses, etc. which has been explained in chapter 2 in sections (2.2).
4. Through this approach and at the end, plant leaf diseases will be detected by applying the stages of the proposed methodology of the approach. Figure (3.1) shows the general block diagram of the approach for detecting plant leaf diseases and its stages will be explained later.

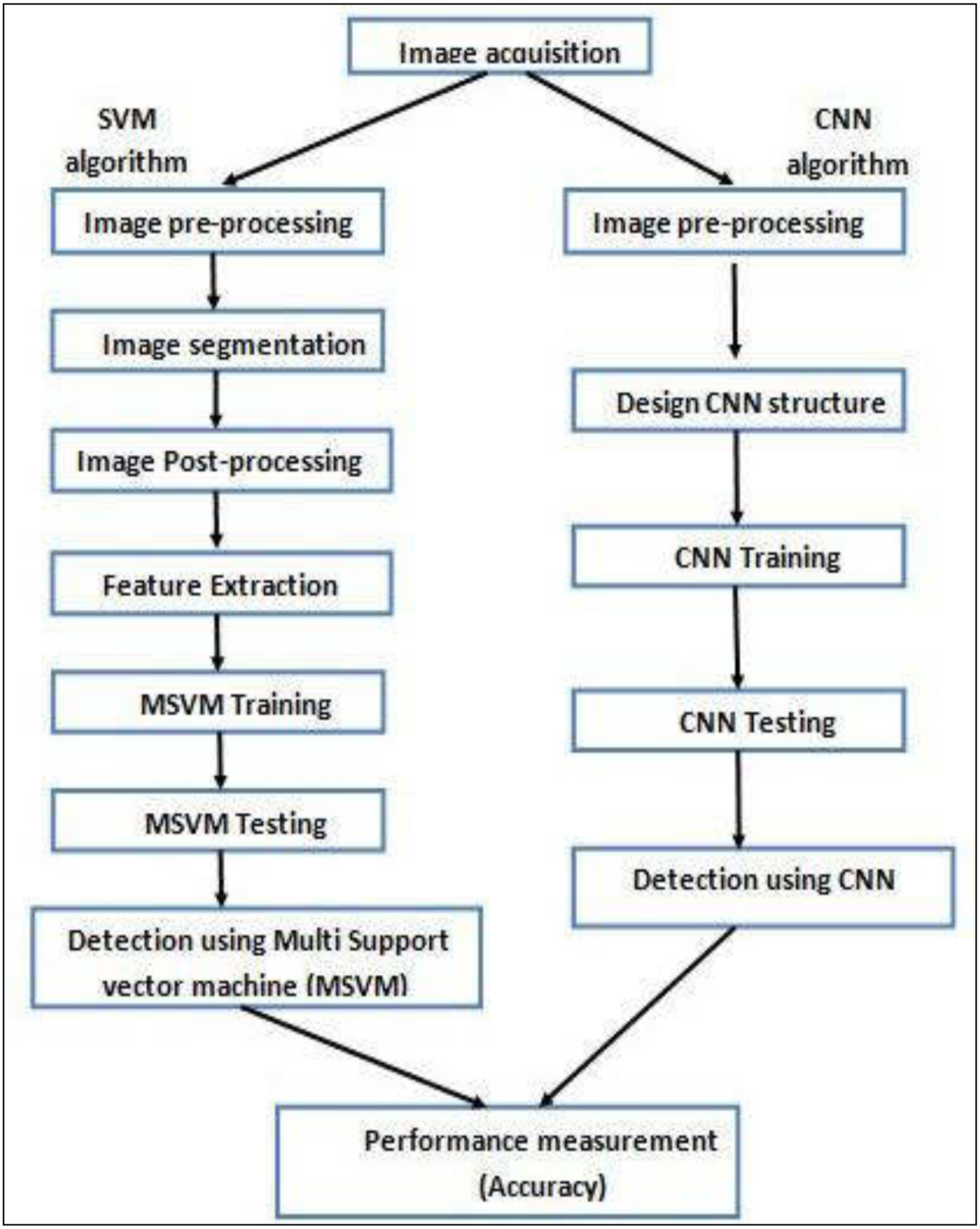


Figure (3.1): General block diagram for Plant leaf diseases detection approach.

Figure (3.1) illustrates the major stages of a general diagram which will be applied for the two used proposed approaches. In this general diagram, the stage (image acquisition) is a common stage between the two proposed approaches and will be discussed in detail later. In the first proposed approach, there are several stages that will have been carried out after image acquisition, these are include, image pre-processing, image segmentation, image post-processing, feature extraction and the detection stage.

In the second proposed approach, after image acquisition there is image pre-processing stage, the CNN structure has been built, and after that training and testing operations will use the dataset to complete the detection process. Before explaining the two proposed approaches, the joint stage between these two proposed approaches, which is the (image acquisition) stage, will be explained.

3.3 Image Acquisition Stage

The difficulty of this work is to creating dataset which required by large farms with various diseases, and the proposed approach seeks to detect and classify many diseases for many plants, not just one disease for single plant. This is the main reasons why it is too difficult to create a dataset consisting of thousands images of different plants. For these reasons, we have recourse to the (Plant Village) treasury which is a highly popular and certified global site that contains thousands images of plant leaves and their diseases. In the proposed approach, three very important plant types were chosen; these types are tomatoes, peppers, and potatoes. These plants have been chosen because of their great importance and their spread in the world in general and in our country Iraq in particular. The collected data consists of about 20636 images related to 15 different categories as shown in figure (3.2). The dataset includes

images of all classes of diseases that can be affecting the three types of these plants; the diseases are bacteria, viruses, fungi, moulds and mites. These diseases and the specific types for each disease will be classified according to the symptoms that have been mentioned in chapter 2 in section (2.2). Every sample in the dataset was RGB colour space with size 256 * 256 pixels and the images in JPG file format.

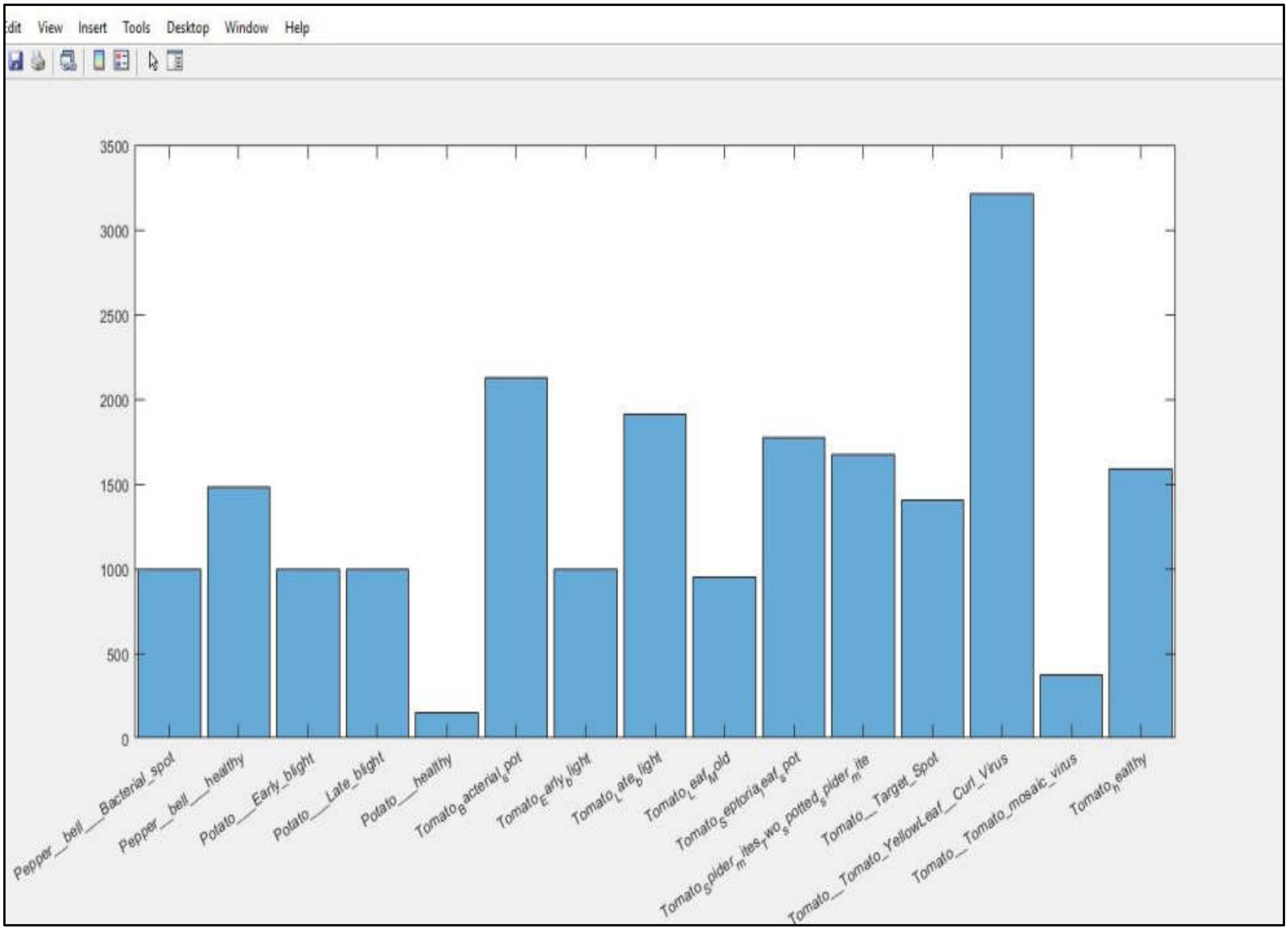


Figure 3.2: Chart shows crops and their disease that be used as a dataset

The approach is divided into two main sub approaches, the first proposed approach was using the Support Vector Machine (SVM) algorithm, while the second proposed approach using Convolutional Neural Network (CNN) algorithm for detection the plant leaf diseases, and they will be explained in details in the following sections.

3.4 The First Proposed Approach: (Multi-Class Support Vector Machine Algorithm (MSVM))

MSVM is an algorithm considered to be amongst the popular algorithms of machine learning technique. Its definition, details and equations is presented in chapter 2, section (2.16). This section uses MSVM for classification after completion of all stages of the approach, these stages are shown in the following block diagram in figure (3.3), and a brief explanation for each stage is presented in the next section.

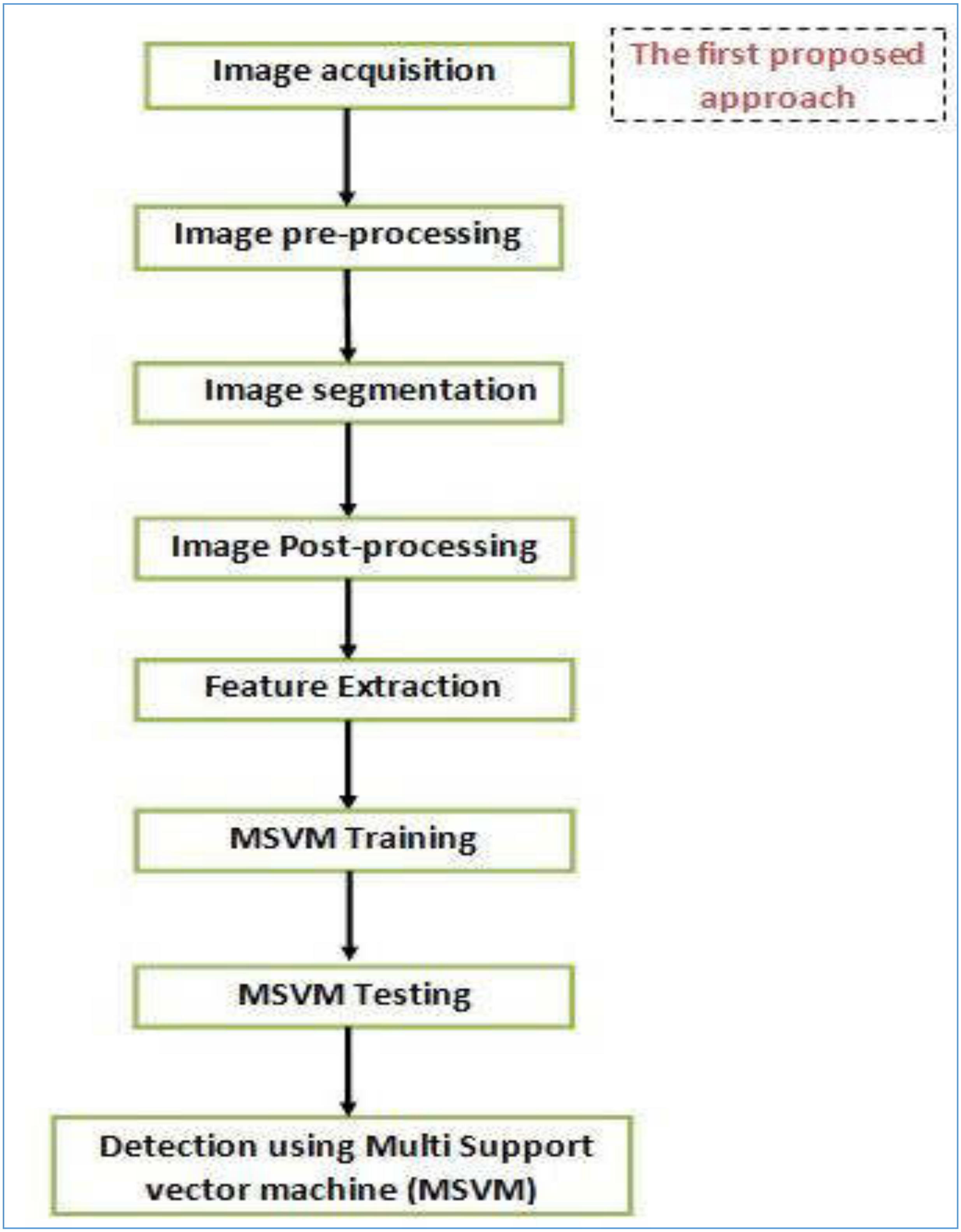


Figure 3.3: The block diagram for the first proposed approach

3.4.1 Image Pre-processing

The pre-processing stages are carried to make the images more suitable for further processing; the main image pre-processing operation in the proposed approach is the conversion of images from RGB to L*A*B*colour space.

3.4.1.1 The Image Conversion from RGB to L*A*B*

At the pre-processing stage, images are converted from RGB to L*A*B*colour space. L represents "Lightness" ((intensity)), A – colour component of Green to Magenta, B – colour component of Blue to Yellow. The mechanism for converting the colour image from RGB to L*A*B*is explained in chapter 2, section (2.6.1), and the equations (2.1), (2.2) and (2.3) are used for conversion. The transformed of the colour space into L*A*B* to distinguish the colour and the affected area based on the colour illumination. In this case, the affected part will be segmented precisely with the use of the k-means clustering in the later stage. It should be noted that the conversion is used only for the image segmentation. Algorithm (3.1) summarizes the conversion of colour space of the plant leaf image.

Algorithm (3.1): The image conversion from RGB colour space to L*A*B* colour space	
Input:	RGB Plant leaf Image
CIEXYZ's //	Colour space coordinates are X, Y, Z, Xn, Yn, and Zn
Output :	Plant leaf Image L*A*B* colour space

Algorithm (3.1) Continued

Step (1): Begin

Step (2): Read The RGB plant leaf image

Step (3): Convert from RGB to XYZ using equation (2.1)

Step (4): Calculate the value of f (x) with the determination of range as in equation (2.2)

Step(5) : Convert from XYZ to CIELab using equation (2.3)

Step (6) : Return image with(L*A*B*colour space)

Step (7): End algorithm

3.5 Image Segmentation

In order to distinguish the affected area of the plant's leaf, the image segmentation process is carried out. This stage is very important in order to determine the part that will be working on later for the process of extracting the selected features. The image segmentation was previously explained with their types in chapter 2, section (2.7). The proposed approach will give more trustworthy results using cluster subtraction by using K-mean clustering algorithm.

3.5.1 Images Segmentation Using K-Mean Clustering Algorithm

In this stage, the segmentation of the image using the K-mean clustering algorithm. This algorithm was explained in detail in chapter 2, section (2.7.2). In this method, the image of plant leaf is divided into k clusters where each remark has a proper place to the region with the nearest mean, pointing to the cluster as a concept. In this work, Euclidean Distance measure was utilized; the clustering of k-means differentiates the sample image onto three clusters with relation to the colour pixel of each component. So, it needed to convert to the L*A*B*colour space because the process of segmentation depends on the

illumination of the colour. The entire nearly related colour will be in the cluster, and will choose the cluster that segments the disease based on its colour as shown in figure (3.4). Algorithm (3.2) clarifies K-mean clustering algorithm for segmentation of plant leaf images.

Algorithm (3.2): k-mean clustering for image segmentation	
Input:	L*AB* Plant leaf Image size 256 *256 Number of cluster \\\\ k
ED //	Euclidian distance
Ci //	Cluster centroid
Output:	Segmented image
Step(1):	Begin
Step(2):	Select number of clusters k
Step(3):	Select the initial cluster centroid ci, i = 1... k randomly.
Step(4):	Compute the Euclidian distance using equation (2.4)
Step(5):	Go to the following points and measure the new centroid
Step(6):	Update new centroid by mean of every clusters point
Step(7):	Compute ED for the next check
Step(8):	Continue until no change in centroid
Step(9):	Return segmented image
Step(10):	End algorithm

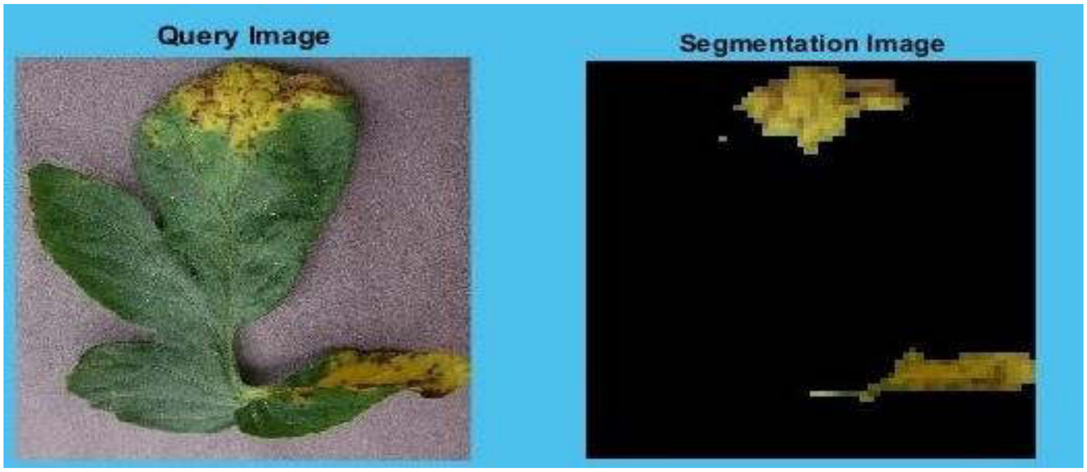


Figure 3.4: Plant leaf image segmentation after using the k-means clustering algorithm

3.6 Image Post-processing Stage

This process is always modifying or smoothing images to make them better and prepare them for the next stage or it is also used to perform another specific process that need in the next stage. In this proposed approach, this process is needed to convert the images to the gray-scale level.

3.6.1 Conversion of Segmented Image to Grayscale Image

After the image segmentation stage, the next stage will be post-processing, to configure the data for the next stage. After image segmentation, the image must be converted to gray-scale in order to extract the features from images in an easier way to be used later. The conversion process was explained in chapter 2, section (2.8.1) and is carried out by using the equation (2.6). Algorithm (3.3) shows the conversion process steps.

Algorithm (3.3): Conversion RGB image to gray-scale
Input: Colour Image (segmented image)
Output: GI \\ Gray-Scale Image
Steps :

Algorithm (3.3) Continued

Step(1) : Begin
Step(2) : Read Plant image
Step(3) : Rows = image. Height
Step(4): Columns = image .Width
Step(5): Loop For i=1 to rows
Step(6) : Loop For j=1 to columns
Step(7): Find pixel =Image (i , j)
Step(8): Compute the GI by using the equation 2.6
Step(9): End for j
Step(10): End for i
Step(11): Return Gray Scale image (GI)
Step(12): End Algorithm

3.7 Feature Extraction Stage

The different features are extracted after segmentation to characterize the infected area. Important features of the images are extracted for later use in the classification process, based on these features. There are several types of features that were explained previously in the second chapter, section (2.9). Extracted additional features are very important by utilizing the Gray Level Co-occurrence Matrix (GLCM) method, from which extracted four important features that were used in the classification process, upon features arrangement, at the beginning will be explained first-order features.

3.7.1 Statistical Features Extraction (First Order Features)

The statistical features are extracted from the images after performing the operations that were demonstrated previously in section (2.10). These features depended on the colour moments of the image is used in the calculation, as illustrated in chapter two. The features are taken from the colour image; which means the segmented images in the previous stage don't need to convert to

gray-scale for certain features. These features are “mean”, “standard deviation”, “entropy”, “root mean square”(RMS),“variance”, “kurtosis” and “skewness”, are shown in table (3.1). The table (3.1) explains each feature with its description and the number of the equation that was used for calculating it.

Table 3.1: first-order features

No.	Features	Description of features	Equation numbers in chapter two
1.	Mean	It’s a medium value, (Average) or mean of matrix elements	(2.7)
2.	Standard Deviation	It is also named the square root of the contrast, showing the contrast of the image	(2.8)
3.	Entropy	It defines corresponding intensity level states which can be adapted by individual pixels	(2.9)
4.	RMS	It is described as the square root of the mean square	(2.10)
5.	Variance	It is the assumption that a random variable will squarely deviate from its mean	(2.11)
6.	Kurtosis	It represents the state of probability dissemination	(2.12)
7.	Skewness	It is a metric of the imbalance of the distribution of the likelihood of a real-valued arbitrary variable over its mean	(2.13)

3.7.2 Gray Level Co-occurrence Matrix(GLCM)

It is a matrix that described as the distribution of co-occurring pixel values (gray-scale values) across the image. GLCM can measure textural features to understand the nature of the information of the image. This method was explained in detail with its features in chapter 2 in section (2.12). GLCM of an image is calculated by using a displacement (d) value of (2) is chosen in order to extract the detailed textural features, with an angle, θ (0) as shown in the algorithm (3.4). Every feature is determined from GLCM for each block four main features are determined for each GLCM according to the equations that described in chapter 2 for every feature. The important texture features

applied in these stage are explained later. The vector of all features is saved in a dedicated database file, called *“ feat-disease”*, to be used in the next stage are shown in algorithm(3.4) which summrized the GLCM steps method.

Algorithm(3.4)	**Features Extraction based on GLCM method**
Input:	Gray Image (plant leaf image) \\\\ ImG (feat_disease) \\\file to save features. GT \\\ gray tone, d=2 \\\ distance between two pixels, w \\\ width, h \\\ hight,
Output:	Features vector (feat-disease)
Steps:	Begin
Step (1):	Set the angle direction to zero , (θ=0)
Step(2):	Compute the width value of gray tone, (w=GT.width)
Step(3):	Compute the height value of gray tone, (h=GT.height)
Step(4):	Loop start from zero to gray tone width value, (For i=0 to w)
Step(5):	Loop start from zero to gray tone height value (For j=0 to h)
Step(6):	Set the values of GM to zero's and c equel to zero,(GM(1 to h, 1 to w) = 0, c=0)
Step(7):	Loop start from zero to gray tone height value,(For i=1 to h)
Step(8):	Loop start from zero to gray tone width value -1,(For j=1 to w-1)
Step(9):	Compute the first gray image and put it in Temprory1, (Temp1=ImG(i,j))
Step(10):	Compute the second gray image and put it in Temprory2, (Temp2=ImG (i,j+1))
Step(11):	GM(Temp1,Temp2)= GM(Temp1,Temp2) +1
Step(12):	GM(Temp2,Temp1)= GM(Temp2,Temp1) +1
Step(13):	Increase the c value by 1 , (c = c +1)

Algorithm (3.4) Continued

Step(14): End of first loop i

Step (15): End of second loop j

Step(16): Compute the probability matrix using Gm divided by c value, (probability_matrix[i][j] = GM/c)

Step(17): Loop start from zero to gray tone hight value, (For i=0 to h)

Step(18): Loop start from zero to gray tone width value, (For j=0 to w)

Step(19): End of first loop i

Step (20): End of second loop j

Step (21): Extract co-occurrence features from each matrix

Step (22) : Save features vector on “feat-disease” file

Step(23): Return Features Vector (“feat-disease”)

Step(24): End algorithm

Generating the GLCMs by using the gray co-occurrence properties function, it can obtain some statistics from them, such statistics give detail on an image's texture. Figure (3.5) shows the process of extracting features using the GLCM algorithm with different angles that used in this algorithm until the four features are extracted and saved in the features vector.

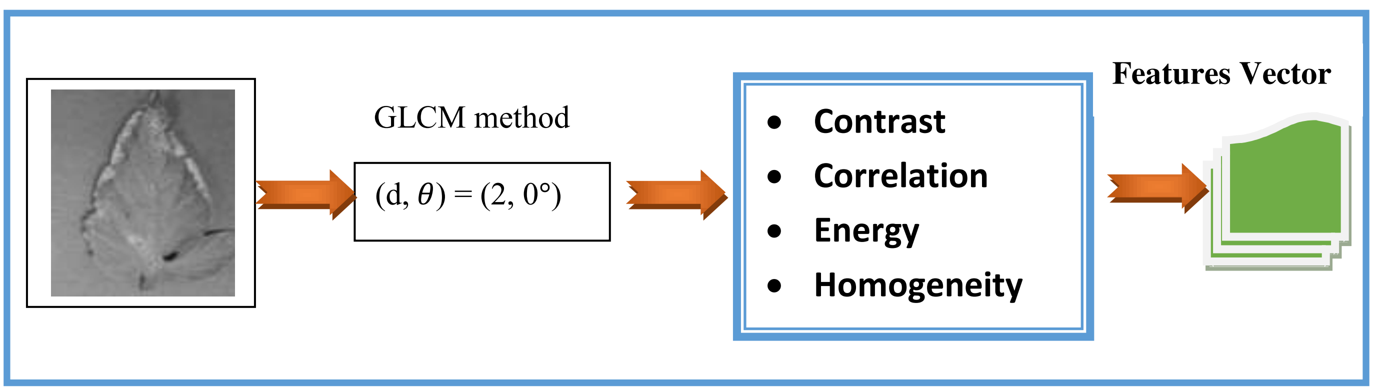


Figure 3.5: Features extraction from GLCM

The table (3.2) below explains the (second-order features) that were extracted from the GLCM algorithm with the description and range for each

of the four features, as well as the equations through which the features were calculated that are mentioned in chapter two.

Table 3.2: second-order features

No.	Features	Features describe and its range	Equation numbers in chapter 2
1.	Contrast	Its intensity contrast over the entire image between a pixel and its neighbor. Range = [0 (size(GLCM,1)-1)^2]	(2.14)
2.	Correlation	It estimates the linear reliance of neighboring pixel gray levels. Range = [-1 ... 1]	(2.15)
3.	Energy	It is the summing up of the GLCM squared components. Range = [0 ... 1].	(2.16)
4.	Homogeneity	It's the closeness of the GLCM component distribution to the diagonal GLCM. Range= [0 ... 1].	(2.17)

3.8 Detection Stage Using MSVM Algorithm

This stage is the last stage in a first proposed approach. The Detection process is the summary of the work through which the decision is made. The MSVM algorithm is chosen, as it one of well-known traditional algorithms of supervised learning of machine learning algorithms. This algorithm was explained in detail with its equations in chapter two, section (2.16). The extracted features from the previous stages will be adopted in this stage. In the previous stage, 11 important features were extracted from the plant leaf image. They were saved in a function that is called in the classification process, which is a function (feat-disease) as mentioned in the previous algorithm (3.4) in this function; the seven features that were extracted before the GLCM method are also stored. The MSVM algorithm as explained previously depends on the establishment of the hyperplane in order for the data to be separated, because this work needs more than two classes in the detection process, for this reason must use the multi-class MSVM method it

was explained in the previous chapter in section (2.16.1.B) it is also called binary SVM. The following algorithm (3.5) shows the steps of MSVM Classifier.

Algorithm (3.5): Multi-class Support vector machine classifier	
Input:	Training _set (Features function (feat-disease))
Output:	Class name
Steps:	
Step 1:	Begin
Step 2:	Establish the training label for all training sets and identify 15 classes
Step 3:	Specify the kernel function ((polynomial function), as calculated in the second chapter in the equation (2.21)) that applied to map the training set to kernel space by hyperplane computation, Its calculation was also defined in the previous chapter in equation (2.19)
Step 4:	Segregated data to 1 against all method
Step 5:	Passes testing set on MSVM depended on the hyperplane to identify the class name
Step 6:	Return class name
Step 7:	End algorithm

In the above algorithm (3.5), the steps of the algorithm was shown to obtain the training label with the features that have been extracted and saved in the features function on the basis of classification. Also, the kernel function was used, more than one kernel function was tried and the best results obtained using the polynomial function that was used to map the training datasets that are explained and calculated in chapter 2 in section (2.16.1.A) in equation (2.21). The data used in this work dividing into two parts for training and testing.

A. MSVM Training

At this stage, the proposed approach has 15 classes of different types of plant leaf diseases to be detected. In general, the MSVM algorithm classifies only two classes, but here it has more than two classes. For this reason, the multiclass MSVM method type One_Versus_Rest (1VR) approach is used, which is explained in the previous chapter two, section (2.16.1.B.1). The main idea is to express the problem oppositely: rather than writing "class A vs. class B vs. class c ...." the issue can be written "class A against the rest, class B against the rest," ...., n, ultimately n differential learning issues are the inverse to "class n against the rest. In the training phase, the images obtained on the MSVM algorithm are entered, in order to be classified in any class for any disease. Each MSVM binary classifier is trained to utilize a vector of training data, every row associates to features extracted as an investigation from a class. Following the training stage, the multi-class MSVM model is able to decide the right class for the input features vector.

B. MSVM Testing

The trained MSVM model was used as the classifier; the classifier produced independent test dataset was categorized to the classifier's test accuracy. In order to estimate and evaluate the quality of the model training that has been proposed as a specific group is used randomly for this stage which is the testing stage. In this stage, each row of features vector that was previously extracted was categorized and unlabelled here, which is in the training stage are labeled rows. The classified approach was created using the information provided in the training stage and in the features vector. Here, the sample is where each feature is related to a specific column and a matrix of samples is formed. The sample size must be divided by the same size of the

training data because the number of columns represents the number of features.

C. Detection for Plant Leaf Diseases in the First Proposed Approach (Performance Measurement)

In detection, there are several forms online or offline to conduct the classification process. In this work, the Offline form was performed to conduct the detection process, based on the previous steps and on the training dataset, and a user interface was created for the detection process as it will be presented in the next chapter.

3.9 The Second Proposed Approach using CNN Algorithm

Convolution Neural Network (CNN) is one of the very famous and powerful algorithms in the field of deep learning. It is used in several fields, the most important of which is the classification between several classes. A CNN is a class of deep neural networks, most widely used to visual imaging analysis. First, the diagram for the first proposed approach to classify and detect the plant's leaf diseases will be displayed in figure (3.6). Then the algorithm (3.6) for CNN training will be showed. In this algorithm, the work of the second proposed approach and the processes that occurred to detect and classify plant leaf diseases were presented starting from the first stage, which is pre-processing, as well as splitting dataset to training and testing, that have been chosen for the ratio It is 70 % for training and 30 % for testing, after testing all ratios, and this was the best ratio as it will be shown in the next chapter, and the network design stage that consists of several layers, also obtaining the best weights for the network after performing several operations in order to obtain the trained network to detect and classify plant leaf diseases.

Algorithm (3.6): CNN training algorithm to classify plant leaf diseases	
Input:	Plant leaf Image with size 256 * 256 *3 (JPG format)
Output:	CNN with optimum weight
Steps:	
Step(1):	Begin
Step(2):	Pre-processing stage in it the image is resizing from 256 * 256*3 to 128 * 128 *3
Step (3):	Splitting the data into two parts, 70% for training and 30% for the testing.
Step(4):	CNN design which consisting of several layers: a) Input layer : RGB image with 128 * 128*3 size b) Convolution layer : Multiple filters were used with size 3*3 c) Nonlinear layer (Activation layer) : Using Rectified linear units (ReLU) d) Pooling layer :Using Max-pooling layer e) Normalize layer : using batch normalization f) Fully connected layer g) Softmax layer
Step(5):	For each pattern in the training dataset: a) Input current pattern (input Image with Label) b) Calculate the real output of the CNN through Softmax layer c) Calculate the error rate by comparing the real output with the desired output. d) Compare the performance goal with the error rate: 1. If the performance goal was not meet, change the connection weights by using the back-propagation learning algorithm. 2. Else, go to next image pattern. e) Stop condition: 1. If the performance goal was meeting with validation data or the maximum iteration was achieved, go to step (6). 2. Else, repeat step (5).
Step(6):	Return the CNN with the optimum weight.
Step(7):	End algorithm.

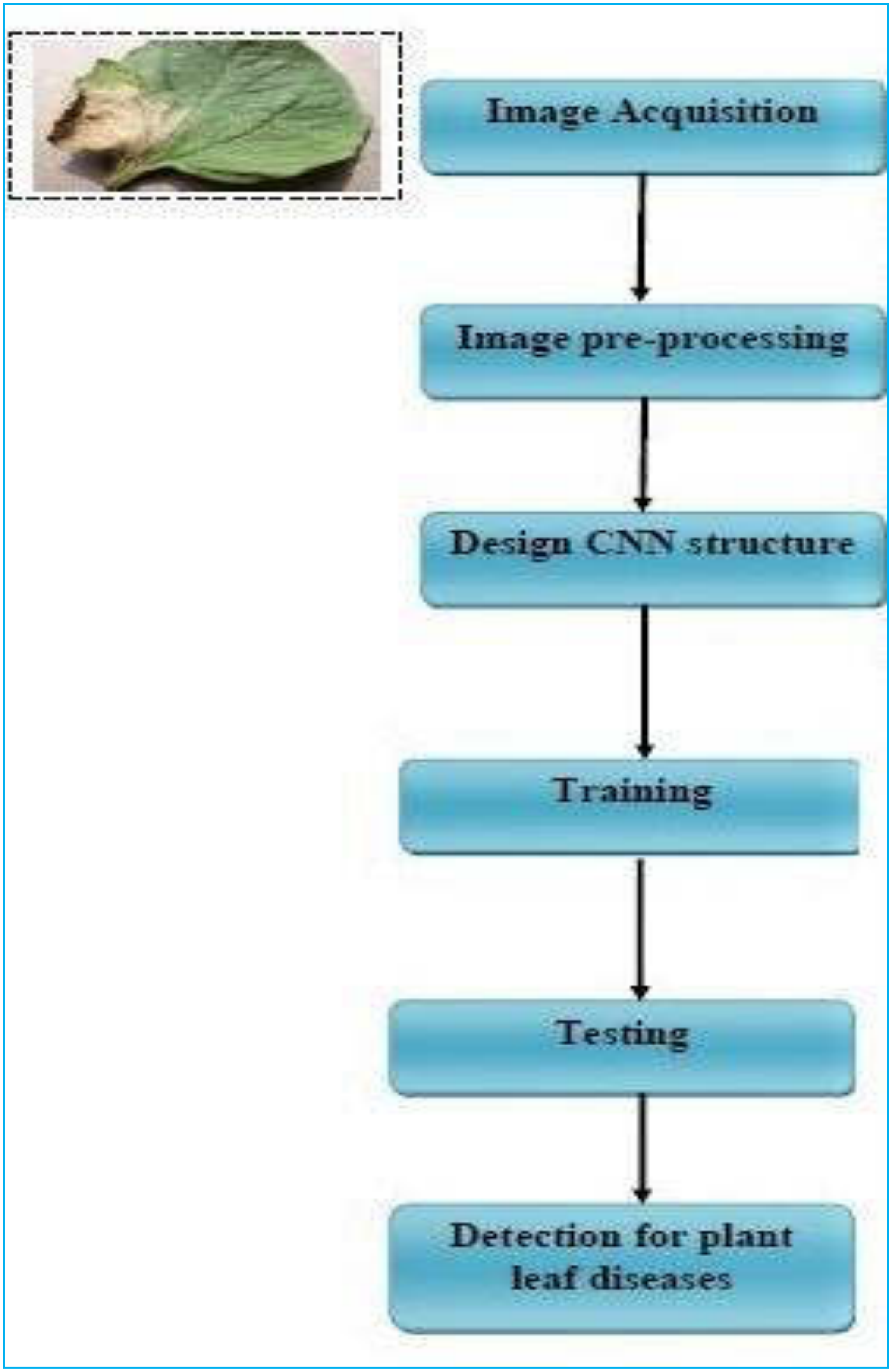


Figure 3.6: The diagram for the second proposed approach stages using (CNN) algorithm for plant leaf diseases detection.

CNN algorithm is explained in details with all its parts in the second chapter in the section (2.19). To facilitate and clarify the explanation more, the second proposed approach has been presented as a chart, as in figure (3.6) present the structure of the second proposed approach stages using CNN algorithm for plant leaf diseases detection.

The first stage, which is the images acquisition stage, it has been explained previously in the section (3.3), so the second proposed approach will start with the next stage, which is the pre-processing stage.

3.10 Image Pre-processing

At this stage, the image size is resized before it enters the CNN. The purpose of reducing the size of the images to increase the speed of the training manner and obtain the model testing calculation realistically. The images are resized from 256 * 256 to 128 * 128 image size. The process of optimizing both the input and variables help to speed up the training process. While preserving the integrity of the data without detriment to the original image data. The figure (3.7) is an example of changing the size of images from the proposed approach. The images were converted by using the resizing function of each width and height of the image from 256 * 256 to 128 * 128.

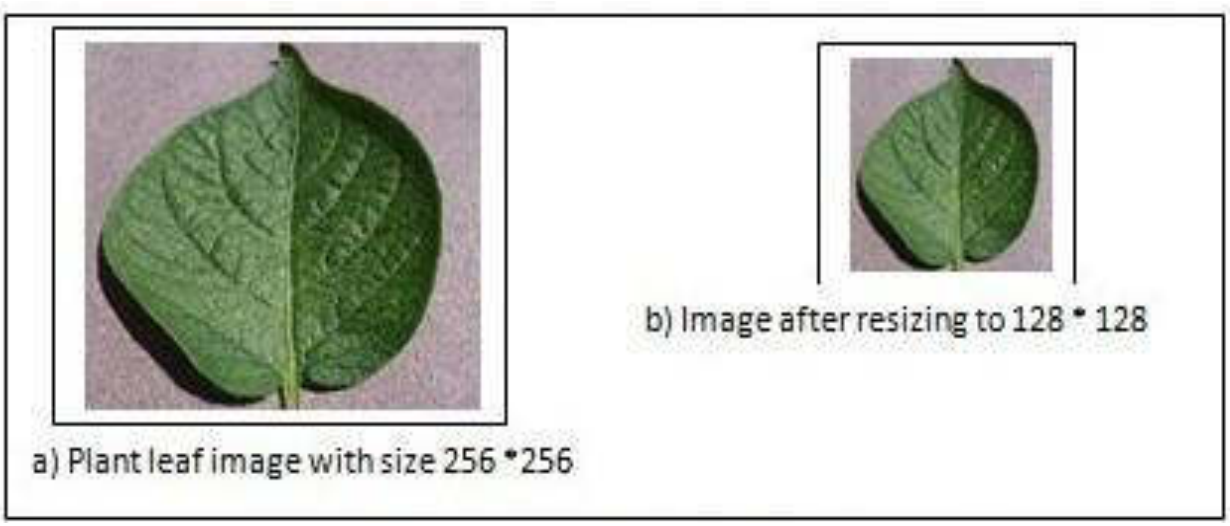


Figure 3.7: Resizing plant leaf image

In Figure (3.7), (a) indicates the original image with a size of 256 * 256, while (b) indicates the image of the leaf of the plant after changing its size to 128 * 128.

3.11 Design Convolution Neural network (CNN) Structure

The CNN is one of the well-known networks in the field of deep learning and contains many characteristics that distinguish it from the traditional neural network, such as the use of fewer parameters and the number of neurons is less in the CNN, so the training time is less, etc. In designing this network, there are several famous types of designs of the convolutional neural network, such as Alex Net, Google Net, LeNet, etc. So the structure is designing in the figure (3.8) to suit the proposed approach and after changing many of the parameters and testing it, by choosing this design for the network for obtaining the best results.

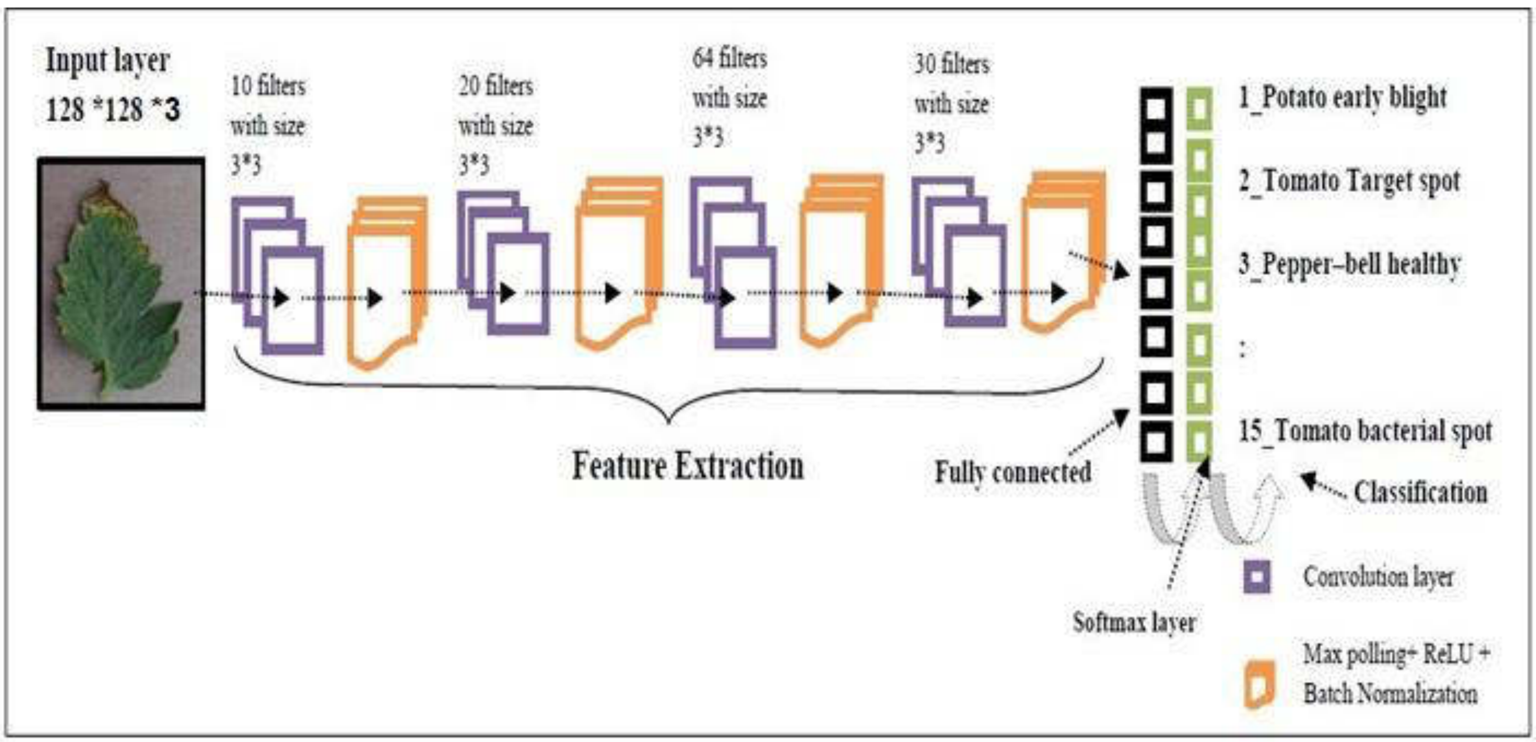


Figure 3.8: Structure of the CNN algorithm

As shown in the figure (3.8) above, the structure of the CNN algorithm consists of several layers for each layer a specific work and a different structure, the structure is designed as follows:

1. Input Layer

The input layer contains the input images and their pixel values. These images are the images of plant leaf diseases that entered in matrix form, or in other words layers, whose size has been reduced in the previous stage to the size of 128 * 128 * 3. The input layer consists of three layers because the dataset is RGB colour images are red, green, and blue, so each colour has a specific layer as in the example shown below in figure (3.9) that explains the input image is an image of the affected tomato leaf and the process of converting it into matrixes to be dealt with it in the other stages.

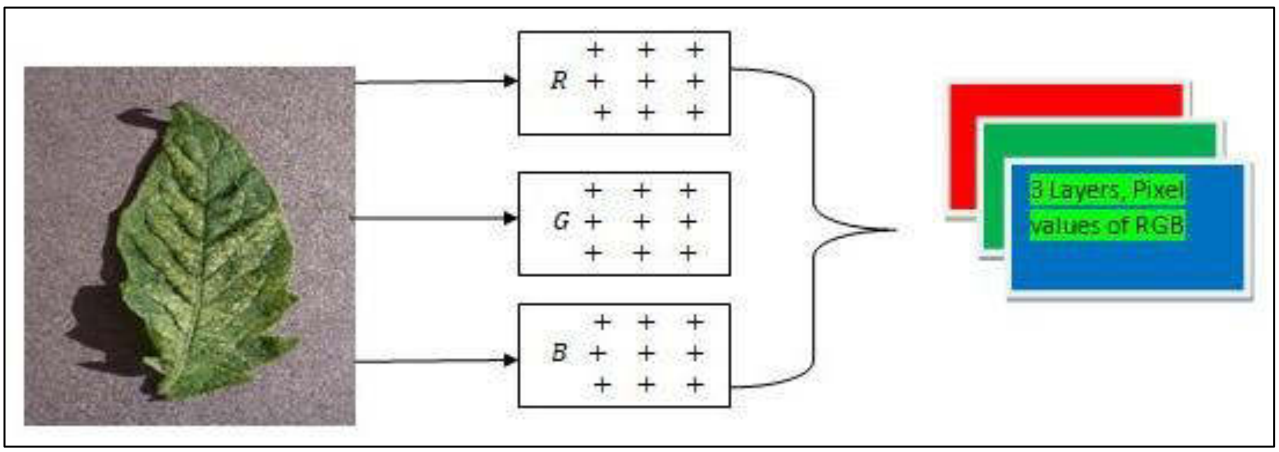


Figure 3.9: The input layer (input image).

2. Convolution Layer

A convolution Neural Network uses the different kernels, in the convolution layers to convert the entire object as well as the optimal feature maps creating different feature maps. Not only does CNN utilize kernel convolution, but it's also a staple of many smart vision algorithms in usual. It's a mechanism that takes a short array of numbers that named a kernel or filter;

the filters are the shared weights and bias. These filters work on every section of the image it searching for the similar feature throughout the image.

The values are calculated according to the equation that was calculated in the second chapter in the part (2.18.1.2), which is the equation (2.23). Also as in the example showed also figure (2.21) in the previous chapter also, that clarifies the work of the convolution layer. In the second proposed approach, four convolution layers are used. In the first convolution layer, 10 filters were used with dimension 3*3 with “same padding”. The padding applied to the input along the edges, the 'same' means padding is set so that the output size is the same as the input size. In the second convolution layer, 20 filters were used with dimension 3*3 with "same" padding. In the third convolution layer, 64 filters were used with 3*3 dimension with padding 'same' and in the fourth convolution layer, and 30 filters were used with 3*3 sizes with the same padding also, as shown in figure (3.8). Choosing the number of filters in each of the convolution layers, based on several experiments that prove the best result that obtained of these numbers that were used in each level.

3. Non-linear Layer (Activation Function)

A non-linear transformation is implemented to the input by the CNN, it is also called (Activation function), and the node activation function describes the node output provided by the input or input set. There are different types of activation functions, in the second proposed approach using the best type after the experiment, which is the most famous type, which is Rectified Linear Unit (ReLU), which was explained previously in chapter two, section (2.18.1.3.A). In this function, negative values in the matrix resulting from the previous step are converted to 0 and positive values remain the same. It also showed with its calculation in the previous chapter in the equation (2.24), and its work also

has been clarified more in the example in figure (2.30). They are used in the proposed approach after each level of the convolution layers were shown in figure (3.8)

4. Pooling Layer

Another building block to the CNN is a pooling layer; the purpose of this layer is to gradually decrease the spatial scale of the representation in order to reduce the number of parameters and the network calculation. A pooling layer often follows a convolutional layer and can be utilized to depreciate the dimensions of feature maps. Pooling layers are also invariant in interpretation, like to convolutional layers because their calculations take into account neighboring pixels were explained in detail in chapter two in the section (2.18.1.4). The most widely used approaches are average pooling and max pooling layer, in the proposed approach the Max-pooling layer is applied. As explained in the previous chapter in the example was shown in figure (2.23), after performing the experiments, Max pooling was chosen because it gives better results. It takes the highest value in the matrix specified for pooling and passes the process to all the values of the matrix and therefore has another matrix with fewer dimensions. In the proposed approach 2*2 Max-pooling is used with two strides, the "stride" is mean “the step scale to vertically and horizontally traverse the input” is shown in figure (3.10) to explain Max-pooling layer process.

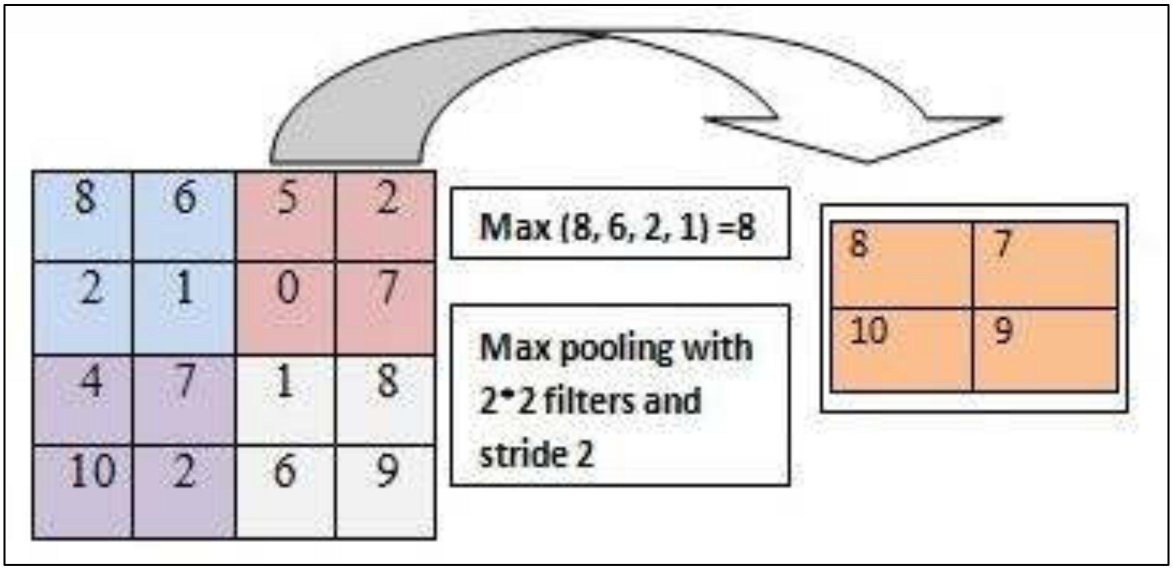


Figure 3.10: Max pooling layer

5. Normalization Layer

It is a very important layer that was explained in chapter two, section (2.18.1.7). Batch normalization layer is using in the second proposed approach; it forms norms any channel by means of a mini-batch. This can help diminish sensitivity to variations in the results. It eliminates the fitting problem, as it has a slight effect on regularization. Batch normalization optimizes the yield of the prior activation layer, to calculate the batch normalization, at first calculate the mean and the variance for the layer that was inserted, and then do scaling and shifting to calculate the batch normalization, as shown in the algorithm (3.7).

Algorithm (3.7): Batch Normalization	
Input:	The matrix of the convolution layer, suppose C = (x_1 .... m)
Output:	Batch normalization (Y_i =BN), β(xi)
Steps:	
Step(1):	Begin
Step(2):	Calculate the mini-batch mean (μB) of the layers input by using equation (2.25).
Step(3):	Compute the mini-batch variance ($\sigma^2 B$) of the layers input by using equation (2.26).
Step(4):	Normalize the layer inputs utilizing the prior calculated batch statistics ($\overline{x_\iota}$) by using equation (2.27).
Step(5):	Scale and shift to get output from the layer $Y_i \leftarrow \overline{x_\iota} + \beta \quad \equiv$ BN, (xi)
Step(6):	Return Batch Normalization
Step(7):	End algorithm

6. Fully Connected Layer

Often the last layers of a CNN are fully connected layers, in a traditional neural network; these would behave identically to the layers. The major difference is that the inputs would be in the form that CNN earlier stages would build. In the two neighboring layers the neurons were connected directly to the neurons within the fully connected network. So what this implies is that every one of neurons in the fully connected layers can collect input data components over time that would help it to predict the right class value in the softmax layer afterwards are shown in figure (3.8). The objective

from this layer is to summarize the weights of the features from the prior layers, and show the value of per class, as was explained previously in the previous chapter in part (2.18.1.5). In proposed approach 15 classes that will be produced from this process because according to the data used for training and determining the number of classes that have been extracted from this stage, these classes will be in the form of value for each class linked to the fully connected layer image and Softmax will be made for them in the last step as seen in figure (3.8).

7. Softmax Layer

The network's performance can be difficult to interpret, it is normal to finish the CNN with a softmax function in classification issues. The result values are between the [0, 1] range which is good because that can prevent binary classification and fit as many classes or measurements in the CNN model that used. The Softmax function equation was mentioned in the previous chapter in part (2.18.1.8) in equation (2.28). After extracting values of 15 classes of plant diseases in the fully connected step, a Softmax will be made for them, so that the class will be selected in each process and according to the features that were extracted through the previous layers that the images of plant diseases went through. Firstly extracting the exponential for the inputs, and then dividing the values by the sum of the exponential values in order to obtain probabilities through which to classify them as shown in the algorithm (3.8). So the greatest probability is the correct class that indicates the disease of the specific plant. According to these probabilities, the disease is classified among 15 classes of plant leaf diseases, as shown in figure (3.8).

Algorithm (3.8): Softmax layer function	
Input:	Fully Connected Layer Values
Z_i	// Fully Connected Layers Values
Output:	Softmax probabilities value for the classes of plant leaf diseases
Steps:	
Step(1):	Begin
Step(2):	Calculate the exponential for every input in fully connected layer $e^{z_i} \leftarrow e$
Step(3):	Calculate the exponential summation for 15 class of input fully connected layer $\sum_{j=1}^{15} e^{z_i}$
Step(4):	Calculate the softmax function(y_i) by using equation (2.28) for all classes after calculating the exponential for each class in step(2), and divide each of them by the sum of the 15 classes after calculating the exponential for them in step(3), to predicate a true class that has the highest probability.
Step (5):	Return softmax probabilities value for the classes of plant leaf diseases (y_i)
Step (6):	End algorithm

3.12 CNN Training

Training a network is a procedure of obtaining kernels in convolution layers and weights in fully connected layers that reduce differences on a training dataset between output predictions and specified area truth labels. As explained in the second chapter in section (2.18.2), this stage is one of the most important stages in CNN because learning comes through this stage so

that the network that has been built learns by extracting features from plant leaf disease images in order to learn from these features for each image to be distinguished on its basis. In the proposed approach also used the pre-trained network method, the network is trained and the training is saved for use later, which saves time and effort and not to be re-trained again. In the training stage, the data are labeled to each type of plant leaf diseases, and training is conducted on the foundation of these labeled, in respect of learning and training the network. The training stage, need three things for training, which is the training set that is obtained from the dataset, the layers in which the network was built, and the different training options that were made for training. Also, this structure has a very important function for evaluating the training process, which is the Loss function, which will also be used.

a. **Training Set**

The data is divided into two main sections, which are training data and testing data. The data is dividing using a function in the Matlab called "Split Each Label", this function divides the data according to the percentages determined by the user and randomly. It was randomly divided and not chained to obtain a better approach and data is randomly taken from the dataset to make detection later better and stronger. In the proposed approach the data was divided so that the training data would be 70% and the testing data is 30%, after testing all ratios, and this best ratio as it will be shown in the next chapter.

b. **Layers (designing)**

The layers are intended as the layers that were built in the construction of the CNN structure in the training stage, all of these layers that mentioned earlier will also be passed in order to extract the features and learn from them to be classified through these features. Training for all data for training will be passed through the layers that built.

c. **Training Option (training algorithm)**

In order for the training process to be completed, that need multiple options that were used in this process using Matlab program by parameters that are created in the training process these options are:

1. Stochastic Gradient Descent with Momentum (SGDM)

The method is utilized in the data training process, due to its good properties that fulfill the purpose of training. It is an optimization method utilized to train CNN and machine learning systems. SGDM is a tool that helps move vectors of gradients in the true directions and thus contributes to faster convergence. It is one of the most famous optimization algorithms and it is used to train several states of the art systems. They were explained in the previous chapter in section (2.19.3) and illustrated in equations (2.29) and (2.30).

2. Max Epoch

The “Epoch” is a metric of how many times all training vectors are once utilized to update the weights. Concurrently in one epoch in the learning algorithm, before weights are upgraded. The maximum number of epochs that used for training is 15 epochs. In addition, the training dataset divided to 15 epochs, and for each epoch, 112 iterations and the number of iterations was

1680 divided by 15 epochs. As in the following figure (3.11), which illustrates the training process, this image was taken from the proposed approach during network training upon execution.

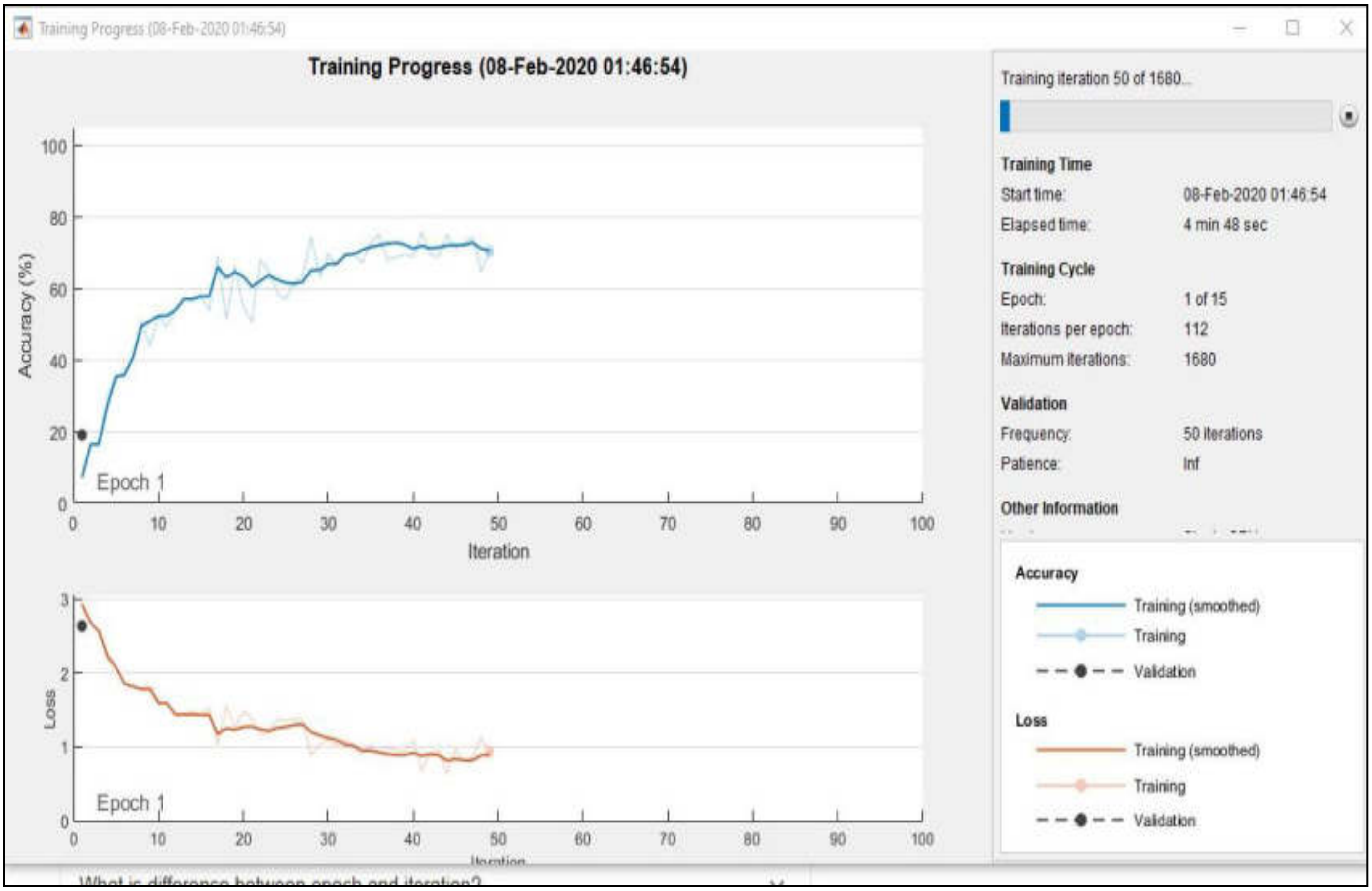


Figure 3.11: CNN during the training process

3. Shuffle

The data is shuffled to have various data for per batch, this command is the training data is shuffled. There are many options are: "never” If no shuffling is implemented if the option is "once" the data will be shuffled only once in the training. If the option is "every-epoch" the dataset will be shuffled before each training epoch. In the second proposed approach, "every-epoch" is used for shuffling option.

4. Plots

Plots to present when training, there are two options "training-progress" or "none" in (default); in the proposed approach was used "training progress" for plot option to see the progress of training.

5. Verbose

It is one of the training options and has two options, “True” or “False”. If this is set to “true”, the command window will be printed with the data on the training progress. If the option is "false" then command window will not be printed with the data on the training progress the default for this is "true", but in the proposed approach use the "false" option.

6. Validation Data

It’s the information to use throughout the training for validation. This may be an Image data store with categorical labels, a Mini-Batch data store with specified responses, and a table where unless image paths or images are in the first column, a cell array{X, Y} where X is a numerical array with the input data and Y is a return array. In the proposed approach, this option was used (testing set, dataset for the testing set. labels) for validation data. The testing set mean is the group that was used for the later testing process, while the dataset is a set of datasets that have taken as a label form.

7. Validation Patience

The number of times the validation loss can be greater than or equal to the initially smallest loss before the network training is finished. There are two choices either an "Inf" choice or a "positive" choice. The definition of the "Inf" choice is the automatic ending of training the network, and this choice is the default choice for training the network. The other choice, which is positive, means specifying a "positive" integer for stop network training. In the proposed approach that uses the "Inf" choice.

d. **Loss Function**

A loss function, also known as a cost function, can be calculated the consistency between the network's output estimates by forwarding propagation and assigned area truth labels, as shown in figure (3.11). It helps in the optimization of CNN parameters. The important purpose here is to reduce a convolution neural network's loss by optimizing its parameters (weights). The loss is measured utilizing loss function, by a convolution neural network combining the target (actual) result and the predicted value with errors. As explained in the second chapter in section (2.19.4), also the back-propagation that explained in the second chapter in section (2.19.5), the loss function and accuracy function was also calculated and displayed by using the confusion matrix as it will be presented in chapter 4.

3.13 CNN Testing

The testing is a dataset utilized to provide an impartial final design fit evaluation on the training set of data. In this stage, the approach uses the groups that were trained in the previous step that was trained in CNN, and the features were extracted by learning the network when the dataset passes from plant leaf diseases on this network. The dataset also used that was allocated to the testing process when it was divided, based on these dataset and the training data for testing; the classification process for plant leaf diseases is done. Before the testing stage, there is the training stage; this means that the network is trained when inserting any image of plant leaf diseases. The disease will be determined because the network was previously learned and trained. So the main difference between training and testing is test data that are unlabeled, unlike training data that is labeled. In the proposed approach, uses this function in Matlab, CLASS = classify (Sample, Training Group), this

categorizes every row of the dataset to one of the training groups. Sample and training must be arrays with the same column size. Training the Group is a group variable and the specific values determine groups, and each component determines the group to which the associated training row belongs. Here we have reached to the last, which is the classification and detection of plant leaf diseases.

3.14 Plant Leaf Diseases Detection in the Second Proposed Approach (Performance Measurement)

In this stage, the offline model was applied to conduct the detection process, based on the previous steps and on the trained network, and a user interface was created for the detection process for the plants leaf diseases as it will be presented in the next chapter.

Chapter four

Implementation

&

Experimental Results

Chapter Four
Implementation and Experimental Results

4.1 Introduction

In this chapter, the implementation and experimental results of the proposed approach were presented and described. This proposed approach "detection of plant leaf diseases using image processing and deep learning techniques" is divided into two different sub approaches according to the used algorithms. The first proposed approach that was previously explained consists of several stages; the outcomes of these stages will be presented with the final results of classification using the MSVM algorithm, in addition to the results of the second approach which is based on CNN algorithm. The results of these two approaches will be presented and a comparison will be made between the results of the two approaches. Moreover, a comparison with related works will be presented. The same dataset was used in the two sub approaches.

4.2 Implementation Environment

The proposed approach is implemented and evaluated by using MatlabR2018b and the proposed approach performed on the windows10 O.S, and the platform Intel Core i5, RAM 4 GB.

4.3 Dataset Acquisition (Plant Leaf Diseases Images)

The collected data consists of 20636 images related to 15 different categories are shown in table (4.1) which includes the number of images of these plants with their diseases. Every downloaded image is recorded to RGB colour space at 256 * 256 sizes and saved in JPG format in an uncompressed file named (Plant Village) and stored on the computer. A small portion of these images that shown in figure (4.1), which includes a sample of images for the different types of plants with the types of diseases that affect them, another sample of the dataset is presented in appendix A.

1. Pepper__bell___Bacterial_spot

2. Pepper__bell___healthy

3. Potato Early blight

4. Potato Late blight

5. Potato Healthy

6. Tomato Target Spot

7. Tomato Mosaic Virus

8. Tomato Yellow Leaf Curl Virus

9. Tomato Bacterial spot

10. Tomato Early blight

11. Tomato healthy

12. Tomato Late blight

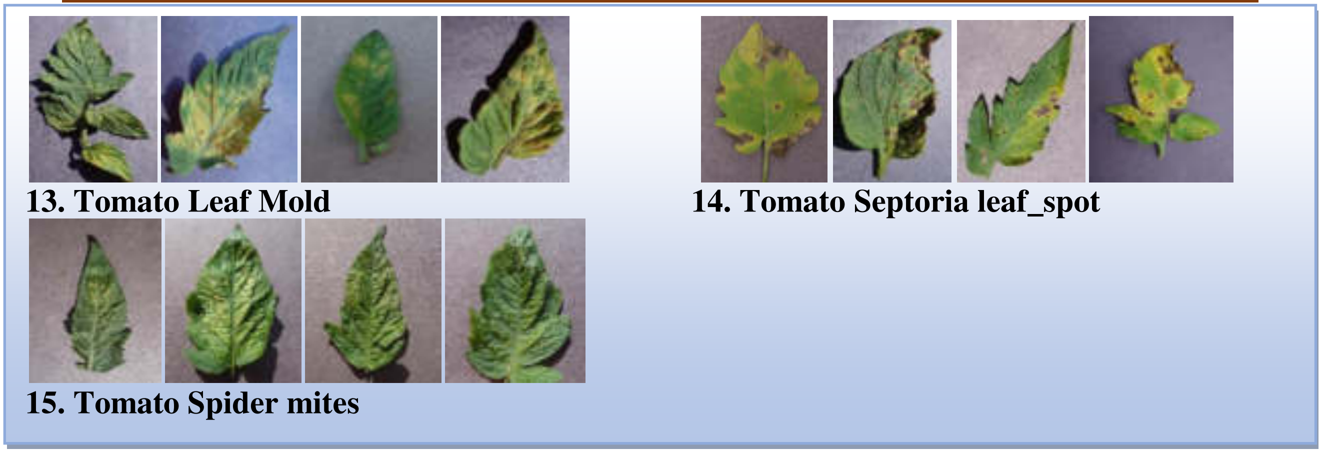


Figure 4.1: Samples from the dataset for (plant leaf diseases)

Table 4.1: List of crops and their disease from (Plant Village dataset)

No.	The plants	Kinds of diseases	Number of images	Totals of image numbers
1.	Pepper bell	Bacterial spot	997	2474
2.		Healthy	1477	
3.	Potatoes	Early blight	1000	2152
4.		Late blight	1000	
5.		Healthy	152	
6.	Tomatoes	Bacterial spot	2127	16010
7.		Early blight	1000	
8.		Late blight	1908	
9.		Leaf Mold	952	
10.		Tomato_Septoria_leaf_spot	1771	
11.		Tomato_Spider_mites	1676	
12.		Tomato__Target_Spot	1404	
13.		Tomato_YellowLeaf_Curl_Virus	3208	
14.		Tomato_mosaic_virus	373	
15.		Healthy	1591	
16.	Total			20636

The used dataset for the proposed approach consists (as mentioned before) of two different approaches, and the results will be presented in order: firstly the results of the first proposed approach, and secondly the results of the second proposed approach, as well as, the comparison between them.

4.4 Evaluation of First Proposed Approach

The first proposed approach that was explained in the third chapter, which consists of several stages, each stage and its results, up to the final results will be shown, it the classification stage with the MSVM algorithm. The first stage is the pre-processing stage.

4.4.1 Result of Image Pre-Processing

At this stage, the images will be converted from RGB to L*A*B*colour space.

4.4.1.1 Result of Images Conversion from RGB to LAB

The conversion was done in order to obtain better results in the process of image segmentation which depends on colour illumination and the conversion process is carried out by applying algorithm (3.1) in chapter three; figure (4.2) shows a sample of converted images of infected plant leaves which are converted from RGB to L*A*B* colour space.

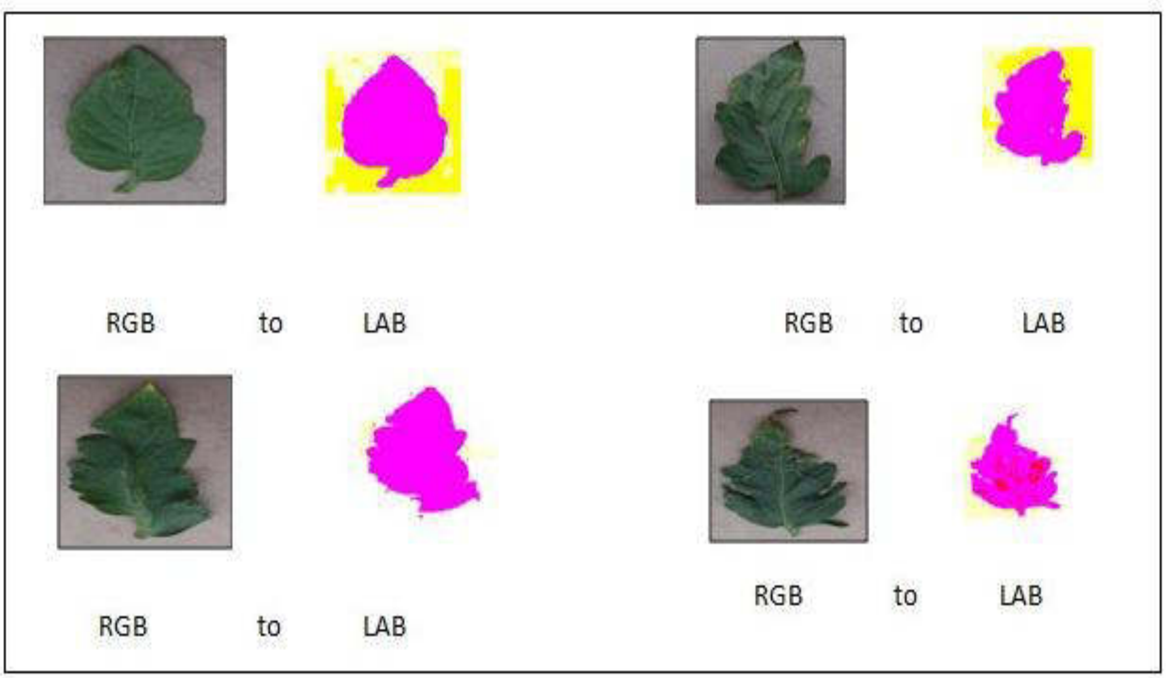


Figure 4.2: RGB to L*A*B*conversion

4.4.2 Result of Image Segmentation

At this stage, the K-means clustering algorithm was applied according to algorithm (3.2) in chapter three. Figure (4.3) shows a sample of some images from the proposed approach after applying the image segmentation process, where, (a) shows the original image of the affected leaf, and (b) shows the image for which segmentation was made.

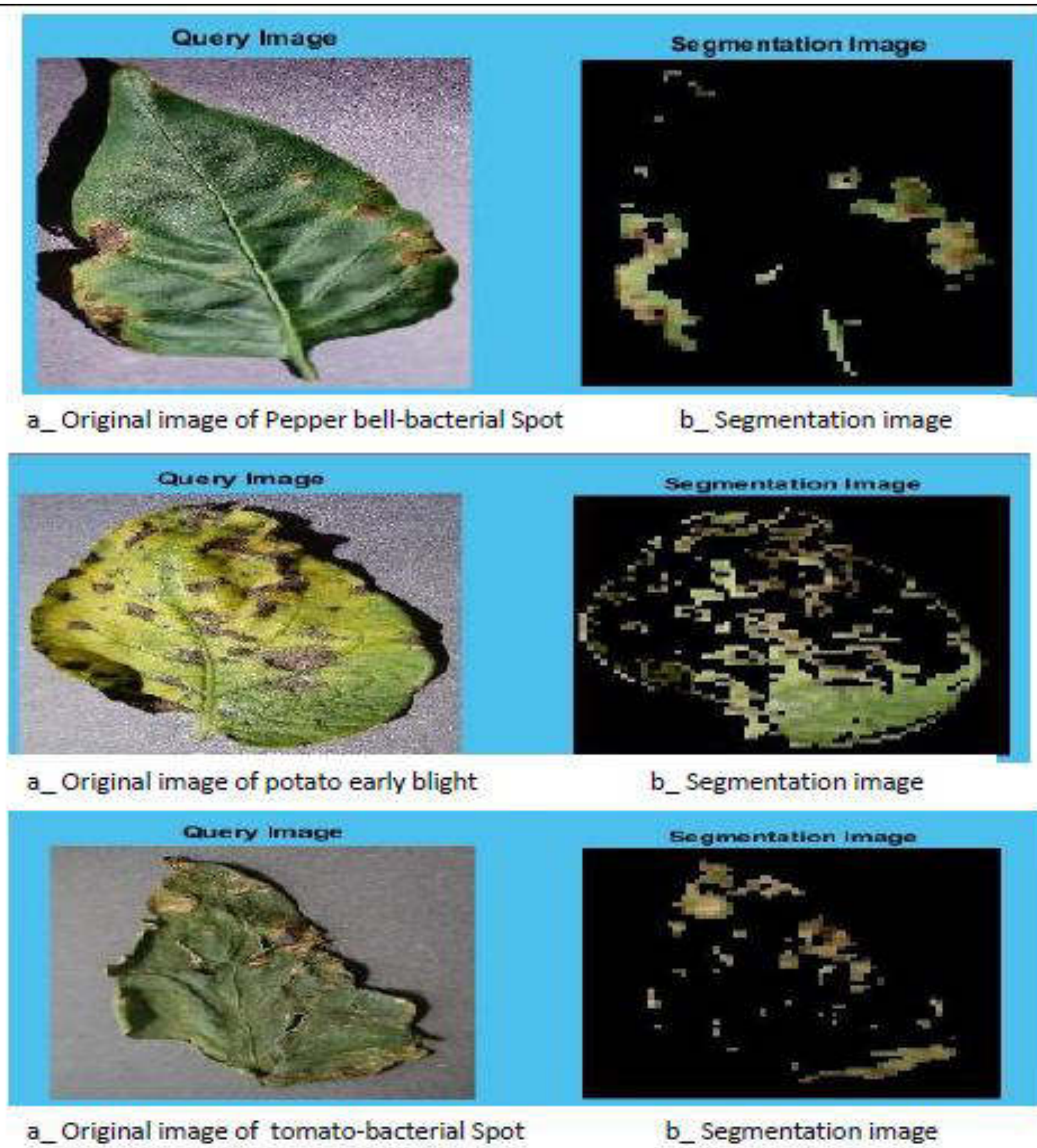


Figure 4.3: The images segmentation from the proposed approach

4.4.3 Image Post Processing

At this stage, in the first proposed approach, a conversion of images from RGB to Grayscale was applied.

4.4.3.1 Result of Images Conversion from RGB to Gray Scale

The conversion process was explained in the previous chapter according to algorithm (3.3). A sample of images that were converted from RGB to gray is shown in figure (4.4).

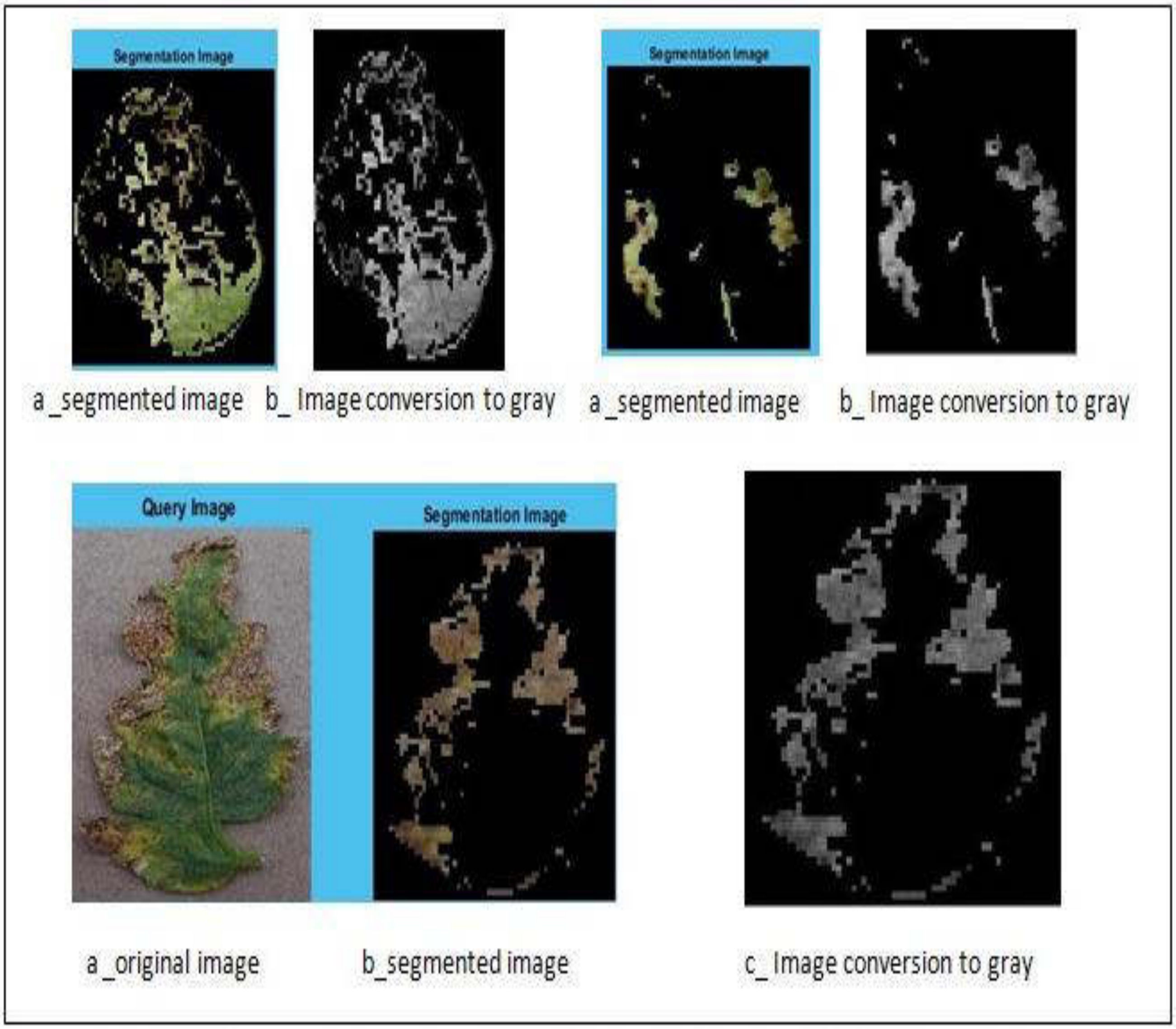


Figure 4.4: Conversion of plant leaf segmented images to the gray scale

In figure 4.4 at the top, (a) segmented image, while (b) the image after it converted to gray scale, and at the bottom, (a) the original image, (b) is the segmented image, and (c) is the image after it converted to the gray scale.

4.4.4 Result of Feature Extraction

Processes continue after the image segmentation process, the process of extracting features that have been explained in the previous chapter in the first proposed approach comes with a feature that has been extracted. Seven features were extracted directly from the segmented images, while the remaining four features were extracted using the GLCM method after conversion to grayscale. Since having the number of images in the dataset is (20636), then each of these images has 11 different features, because of the data size is very large here will show a sample of these features, so in the table (4.2) which displays a samples from the proposed approach of these features were taken for 25 images randomly and each image 11 features as explained previously.

Table 4.2: The samples of features extracted from 25 random images

No	Contrast	Correlation	Energy	Homogeneity	Mean	Standard deviation	Entropy	RMS	Variance	Kurtosis	Skewnees
1	0.592279	0.822866	0.547288	0.909468	22.30307	43.41168	2.507427	7.21722	1590.343	4.971434	1.773529
2	0.225444	0.979807	0.337518	0.950978	83.78798	83.41042	4.520835	11.19611	4882.925	1.079152	0.062012
3	0.337347	0.958763	0.278958	0.897901	70.72915	73.2363	4.542054	10.77996	2946.677	1.238408	0.207967
4	0.315656	0.888505	0.435247	0.942614	26.15154	41.05059	3.413462	8.747651	1362.228	3.700598	1.370876
5	0.329458	0.972578	0.373749	0.934955	72.93479	87.3299	3.817506	9.903693	5364.764	1.278392	0.442379
6	0.238297	0.911644	0.47547	0.950109	21.41608	39.57384	3.138376	7.669268	1270.672	5.433158	1.842251
7	0.379917	0.974689	0.344319	0.924184	87.05257	95.4986	3.846736	10.41606	6801.56	1.10731	0.219129
8	0.34565	0.751279	0.720603	0.943494	10.17303	29.10085	1.586408	5.169401	753.7774	12.50081	3.104407
9	0.197809	0.979349	0.318075	0.943423	72.76216	77.8968	4.247431	10.5748	4228.742	1.216121	0.240017
10	0.418735	0.957801	0.390409	0.88534	62.81029	78.48908	3.996628	9.821885	4137.665	1.462108	0.557442
11	0.265119	0.982934	0.395021	0.96118	83.21309	97.10849	3.591591	9.826572	6151.366	1.152407	0.336583
12	0.704871	0.822504	0.43892	0.878787	28.56206	47.70847	3.224716	8.270476	1914.899	3.932574	1.464783
13	0.301716	0.964993	0.322869	0.913481	67.0609	74.65947	4.242246	10.59386	4150.188	1.360211	0.347292
14	0.537469	0.870056	0.439516	0.915163	32.6115	50.28324	3.250256	8.529666	2188.044	2.734859	1.131236
15	0.509038	0.965678	0.216595	0.848531	124.7131	93.86916	5.009729	12.49284	5514.971	1.368477	-0.51998
16	0.343153	0.965619	0.309504	0.924694	69.69548	79.32045	4.592327	11.02739	5267.97	1.411427	0.459822

17	0.465717	0.967869	0.258427	0.870519	107.5883	93.64422	4.688306	11.71687	5325.328	1.132689	-0.22664
18	0.359972	0.979614	0.34462	0.924208	104.3031	101.417	4.373107	11.16216	6596.563	1.058172	-0.00961
19	1.248346	0.808367	0.389881	0.844026	43.25243	62.01927	3.641284	8.591095	2905.44	2.347698	0.95959
20	0.303738	0.883571	0.552087	0.942945	20.0421	38.67695	2.617514	6.548632	1091.732	5.017573	1.795916
21	0.893367	0.870878	0.426169	0.892464	36.64278	61.58473	3.434729	8.277431	2648.548	3.435075	1.388979
22	0.237286	0.89399	0.480465	0.943465	22.60989	37.50563	2.986898	8.161654	1186.151	3.577993	1.379902
23	0.452237	0.883011	0.582061	0.945811	22.08996	45.78688	2.420925	6.785329	1812.695	5.688844	1.962699
24	0.665196	0.834113	0.558849	0.905698	23.27353	47.06804	2.512007	6.985741	1969.571	5.38261	1.88723
25	0.355729	0.855047	0.63254	0.940077	17.34054	38.47002	2.092487	6.385941	1343.877	6.522064	2.129738

Figure (4.5) shows example from the proposed approach, where eleven features were extracted from one of the pepper bell plant images after conducting operations on them, according to the equations that mentioned in the previous chapters. These features will be used later in classification process.

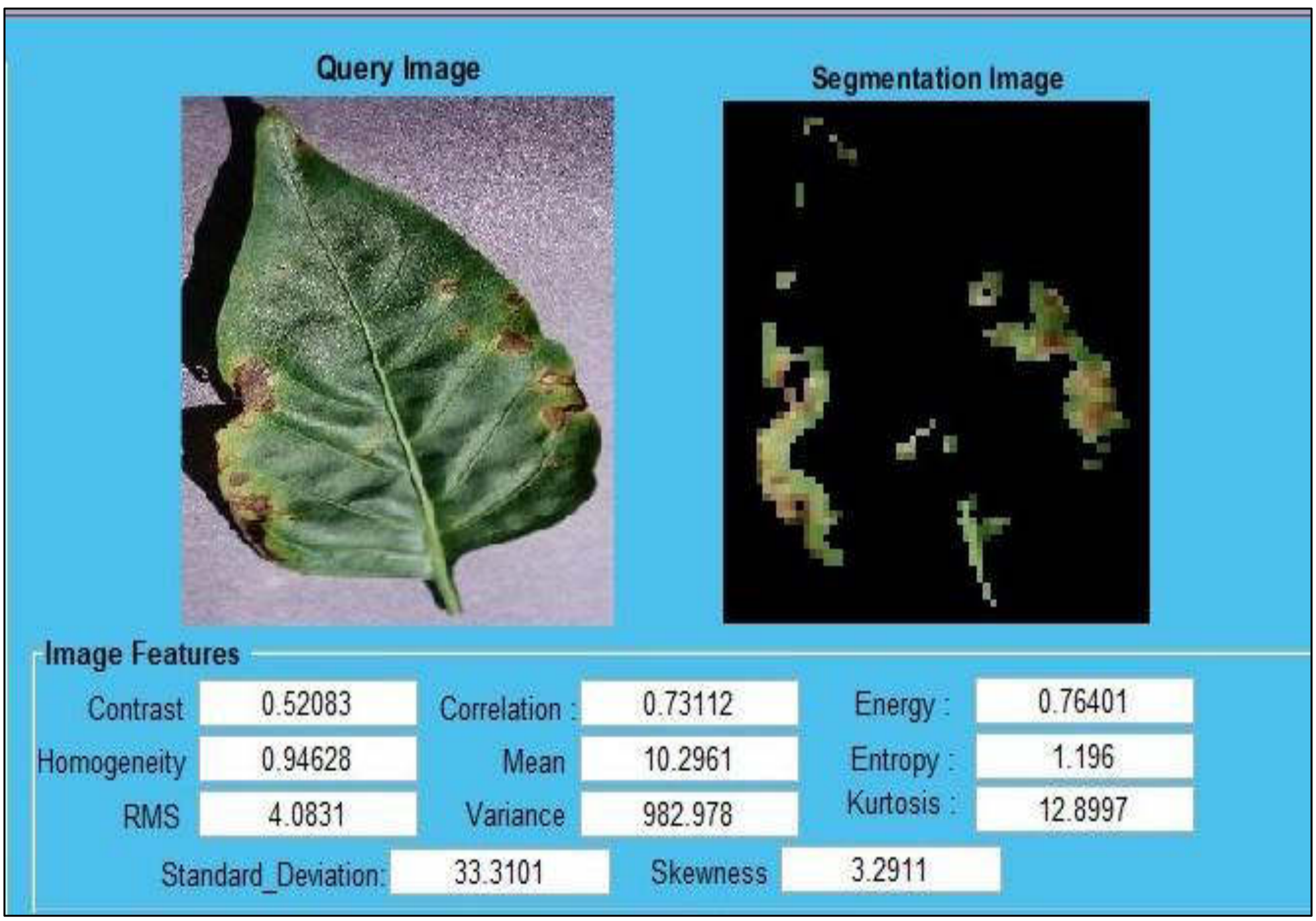


Figure 4.5: Example of features extraction from the pepper plant image

4.4.5 Result of Training and Testing Using MSVM Algorithm

After the process of extracting features, the approach reached the classification stage. In this stage, the diseases of plant leaves are classified according to the features that were extracted, and as the MSVM algorithm was previously explained in the algorithm (3.5). In order to complete the classification process, the approach needs two stages that have been explained previously, which are the training stage and the testing stage. The results of these two stages will be displayed in order to then detect and classify the diseases of plant leaves. The confusion matrix is used to display the accuracy of the training and testing results for each class and the overall accuracy of all classes. The training results will be displayed first and then the test results.

A-Result of MSVM Training

This operation was explained previously in section (3.8.A), where the training process for images is done in an MSVM method after performing the operations and to detect the disease in the testing stage later. The display of the results using the confusion matrix as in figure (4.6), which shows the accuracy of the training results for each class, and also displays at the top of the matrix the average total accuracy of all classes in the training stage is equal to (69.33). The overall accuracy and accuracy of each class in training was calculated according to that was mentioned in the second chapter in section (2.19) and according to the equations (2.31), (2.32) and (2.33). In the following equations, an example from the proposed approach for calculating the accuracy of the first class in this matrix is the bacterial spot disease of pepper bell plant, as well as how the overall accuracy rate was calculated for all classes in the training stage:

$$\textit{The correct accuracy for pepper bell bacterial spot class} = \frac{361}{536} * 100\% = 67.4$$

$$\textit{The incorrect accuracy for pepper bacterial spot class} = \frac{175}{536} * 100\ \% = 32.6\%$$

The rate the overall performance Accuracy for training = (67.4 + 58 + 100 + 100 + 83.1 + 56.2 + 58.5 + 81.4 + 72 + 69.2 + 60.9 + 60.8 + 65.2 + 55 + 62.2) / 15 = 69.33% .

Thus, the process of calculating the accuracy of all classes was done, as in figure (4.6), which shows the confusion matrix of the training process, to clarify the numbers that are in figure more, they will be shown in the comparison table (4.5), it will be presented later.

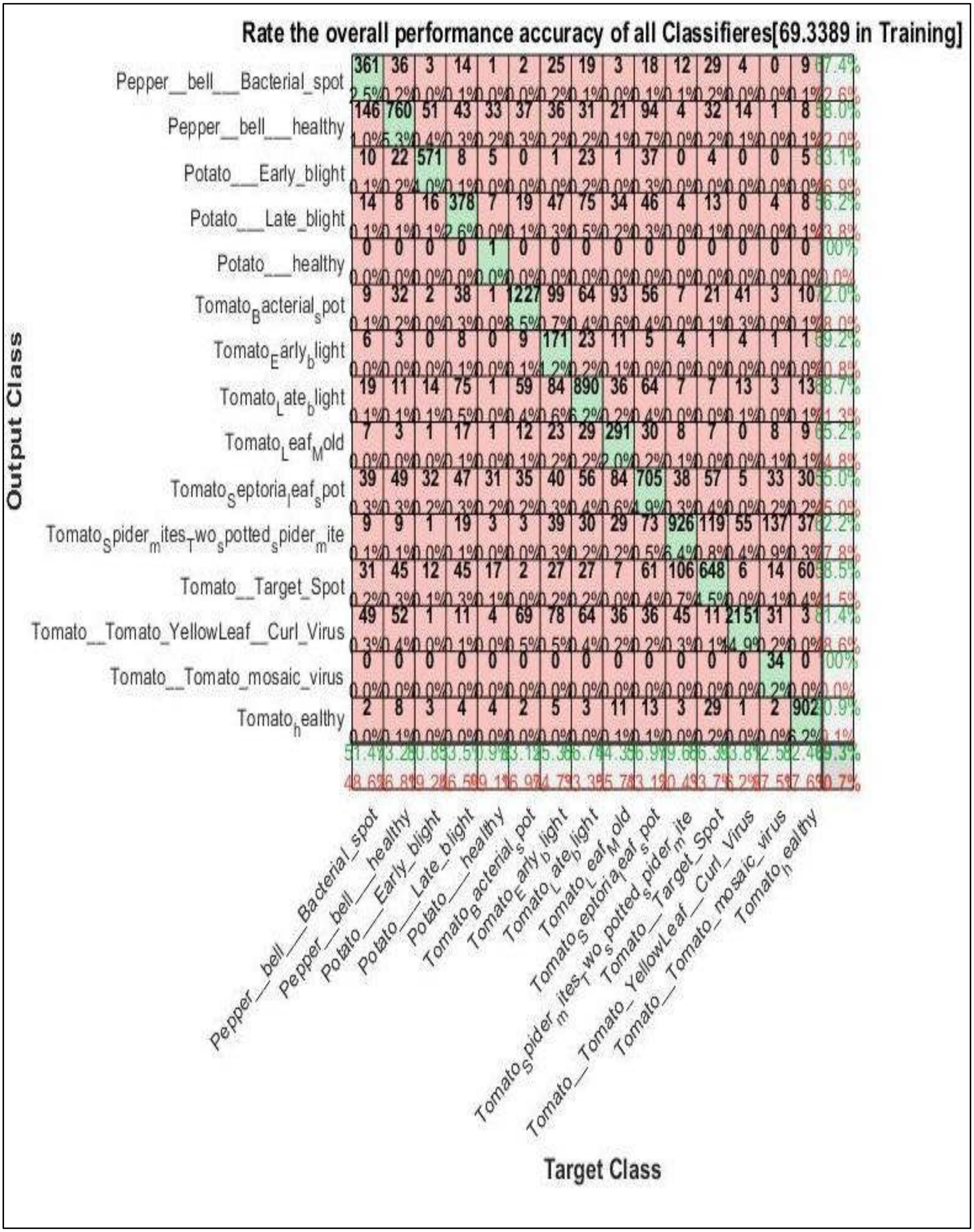


Figure 4.6: Confusion matrix for MSVM accuracy for training data.

B- Result of MSVM Testing

This stage is an examination of the approach by testing it with the remainder of the data that is not labeled in order to classify the diseases of plant leaves as explained in the previous chapter. At this stage also the results were presented in the form of a confusion matrix that shows the accuracy of each class in the testing stage also, the average overall accuracy of all classes is equal to (61.79), it is calculated in the same way as calculating the total accuracy of the training stage as the following equation:

$The\ rate\ the\ overall\ performance\ Accuracy\ for\ testing = (59 + 47.3 + 72.7 + 44.1 + 80 + 67.1 + 56 + 59.7 + 53.5 + 50.1 + 55.8 + 49.3 + 75.3 + 66.7 + 88.9) / 15 = 61.79\%$.

The following figure (4.7) shows the confusion matrix that shows the accuracy for each class of the 15 class used with the average total accuracy in testing at the top of the matrix, to clarify the numbers that are in figure more, they will be shown in the comparison table (4.5), it will be presented later.

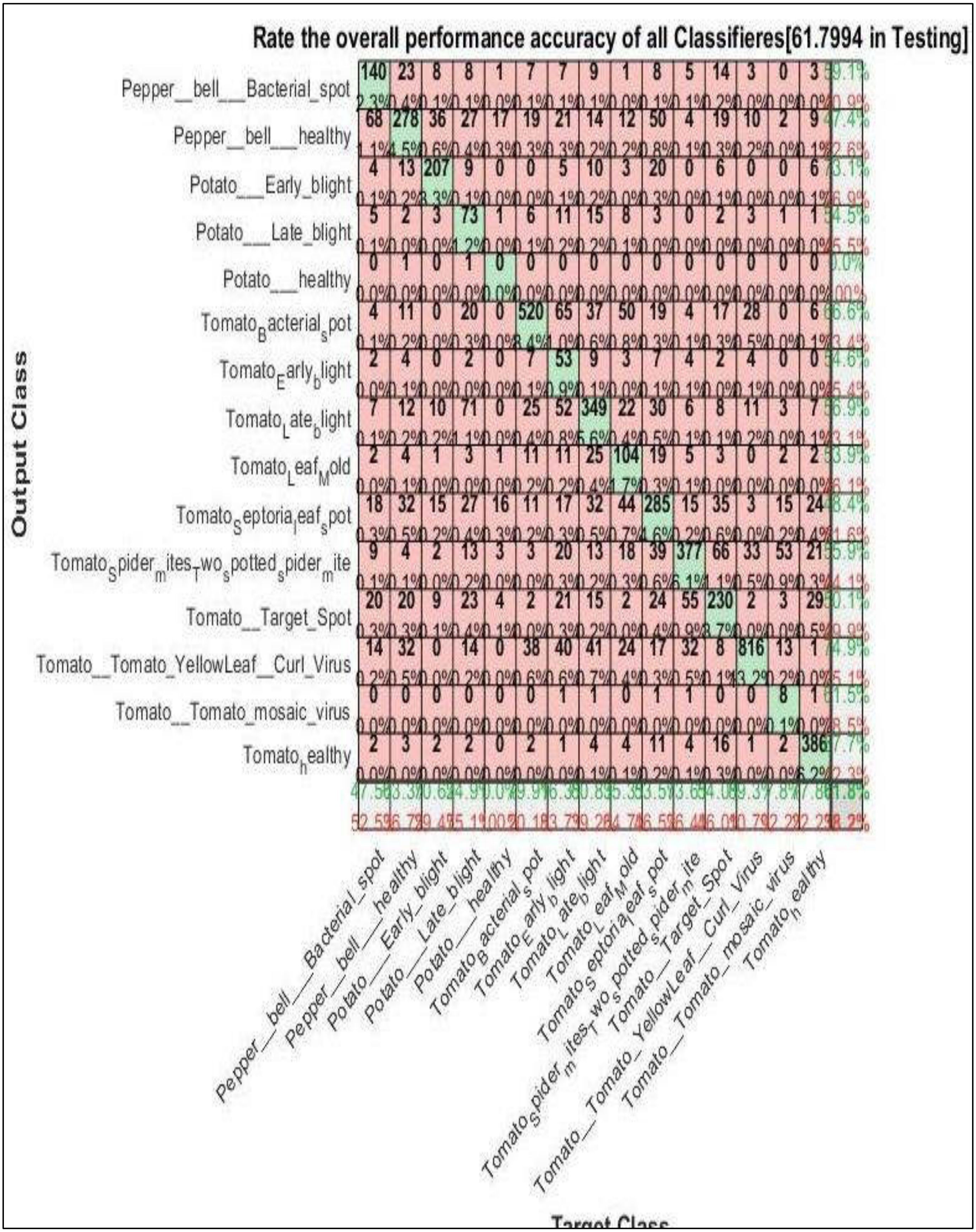
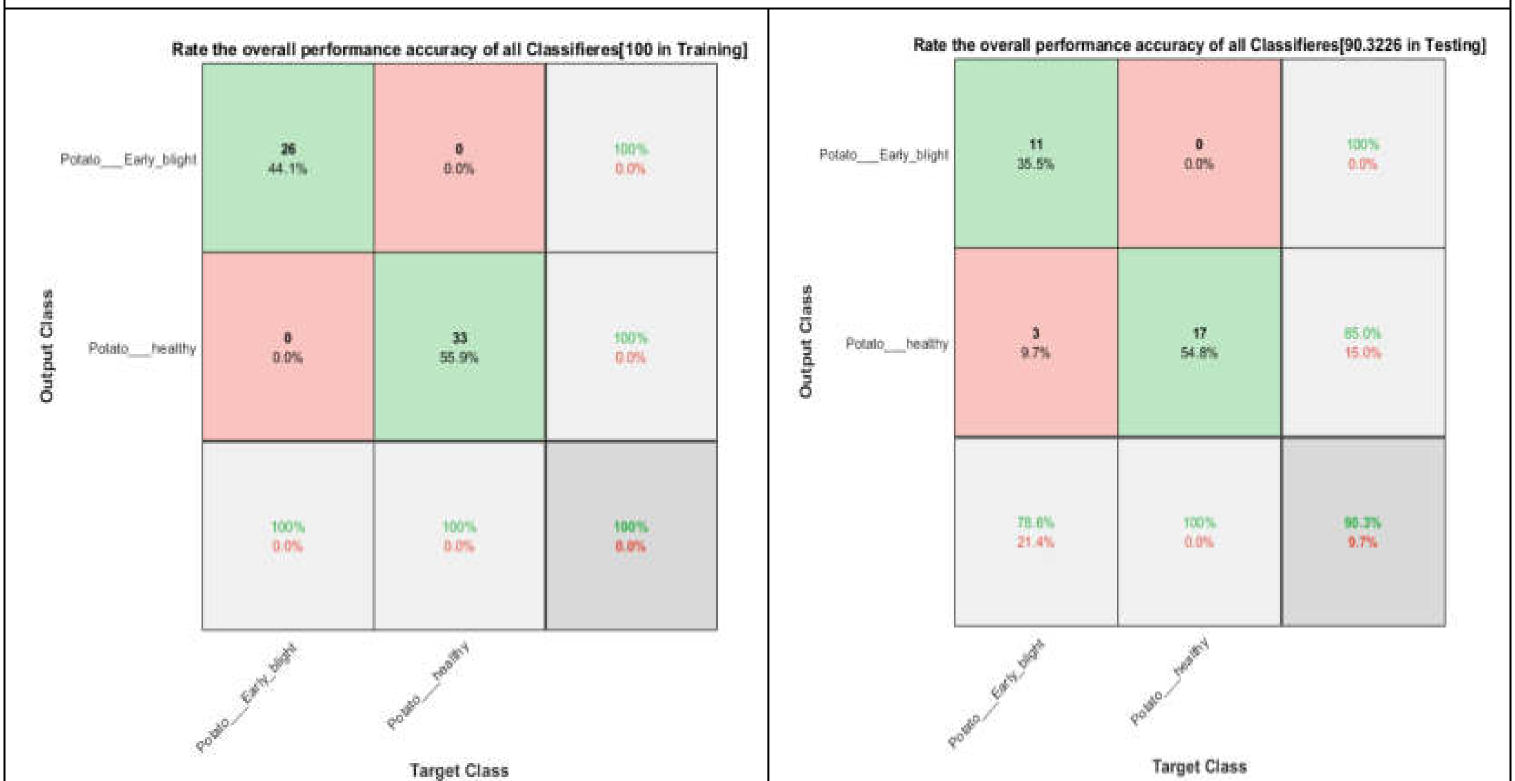


Figure 4.7: Confusion matrix for MSVM performance accuracy for testing data.

It should be note that the main reasons for low accuracy here are: the first reason is a large number of classes, there are 15 classes in the proposed approach, usually the MSVM algorithm is ideal is when having two classes but when to use more than two classes, must use a Multi-Class SVM. After research on this topic, it was found that the accuracy is inversely proportional to the number of classes in the MSVM algorithm, and this is one of the disadvantages of this algorithm. Also because using non-linear SVM, the number of classes is more than two because of this, it will be a great overlap in the data as have observed in the Confusion matrix, in the training, also in testing stages, in addition, the large data used in this approach also from these reasons, which led to an exacerbation of accuracy in the training and testing stages. This is the reasons that led to the search for another algorithm to detect plant leaf diseases with greater accuracy. In order to prove these reasons that led to a lack of accuracy in the training and testing stages, the approach was trained and testing on two classes only for classification and high accuracy was obtained in the training and testing stages. These results were presented to show the reason for the lack of accuracy in relation to the first proposed approach, and how accuracy increased when only two classes are classified and smaller dataset, as in figure (4.8). In the figure (a) it displays the confusion Matrix, to the tomato plant training and testing process and the accuracy rates was obtained in training (100%) and in the testing (100%), and in (b) the two confusion matrices of the potato plant were displayed and a high accuracy rate was obtained also in training (100%) and in the testing (90.32%).

a_ The Confusion Matrix for the tomato leaf diseases training and testing



b The Confusion Matrix for the potato leaf diseases training and testing

Figure 4.8: Confusion matrices for classifying two classes of leaf diseases tomato and potato plants

Before performing the detection process, it should be noted here that an important note is that the kernel function that is relied upon is (polynomial function) due to the experience of the rest of the kernel functions and the best results were obtained with this function for classification the 15 classes with the highest accuracy in the training and testing stages as in the table (4.3), which shows a comparison between the types of kernel functions that were used in terms of the overall accuracy in training and testing.

Table4.3: Comparison of different kernel functions for the overall accuracy rate

No.	Kernel functions	The overall accuracy rate for all classes in training %	The overall accuracy rate for all classes in testing %
1.	**Polynomial function**	**69.33%**	**61.79%**
2.	Radial basis function (RBF)	63.22 %	54.53 %
3.	Linear function	43.68%	42.10 %

4.4.6 Plant Leaf Diseases Detection in the First Proposed Approach (Using A MSVM Algorithm)

Following conducting the training and testing processes of the approach, the process of detection of plant diseases is carried out. The detection mechanism in the first proposed approach using the multi-class MSVM algorithm for classification and after conducting all the previous operations that were mentioned, this disease is detected and classified, but the accuracy in detection is not high due to the lack of accuracy of the test and training as well, and the reasons have been explained, and for this reason, a second proposed approach has been built that is better than the first approach. Here, interfaces will be displayed from the first proposed approach that shows the detection process for pepper bacterial spot, in the form of stages, each of these stages has its own procedure, in order to facilitate the use of the approach for the user, as shown in the figure (4.9) which shows the first approach for plants leaf

detection using the MSVM algorithm with all the stages shown to perform the classification process, from the image load button to the detection button.

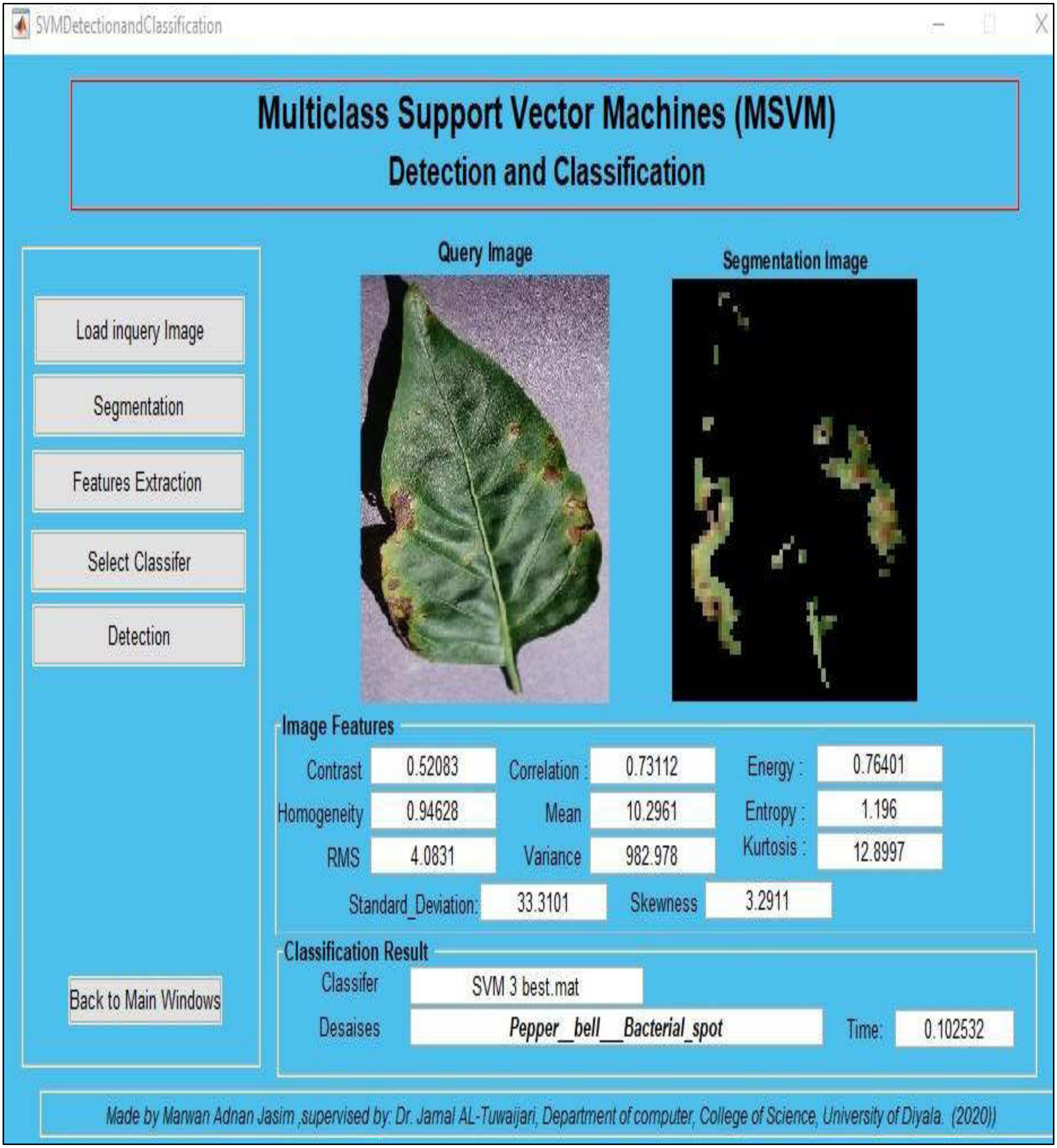


Figure 4.9: The interface of the first proposed approach (Plant leaf diseases detection using multiclass MSVM algorithm)

4.5 Evaluation of the Second Proposed Approach

The lack of accuracy and the weak results that obtained in the first proposed approach made our seek for a second approach and be better than the first proposed approach, and this actually happened, here the results obtained from this approach will be presented through deep learning techniques, using the CNN algorithm. As this approach was explained in the previous chapter, which consists of multiple stages, the difference in this approach is that the convolutional neural network is that performs these stages, it means the process of extracting features until the classification process. That means just inserting images of diseased plant leaves on the network, and in the outputs, this is revealed and classified disease that is why it was mentioned in this approach that there are only the end results obtained, there are no results for each stage as in the first proposed approach.

The first stage is the process of resizing images from 256 * 256 to 128 * 128. Then these pictures enter the network in two stages, which are the training stage and the testing stage. As in figure (4.10) which shows an interface from the second proposed approach that clarifies the process of entering the dataset with its division into the testing and training stages, in order to complete the training and testing operations, after which the trained network is saved so that it can be used in the next stage, which is the stage of detection of diseased plant leaf diseases. The trained network is saved in order not to repeat the training process at each detection stage, thus saving time for each training process.

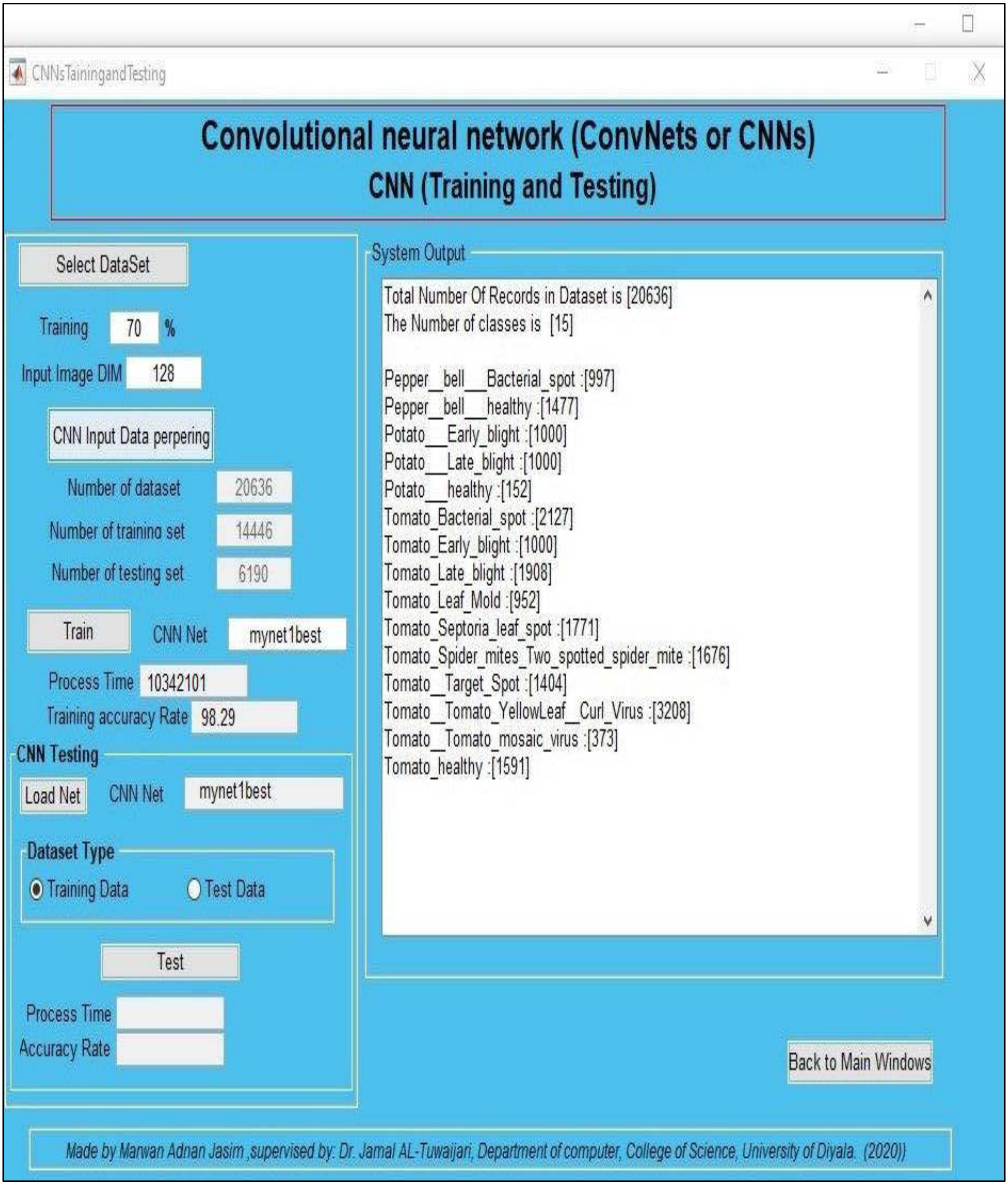


Figure 4.10: The interface of the second proposed approach for displaying the dataset selection and dividing processes for training and testing.

4.5.1 Result of the CNN Training

At this stage, the network training process was conducted to learn about the features of each disease and detect it at the detection stage. When the network is trained and all images pass on the CNN of all layers in order to teach this network, this is the main purpose of the training process. The main problem that occurred during the training is that it takes a long time, and also the speed of the computer and its properties have a big role in the time spent on the training the network. This stage was explained previously; here the results of the training process will be presented. These results are presented in the confusion matrix as well as the figure (4.11) that shows the results of each class of the fifteen classes of plant leaf diseases, as well as, the overall accuracy rate is calculated by summing the accuracy of the 15 class are shown in the confusion matrix and dividing it by its number which is 15 and its result is (98.29%).

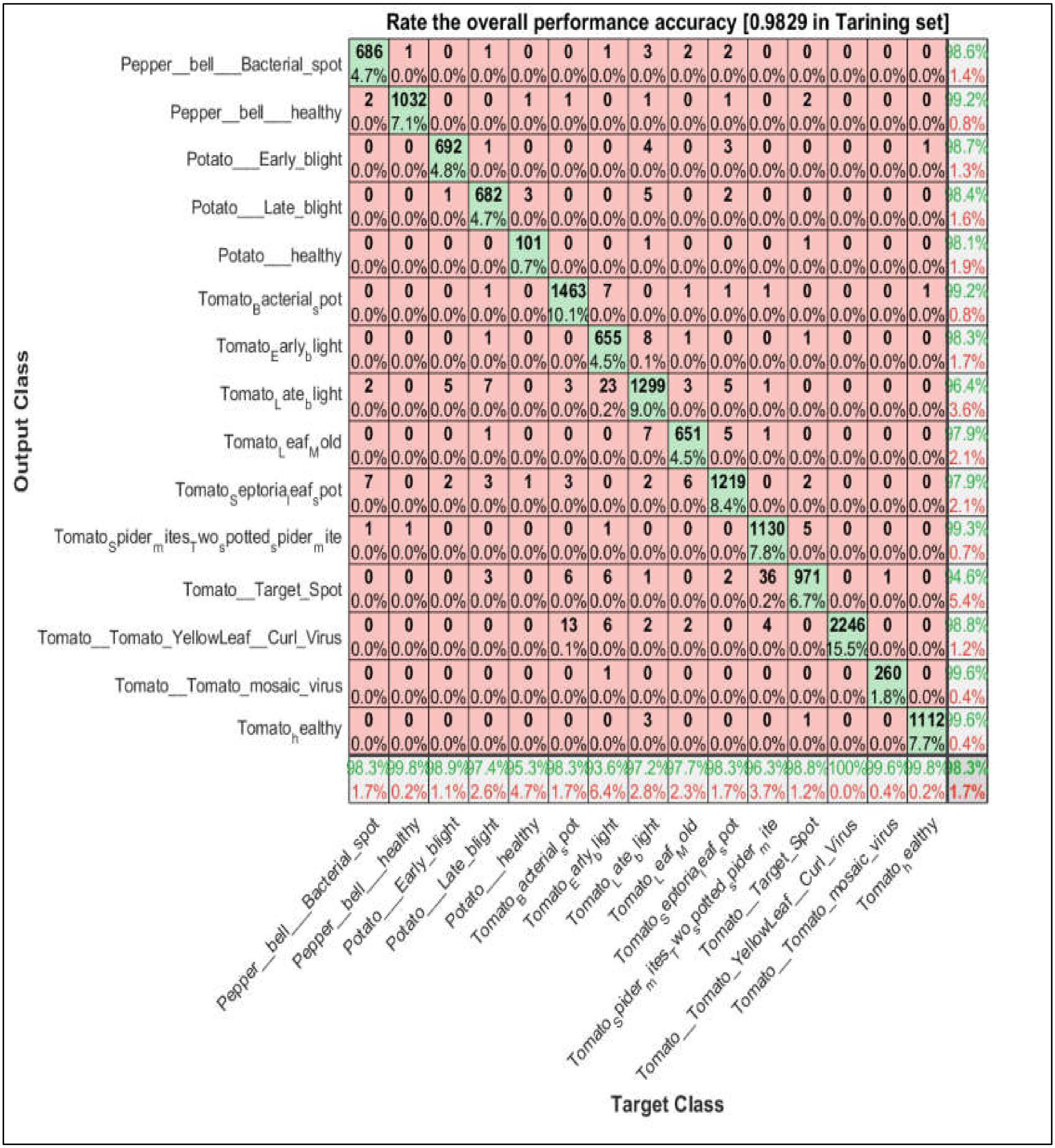


Figure 4.11: Confusion matrix for CNN performance accuracy for training data.

In the training process also, there is the process to calculate the consistency between the network's output estimates by forwarding propagation

and assigned area truth labels by using the loss function that has been explained in the previous two chapters. In figure (4.12) a chart will be displayed the loss function and the accuracy. The loss function expressed in the red line, which is descending to the bottom, in contrast to the accuracy expressed in the blue line that goes from the bottom to the up. In addition, the number of epochs was used in the training process is 15 epochs, as for the number of iterations is 1680 for each epoch 112 iterations, to extract the total number of iterations by multiply 15 * 112 and the result was 1680 iterations.

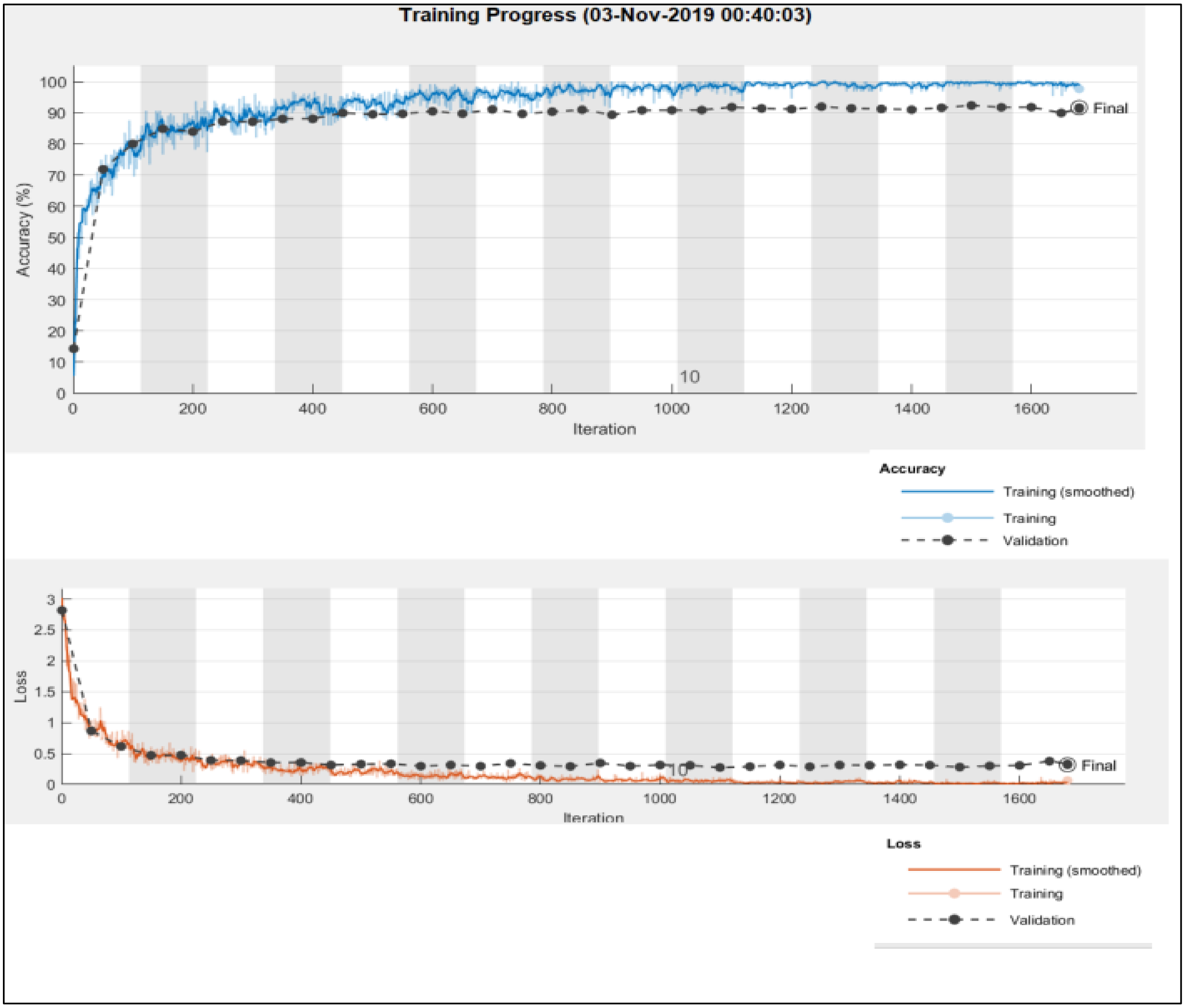


Figure 4.12: The Training progress, accuracy, and Loss function.

4.5.2 Result of the CNN Testing

In this stage, the results of the testing were displayed. This process was explained in the previous chapter and how the convolutional neural network is tested in order to be used for the process of detection of plant leaf diseases. In the testing stage, excellent results were obtained and presented as a confusion matrix as well as in figure (4.13), which shows the results of the accuracy of each class of the 15 classes in the testing stage, to clarify the numbers that are in figure more, they will be shown in the comparison table (4.5), it will be presented later. An example from the proposed approach for calculating the accuracy of the first class in this matrix is the bacterial spot disease of pepper bell plant, as well as how the overall accuracy rate was calculated for all classes in the testing stage.

The correct accuracy for pepper bell bacterial spot class $= 304/396 * 100\% = 97.4$

The incorrect accuracy for pepper bell bacterial spot $= 8/304 * 100\,\% = 2.6\%$

Thus the accuracy was calculated for the rest of the classes, and also the rate overall accuracy in the testing process that is (98.029) which is at the top of the confusion matrix, It is calculated as follows:

$$The\ rate\ the\ overall\ performance\ accuracy\ for\ testing = (97.4\% + 99.3\% + 99.3\% + 99\% + 93.8\,\% + 99.4\,\% + 96.9\% + 97.8\% + 97.2\% + 97\,\% + 99.6\,\% + 96.9\% + 98.4\% + 99.1\% + 99.6\%)\,/\,15 = 98.02\,\%$$

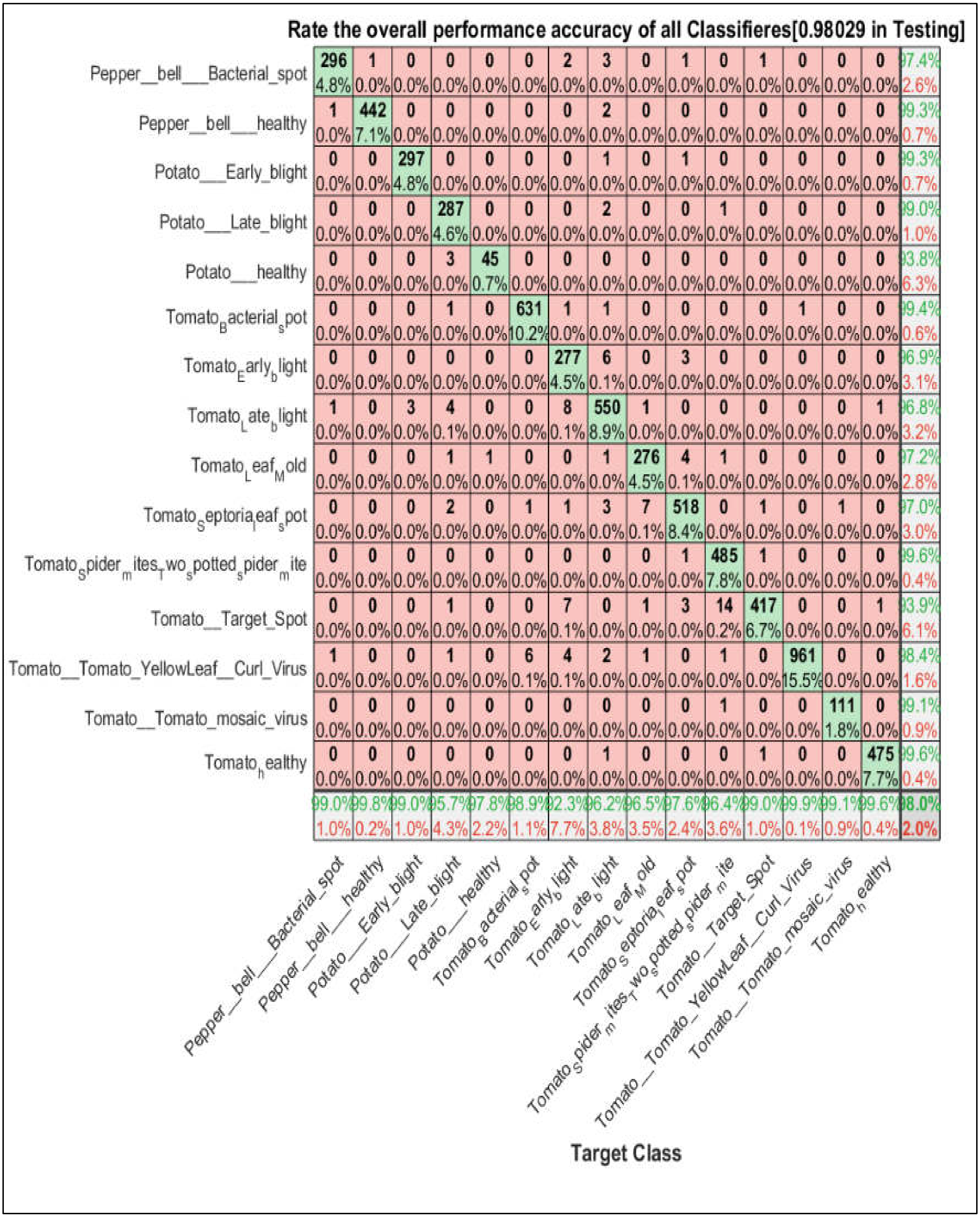


Figure 4.13: Confusion matrix for CNN performance accuracy for testing data

An important note that must be mentioned, the process of dividing the dataset, which was relied upon in our work, is 70% for the training dataset and 30% for test dataset. This division process is not arbitrary, but all data rates for training and testing have been tried and start with 50% only for training data collection and 50% for testing, until 90% of the training data versus 10% of the testing data. The highest percentage and the best result obtained were the 70% for training and 30% for the set of the testing dataset, especially for the testing process that depends on it in the classification and detection process of plant leaf diseases, are shown in table (4.4) which illustrates these percentages versus total accuracy training data and test data.

Table 4.4: Comparison of the different ratio of the size of data in training and testing for the overall accuracy rate

No.	Percentage of training dataset and testing dataset %	The overall accuracy rate for all classes in training %	The overall accuracy rate for all classes in testing %
1.	50% training _ 50% testing	99.72%	91.81%
2.	60% training _ 40% testing	99.71 %	9294 %
3.	**70% training _ 30% testing**	**98.29 %**	**98.02 %**
4.	80% training _ 20% testing	99.91 %	94.25 %
5.	90% training _ 10% testing	99.95 %	95.51 %

4.5.3 Plant Leaf Diseases Detection in the Second Proposed Approach (Using A CNN Algorithm)

The last stage in the second proposed approach is the detection stage of plant leaf diseases, which is the most important stage of the approach. When the network has been trained and tested so that any image that is chosen will be classified according to the 15 classes of different diseases, furthermore after obtaining high and excellent results in the training and testing, thus detection accuracy will be very high and the errors are almost non-existent or very rare.

In the figure (4.14) that showed interface of the second proposed approach used the same image that was used in the first proposed approach, which the pepper plant leaf infected with bacterial spot disease that was detected and classified using the convolutional neural network algorithm, and selecting the classifier network that previously trained, tested and saved and when pressing the detection button, it will be classified the disease, and also the time taken for the detection process is shown, in our example it reached (0.062s).

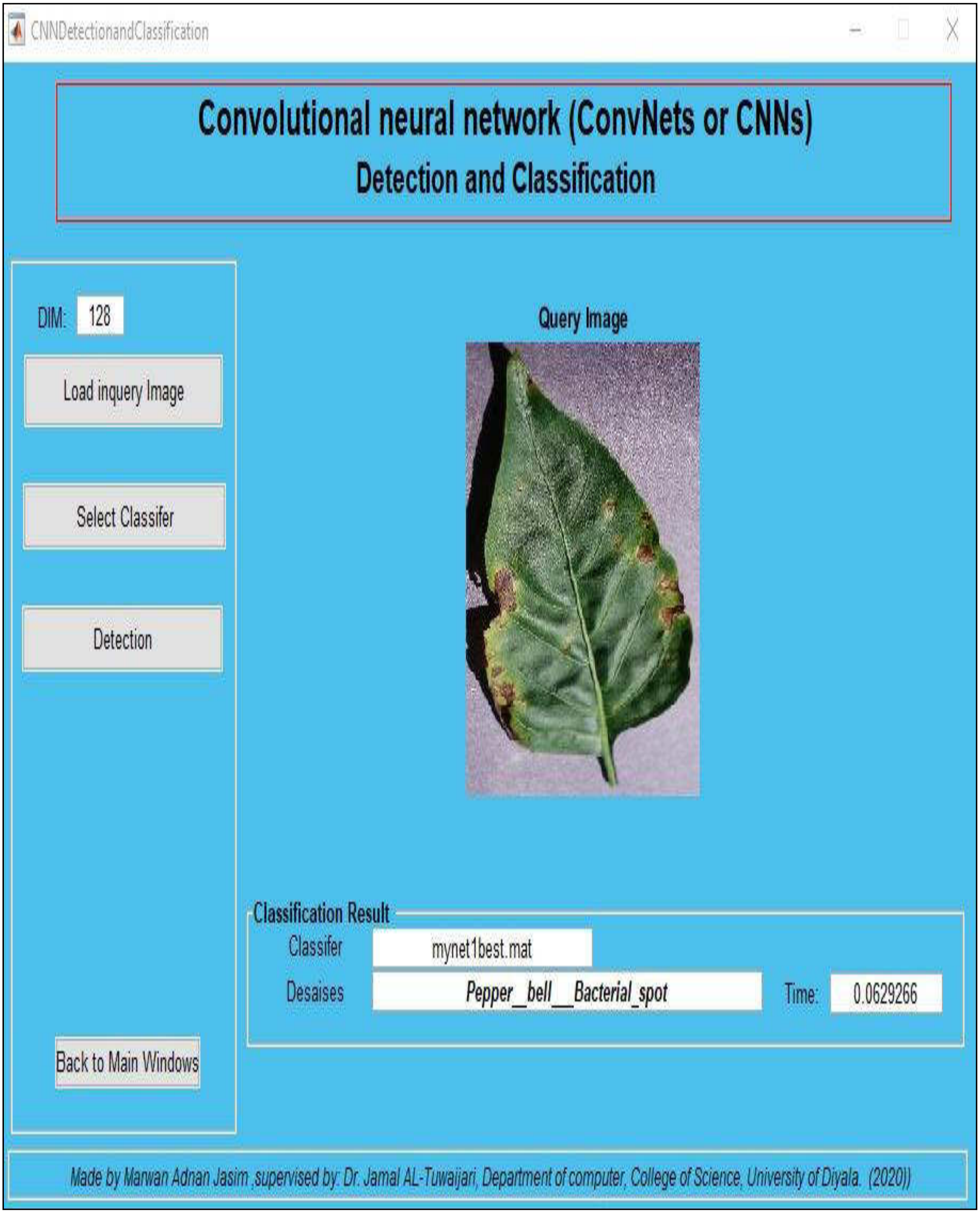


Figure 4.14: The interface of the second proposed approach (Plant leaf diseases detection using a CNN algorithm)

4.6 Comparison, Between the First Proposed Approach with the Second Proposed Approach

The comparison between the two proposed approaches is very important in order to show the strengths and weaknesses of each of them. Initially, when choosing this topic which is the detection of plant leaf diseases, at first, the MSVM algorithm was used in the first proposed approach, but after getting the results of accuracy is not high, then searched for an alternative approach is better than the first proposed approach in order to obtain a high detection accuracy and very few errors, unlike the first proposed approach. After experiments and research, the CNN algorithm was chosen in the second proposed approach for classification the plant leaf diseases. It is worth noting that the same dataset were used, as well as the same data division for training and testing, which is 70% for training and 30% for testing for both algorithms. After showing the results of the two approaches at all stages and found that the best is the second proposed approach for the following reasons:

1- The high accuracy obtained from the second proposed approach in the training and testing stages for each of the 15 classes, as well as the overall accuracy rate in these two stages as the table (4.5) that shows the accuracy of all classes and overall accuracy in the two proposed approaches, plus figure (4.15) that shows the chart of the accuracy comparison for MSVM and CNN algorithms for each class.

2- The time taken for the training process in the first proposed approach using the MSVM algorithm is more than the time taken in the second proposed approach that used the CNN algorithm because the first proposed approach needs a long additional time to extract the features from the images. This is also an advantage of the CNN algorithm which extracts its features internally without the need for that extra time.

no.	Class name(plant diseases)	Accuracy in MSVM training	Accuracy in CNN training	Accuracy in MSVM testing	Accuracy in CNN testing
1	Pepper bell Bacterial spot	67.40	98.60	59.10	97.40
2	Pepper bell healthy	58.00	99.20	47.40	99.30
3	Potato Early blight	83.10	98.70	72.70	99.30
4	Potato healthy	100	98.10	80.00	93.30
5	Potato Late blight	56.20	98.40	54.50	99.00
6	Tomato Target Spot	58.50	94.60	49.30	93.90
7	Tomato_mosaic_virus	100	99.60	66.70	99,10
8	Tomato Yellow Leaf_ Curl Virus	81.40	98.80	75.30	98.40
9	Tomato Bacterial spot	72.00	99.20	67.10	99.40
10	Tomato Early blight	69.20	98.30	56	96.90
11	Tomato healthy	60.90	99.60	88.10	99.60
12	Tomato Late blight	60.80	96.40	59.70	96.80
13	Tomato Leaf Mold	65.20	97.90	53.50	97.20
14	Tomato_Septoria leaf spot	55.00	97.90	50.10	97.00
15	Tomato Spider mites	62.20	99.30	55.80	99.60
16	Rate overall accuracy of the training for MSVM	**69.33**			
17	Rate overall accuracy of the training for CNN	**98.29**			
18	Rate overall accuracy of the testing for SVM	**61.79**			
19	Rate overall accuracy of the testing for CNN	**98.029**			

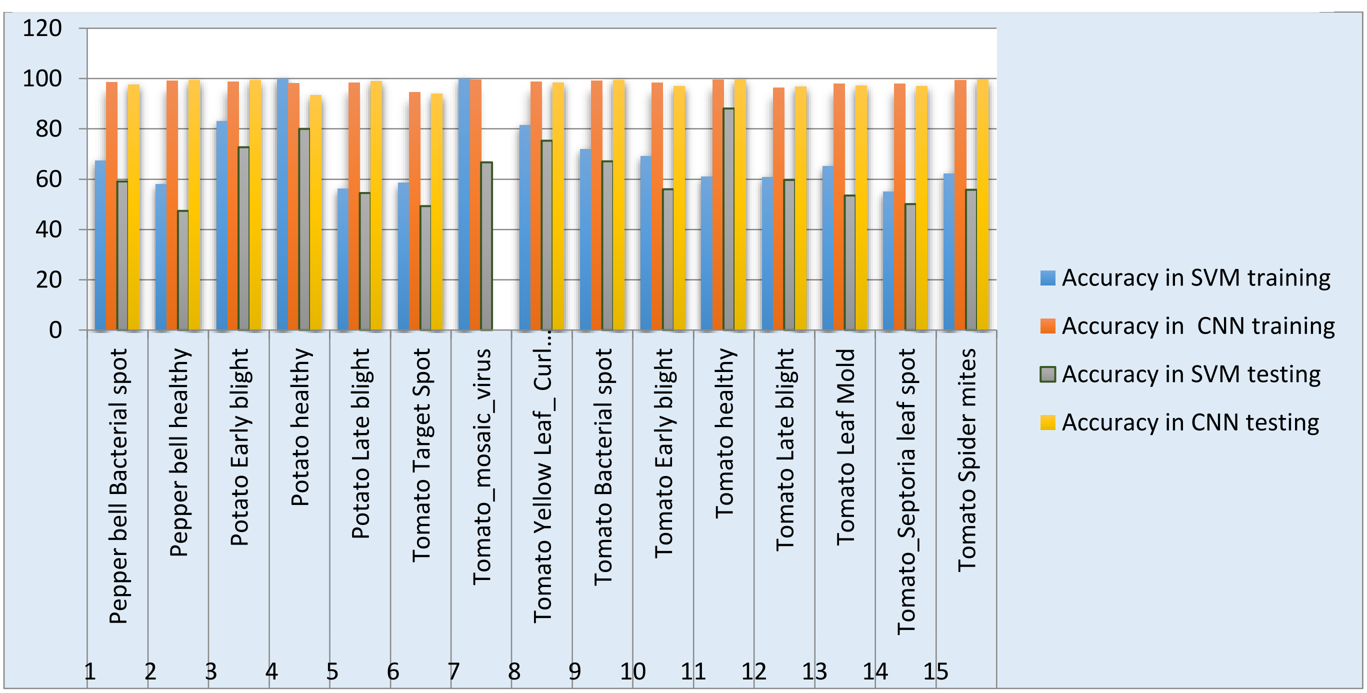


Figure 4.15: The chart shows the accuracy comparison between MSVM and CNN algorithms

4.7 Proposed approach vs. Related Works

After obtained high accuracy results on the large data of plant leaf diseases images, and many classes that classified, a comparison was made between the second proposed approach and the work related to it, which showed the efficiency and high accuracy of the proposed approach. As the table (4.6) showed a comparison of the second proposed approach with some work related to our work. It shows the high accuracy with the largest numbers of data for the proposed approach, it also illustrates the weaknesses of the related works.

Table 4.6: Comparison between the second proposed approach VS related work

Author(s), Year	Ref. No.	Algorithm for (classification)	Dataset size (Images Number)	Accuracy
Kawasaki, et al. 2015	[8]	CNN	800	94.9%
Tm, Prajwala, et al , 2018	[13]	CNN	18160	94_95%
Ozguven, Mehmet Metin, and Kemal Adem2019	[15]	CNN	155	95.48%
Francis, Mercelin, and C. Deisy,2019	[16]	CNN	3663	87%
Sambasivam, G., and Geoffrey Duncan Opiyo, 2020	[17]	CNN	10,000	93%
The second proposed approach	-	**CNN**	**20636**	**98.029%**

Chapter Five

Conclusions and Suggestions for Future Works

Chapter Five

Conclusions and Suggestions for Future Works

5.1 Conclusions

In this chapter, the proposed approach is summarized; the following conclusions were taken from collection of test results. Some of those conclusions are listed in the following:

1-The proposed approach is a robust methodology to detect and classify plant leaf diseases with accurate, fastest results based on computer facilities and its modern technologies.

2-This work is conducted to obtain the results using two different proposed approaches each one used different algorithm for classification and make a comparison between them. The first proposed approach use multi-class support vector machine (MSVM) algorithm and the second proposed approach use a Convolutional Neural Network (CNN) algorithm.

3-The same dataset is used in the two proposed approaches and the data is divided in the same percentage also, which was 30% for the testing and 70% for training; this was the best ratio after trying all percentages as in the table (4.4).

4-The results of the comparison showed the preference of the second proposed approach which is based on deep learning field using CNN algorithm, in terms of performance and accuracy. The obtained accuracy in testing phase for the second proposed approach was (98,029%), while the accuracy for the first proposed approach was (61.79%).

In addition, the comparison explained the main reason for low accuracy in the first proposed approach, which is related to the number of classifiers that

have been classified and also the large size of dataset. This has been proven by applying the MSVM algorithm to classifying only two classes and on a smaller number of the dataset, which obtained a very high accuracy of (100%) in training and testing phases for tomato plant infected and uninfected as shown in figure (4.8). At the conclusion of the comparison, it is proven that the second proposed approach is the best, the fastest, most accurate and powerful approach for detecting and classifying plant leaf diseases.

5.2 Suggestions for Future Works

The proposed approach plant leaf diseases detection is flexible approach and there are many suggestions that can be performed for future work as follows:

1. Design an expert system that can detect plant leaf diseases in automated way depending on multiple techniques and presents ways to prevent these diseases and treatment.
2. Using various machine learning techniques such as (KNN, Native Bayes, etc...) with comparison between them, also other optimizers for the proposed approach to obtain a better accuracy rate.
3. Applying the approach to other various types of plants and other diseases to make the approach more comprehensive.

References

REFERENCES

[1] Pooja, V., Rahul Das, and V. Kanchana. "Identification of plant leaf diseases using image processing techniques." In *2017 IEEE Technological Innovations in ICT for Agriculture and Rural Development (TIAR)*, PP 130-133. IEEE, 2017.

[2] Dr. Akleshwar Mathur, "A Study of Some Plants of Economic Importance and their Values in JIET Campus", 4th International Conference on Multidisciplinary Research & Practice, PP 84-87, 2017.

[3] Arivazhagan S., Newlin Shebia R. "Detection of unhealthy region of plant leaves and classification of plant leaf diseases using texture features", Agricultural Engineering Institute: CIGR journal, Vol.15, No.1, 2013.

[4] Sandesh Raut1, Kartik Ingole2, "Review On Leaf Disease Detection Using Image Processing Techniques", International Research Journal of Engineering and Technology (IRJET) Vol.04, 2017.

[5] Dhingra, Gittaly, Vinay Kumar, and Hem Dutt Joshi, "Study of digital image processing techniques for leaf disease detection", Multimedia Tools and Applications 77, No.15, PP 19951-20000, 2018.

[6] Deepika Choudhary, Anjali Lather and Sandeep Yadav, "Overview of Image Processing", Computer Science and Engineering, Vol.2, PP 115-119, October 2014.

[7] Pavel, Monirul Islam, Syed Mohammad Kamruzzaman, Sadman Sakib Hasan, and Saifur Rahman Sabuj, "An IoT Based Plant Health Monitoring System Implementing Image Processing," In 2019 IEEE 4th International Conference on Computer and Communication Systems (ICCCS), PP 299-303, 2019.

[8] Kawasaki, Yusuke, Hiroyuki Uga, Satoshi Kagiwada, and Hitoshi Iyatomi. "Basic study of automated diagnosis of viral plant diseases using convolutional neural networks", In International Symposium on Visual Computing, PP 638-645. Springer, Cham, 2015.

[9] Mokhtar, Usama, Mona AS Ali, Aboul Ella Hassanien, and Hesham Hefny. "Identifying two of tomatoes leaf viruses using support vector machine."

In *Information Systems Design and Intelligent Applications*, pp. 771-782. Springer, New Delhi, 2015.

[10] Prajakta Mitkal, Priyanka Pawar, Mira Nagane, Priyanka Bhosale, Mira Padwal and Priti Nagane "Leaf Disease Detection and Prevention Using Image processing using Matlab", International Journal of Recent Trends in Engineering & Research (IJRTER), Vol.02, February 2016.

[11] Amara, Jihen, Bassem Bouaziz, and Alsayed Algergawy, "A Deep Learning-based Approach for Banana Leaf Diseases Classification", In: BTW (Workshops), PP 79–88, 2017.

[12] Atabay, Habibollah Agh, "DEEP RESIDUAL LEARNING FOR TOMATO PLANT LEAF DISEASE IDENTIFICATION," Journal of Theoretical & Applied Information Technology 95, No. 24, 2017.

[13] Tm, Prajwala, Alla Pranathi, Kandiraju SaiAshritha, Nagaratna B. Chittaragi, and Shashidhar G. Koolagudi, "Tomato leaf disease detection using convolutional neural networks," In 2018 Eleventh International Conference on Contemporary Computing (IC3), PP 1-5, 2018.

[14] P. Goncharov1, G. Ososkov2, A. Nechaevskiy2, A. Uzhinskiy2, and I.Nestsiarenia, "Disease Detection on the Plant Leaves by Deep Learning", International Conference on Neuroinformatics, No. 18-07-00829, PP 1-9, October 2018.

[15] Ozguven, Mehmet Metin, and Kemal Adem. "Automatic detection of leaf spot disease in sugar beet using deep learning algorithms." *Physica A: Statistical Mechanics and its Applications* 535: 122537, 2019.

[16] Francis, Mercelin, and C. Deisy. "Disease Detection in Agricultural Plants Using Convolutional Neural Networks—A Visual Understanding." In *2019 6th International Conference on Signal Processing and Integrated Networks (SPIN)*, PP 1063-1068, IEEE, 2019.

[17] Sambasivam, G., and Geoffrey Duncan Opiyo. "A predictive machine learning application in agriculture: Cassava disease detection with imbalanced dataset using convolutional neural networks." *Egyptian Informatics Journal* (2020).

[18] Jhuria, Monika, Ashwani Kumar, and Rushikesh Borse. "Image processing for smart farming: Detection of disease and fruit grading." In *2013 IEEE Second International Conference on Image Information Processing (ICIIP-2013)*, PP 521-526, 2013.

[19] Dubey, Sachin and Anand Singh Jalal. "Detection of Apple Fruit Diseases Using Complete Local Binary Patterns", Third International Conference on Computer and Communication Technology, PP 346-351, 2012.

[20] Hughes, David, and Marcel Salathé, "An open access repository of images on plant health to enable the development of mobile disease diagnostics," arXiv preprint arXiv: 1511.08060, 2015.

[21] John P. Damicone & Lynn Brandenberger, "Diseases Caused by Bacteria, Viruses, and Nematodes", Oklahoma Cooperative Extension Fact Sheets are also available on our website at: http://osufacts.okstate.edu .

[22] Leonberger, Kimberly, Kelly Jackson, Robbie Smith, and Nicole Ward Gauthier, "Plant diseases [2016]", 2016.

[23] Verkley, G. J. M., W. Quaedvlieg, H-D. Shin, and P. W. Crous, "A new approach to species delimitation in Septoria," Studies in Mycology 75, PP 213-305, 2013.

[24] Missouri Botanical Garden, "Leaf Spot Diseases of Shade Trees and Ornamentals", November 2014.

[25] S. A. Ganie1*, M. Y. Ghani1, Qazi Nissari, Naymma Jabeen2, Qaisar Anjum3, F. A. Ahanger1 and AAdil Ayaz, "Status and symptomatology of early blight (Alternaria solani) of potato (Solanum tuberosum L.) In Kashmir valley", African Journal of Agricultural Research, Vol. 8, 2013.

[26] Dixon, L.J., Schlub, R.L., Pernezny, K. and Datnoff, L.E., "Host specialization and phylogenetic diversity of Corynespora cassiicola", The American Phytopathological Society, Vol. 99, No. 9, 2009.

[27] Pernezny, Ken, Peter Stoffella, Nikol Havranek, Jairo Sanchez, and Audrey Beany, "Kicking control of target spots up a notch", Acta, Hortic, Vol 695, PP 175-180, 2005.

[28] Okmen, Bilal, and Pierre JGM de Wit. "10 Cladosporium fulvum–Tomato Pathosystem: Fungal Infection Strategy and Plant Responses", Molecular Plant Immunity, PP 211-224, 2013.

[29] Douglas, S.M., "Powdery mildew of tomato", The Connecticut Agricultural Experimental Station, PP 037 (11/03R), 2003.

[30] Morgan, Mike. "Moulds", Microscopy UK, June 2012.

[31] Nyankanga, R., M. Njogu, J. Muthomi, and M. Olanya, "Efficacy of fungicide combinations, phosphoric acid and plant extract from stinging nettle on potato late blight management and tuber yield," *Archives of phytopathology and plant protection* 45, No. 12 , PP 1449-1463, 2012.

[32] Glick, Efrat, Yael Levy, and Yedidya Gafni, "The viral etiology of tomato yellow leaf curl disease–a review", Plant Protection Science 45, No. 3, PP 81-97, 2009.

[33] Brunt, Alan A. "Tomato mosaic virus", in the plant viruses, PP 181-204. Springer, Boston, MA, 1986.

[34] Klompen, Hans, and David Grimaldi. "First Mesozoic record of a parasitiform mite: a larval argasid tick in Cretaceous amber (Acari: Ixodida: Argasidae)," Annals of the Entomological Society of America 94, No.1, PP 10-15, 2001

[35] Fasulo TR, Denmark HA, "Two spotted spider mite", Institute of Food and Agricultural Sciences, No. EENY-150, December 2009.

[36] R. Gonzalez and R. Woods, "Digital image processing" 2nd ed. upper saddle river new jersey: prentice hall.

[37] Priti Badar1, Suchitra.C2, "Disease Detection of Pomegranate Plant using Image processing", International Journal of Computer Science and Mobile Computing, Vol. 5, PP 328-334, May 2016.

[38] R. Gonzalez, R. Woods and S. Eddins, "Digital Image Processing Using MATLAB" 2nd ed., Tata McGraw Hili New Delhi, 2010.

[39] Rathore, Vivek Singh, Messala Sudhir Kumar, and Ashwini Verma. "Colour based image segmentation using L* a* b* colour space based on genetic algorithm." International Journal of Emerging Technology and Advanced Engineering 2, No. 6, PP156-162, 2012.

[40] Dr.S.Kannan,Vairaprakash Gurusamy & Subbu Kannan, "REVIEW ON IMAGE SEGMENTATION TECHNIQUES" Madurai Kamaraj University Conference Paper, October 2014.

[41] Khan, Shehroz S., and Amir Ahmad, "Cluster center initialization algorithm for K-means clustering," Pattern recognition letters 25, No. 11, PP 1293-1302, 2004.

[42] Dhanachandra, Nameirakpam, Khumanthem Manglem, and Yambem Jina Chanu, "Image segmentation using K-means clustering algorithm and subtractive clustering algorithm", *Procedia Computer Science* 54, PP 764-771, 2015.

[43] K. Rohini and G. Suseendran, "Aggregated K Means Clustering and Decision Tree Algorithm for Spirometry Data", Indian Journal of Science and Technology, Vol 9(44), November 2016.

[44] Chandhok, Chinki, Soni Chaturvedi, and A. A. Khurshid, "An approach to image segmentation using K-means clustering algorithm", International Journal of Information Technology (IJIT) 1, No. 1, PP 11-17, 2012.

[45] Betke, Margrit, and Nicholas C. Makris, "Fast object recognition in noisy images using simulated annealing", In Proceedings of IEEE International Conference on Computer Vision, PP 523-530. IEEE, 1995.

[46] Saravanan, C, "Color image to grayscale image conversion", In 2010 Second International Conference on Computer Engineering and Applications, Vol. 2, PP 196-199, IEEE, 2010.

[47] S. T. Rasheed, "Counting and Classification Moving Objects in Digital Video", M.Sc thesis, University of Technology, Department of Computer Sciences, 2017.

[48] Kumar, Gaurav, and Pradeep Kumar Bhatia, "A detailed review of feature extraction in image processing systems", In 2014 Fourth international conference on advanced computing & communication technologies, PP 5-12, IEEE, 2014.

[49] Patil, J.K. and Kumar, R., "Feature extraction of diseased leaf images" Journal of signal and image processing 3, No.1, PP 60-63, 2012.

[50] Vyas, Ankur M., Bjial Talati, and Sapan Naik, “Colour feature extraction techniques of fruits: a survey”, International Journal of Computer Applications 83, No. 15, 2013.

[51] Duanmu, Xiaoyin. "Image retrieval using color moment invariant." In *2010 Seventh International Conference on Information Technology: New Generations*, pp. 200-203. IEEE, 2010.

[52] Pradeep, N., H. Girisha, and K. Karibasappa, “Segmentation and feature extraction of tumors from digital mammograms”, Computer Engineering and Intelligent Systems 3, No. 4, PP 37-46, 2012.

[53] Aggarwal, Namita, and R. K. Agrawal, “First and Second Order Statistics Features for Classification of Magnetic Resonance Brain Images”, Journal of Signal and Information Processing, Vol 3, PP 146-153, 2012.

[54] Er. Kanchan Sharma, Er. Priyanka , Er. Aditi Kalsh, and Er.Kulbeer Saini, “GLCM and its Features” International Journal of Advanced Research in Electronics and Communication Engineering (IJARECE) Vol 4, Issue 8, August 2015.

[55] Azawi, Raghad Majeed, Dhahir Abdulhade Abdulah, Jamal Mustafa Abbas, and Ibrahim Tareq Ibrahim. "Brain Tumors Classification by Using Gray Level Co-occurrence Matrix, Genetic Algorithm and Probabilistic Neural Network." *Diyala Journal of Medicine* 14, No. 2 (2018): 138-151.

[56] Theodoridis and K. Koutroumbas, “Pattern recognition”, 4th ed. Elsevier/Acad. Press, 2008.

[57] Mohanaiah, P., P. Sathyanarayana, and L. GuruKumar, “Image texture feature extraction using GLCM approach”, International journal of scientific and research publications 3, No. 5, PP 1-5, 2013.

[58] Kaur, Sukhvir, Shreelekha Pandey, and Shivani Goel, “Plants disease identification and classification through leaf images: A survey”, Archives of Computational Methods in Engineering 26, No. 2, PP 507-530, 2019.

[59] Richert, Willi, “Building Machine Learning Systems with Python”, Packt Publishing Ltd, No. 1200713, 2013.

[60] Al-Tuwaijari, Jamal Mustafa, and Dalal Adnan Abbas. "Intelligent Human Age Estimation Based Machine Learning Techniques." In 2019 1st AL-

Noor International Conference for Science and Technology (NICST), PP 79-84. IEEE, 2019.

[61] Ayodele, Taiwo Oladipupo, “Types of machine learning algorithms”, new advances in machine learning, PP 19-48, 2010.

[62] Russel, Stuart, and Peter Norvig, “Artificial Intelligence: A Modern Approach”, Third Edition, Prentice Hall, 2010.

[63] Mehryar Mohri, Afshin Rostamizadeh, Ameet Talwalkar, “Foundations of Machine Learning”, the MIT Press ISBN 9780262018258, 2012.

[64] Jamal. M. AL-Tuwaijari and S. I. Mohammed, “Face Image Recognition Based on Linear Discernment Analysis and Cuckoo Search Optimization with SVM”, International Journal of Computer Science and Information Security. ISSN 1947 5500, IJCSIS, Vol. 15, No. 11, November 2017.

[65] Kolluru, Pavan Kumar, “SVM Based Dimensionality Reduction and Classification of Hyperspectral Data”, University of Twente Faculty of Geo-Information and Earth Observation (ITC), 2013.

[66] Lasse Alfredsson, “Feature extraction for image selection using machine learning”, M.s.C Thesis in Electrical Engineering Department of Electrical Engineering, Linkoping University, 2017.

[67] Ghaidaa Waleed., and Jamal Mustafa Al-Tuwaijari. "Satellite Images Scene Classification Based Support Vector Machines and K-Nearest Neighbor." *Diyala Journal For Pure Science* 15, No. 03, PP. 70-87, 2019.

[68] Ramon Huertaa, Fernando Corbachob ,and Charles Elkanc, “Nonlinear support vector machines can systematically identify stocks with high and low future returns”, Algorithmic Finance 2, PP 45-58, 2013

[69] Achirul Nanda, Muhammad, Kudang Boro Seminar, Dodi Nandika, and Akhiruddin Maddu, “A comparison study of kernel functions in the support vector machine and its application for termite detection”, Information 9, No. 1 , P 5, 2018.

[70] Soentpiet, Rosanna, “Advances in kernel methods: support vector learning”, MIT press, 1999.

[71] Vapnik, Vladimir, and Vlamimir Vapnik. “Statistical learning theory”, 1998.

[72] Wang, Zhe, and Xiangyang Xue.,"Multi-class support vector machine.", In Support Vector Machines Applications, Springer, PP 23-48, 2014.
[73] Nuha Jameel Ibrahim, "Intelligent Human Detection and Recognition System, electronic thesis, Computer Sciences, University of Technology, 2014.
[74] I. Arel, D. C. Rose, and T. P. Karnowski, "Deep machine learning-a new frontier in artificial intelligence research [research frontier]", Computational Intelligence Magazine, IEEE, vol. 5, PP 13-18, 2010.
[75] Glauner, Patrick O. "Deep convolutional neural networks for smile recognition", arXiv preprint arXiv: No.1508.06535, 2015.
[76] Benuwa, Ben Bright, Yong Zhao Zhan, Benjamin Ghansah, Dickson Keddy Wornyo, and Frank Banaseka Kataka, "A review of deep machine learning," In International Journal of Engineering Research in Africa, Vol. 24, PP 124-136. Trans Tech Publications Ltd, 2016.
[77] Bengio, Yoshua, Aaron Courville, and Pascal Vincent, "Representation learning: A review and new perspectives", IEEE transactions on pattern analysis and machine intelligence 35, No. 8, PP 1798-1828, 2013.
[78] LeCun, Yann, Léon Bottou, Yoshua Bengio, and Patrick Haffner. "Gradient-based learning applied to document recognition." Proceedings of the IEEE 86, No. 11, PP 2278-2324, 1998.
[79] Guo, Yanming, Yu Liu, Ard Oerlemans, Songyang Lao, Song Wu, and Michael S. Lew, "Deep learning for visual understanding: A review.", Neurocomputing 187, PP 27-48, 2016.
[80] Litjens, Geert, Thijs Kooi, Babak Ehteshami Bejnordi, Arnaud Arindra Adiyoso Setio, Francesco Ciompi, Mohsen Ghafoorian, Jeroen Awm Van Der Laak, Bram Van Ginneken, and Clara I. Sánchez, "A survey on deep learning in medical image analysis", Medical image analysis, PP 60-88, 2017.
[81] Anna Gummeson "Prostate Cancer Classification using Convolutional Neural Networks" M.S.C thesis Lund Institute of Technology, 2016.

[82] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton, “Imagenet classification with deep convolutional neural networks”, In Advances in neural information processing systems, PP 1097-1105, 2012.

[83] Zeiler, Matthew D, “Hierarchical convolutional deep learning in computer vision”, Ph.D Thesis, New York University, 2013.

[84] Yamashita, Rikiya, Mizuho Nishio, Richard Kinh Gian Do, and Kaori Togashi, “Convolutional neural networks: an overview and application in radiology”, Insights into imaging 9, PP 611-629, 2018.

[85] Iftene, M., Q. Liu, and Y. Wang. “Very high resolution images classification by fine tuning deep convolutional neural networks” In Eighth International Conference on Digital Image Processing (ICDIP 2016), International Society for Optics and Photonics, vol.10033, PP 100332D. 2016.

[86] Demuth, Howard B., and Mark H. Beale. , Orlando De Jess, and Martin T Hagan. *“Neural Network Design”, Martin Hagan*, 2014

[87] LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton, “Deep learning”, Nature 521, No. 7553, PP 436-444, 2015.

[88] Ioffe, Sergey, and Christian Szegedy. “Batch normalization: Accelerating deep network training by reducing internal covariate shift.”, *arXiv preprint arXiv:1502.03167,* 2015.

[89] Bottou, Léon, and Olivier Bousquet, “The tradeoffs of large scale learning”, In Advances in neural information processing systems, PP 161-168, 2008.

[90] Kantardzic, Mehmed. *Data mining: concepts, models, methods, and algorithms*. John Wiley & Sons, 2011.

[91] Reem Majeed Ibrahim, “Classification of Face Image Based on Gender Using Intelligent Method”, M.s.C electronic thesis, Computer Sciences, University of Technology, 2015.

Appendix A: A sample of (plant village) dataset used in the proposed approach

1_Pepper _bell _bacterial spot

2_Pepper_bell_healthy

3_Potato___Early_blight

4_Potato___healthy

5_Potato__Late_blight

6_Tomato__Target_Spot

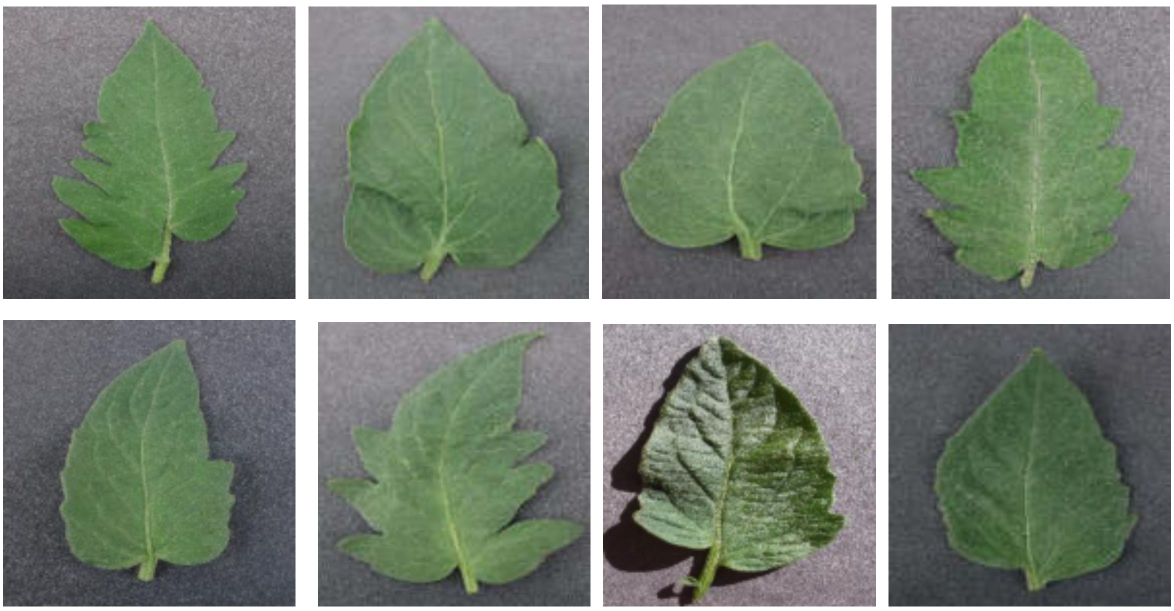
7_Tomato _mosaic virus

8_Tomato_Yellow Leaf_Curl_Virus

9_Tomato_Bacterial_spot

10_Tomato_Early_blight

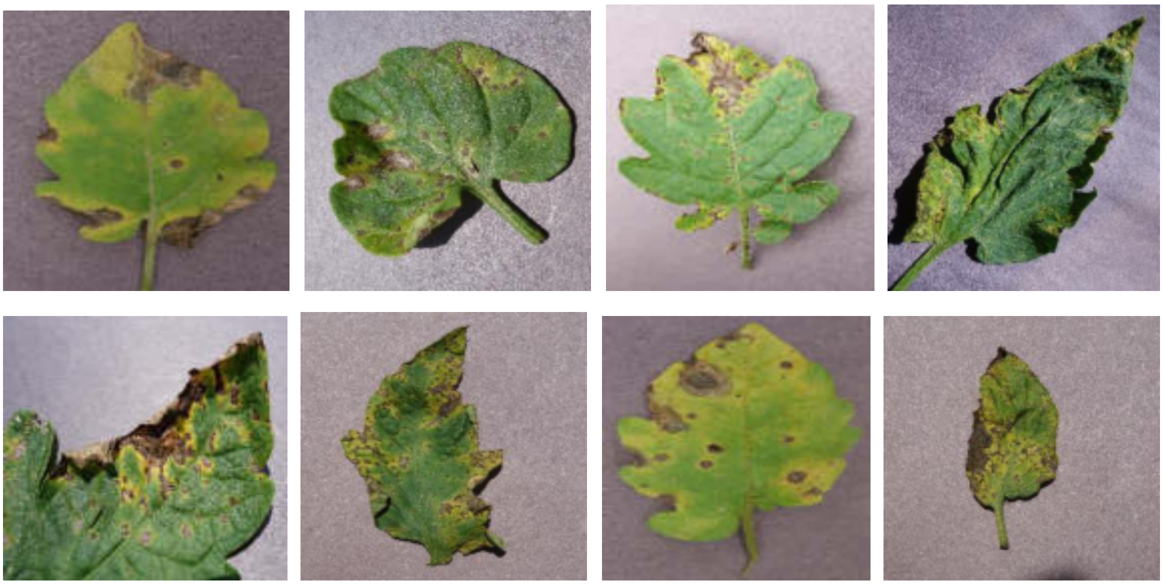
11_Tomato_healthy



12_Tomato_Late_blight

13_Tomato_Leaf_Mold

14_Tomato_Septoria_leaf_spot



15_Tomato_Spider_mites_Two_spotted_spider_mite

الخـلاصـــة

المنتجات الزراعية هي ضرورية جدا ويحتاجها كل بلد. إذا كانت النباتات مصابة بالأمراض ، فإن ذلك يؤثر على الإنتاج الزراعي للبلاد ومواردها الاقتصادية وهذا سبب عملنا في هذا المجال. يقدم هذه المشروع نظامين مقترحين لاكتشاف وتصنيف أمراض أوراق النبات باستخدام معالجة الصور وتقنيات التعلم العميق وإجراء مقارنة بينهما.. تم الحصول على مجموعة البيانات المستخدمة من موقع (Plant Village) ومجموعة البيانات هذه موثوقة ومعتمدة دوليًا . في عملنا اقتصرنا على أنواع معينة من النباتات ، وهي الطماطم والفلفل والبطاطس ، لأنها واحدة من أكثر أنواع النباتات شيوعًا في العالم بشكل عام وفي العراق بشكل خاص. تحتوي مجموعة البيانات على 20636 صورة للنباتات المصابة في هذه الأنواع المذكورة أعلاه والتي تم اعتمادها في نظامنا المقترح. في النظام المقترح الأول ، حيث يتم استخدام عدة خطوات على شكل مراحل ، وهذه المراحل هي مرحلة الحصول على الصورة ، والمعالجة المسبقة للصورة ، وتجزئة الصورة , ومرحلة الصورة ما بعد المعالجة ، واستخراج الميزات ، ومرحلة التصنيف. في مرحلة التقسيم استخدمت خوارزمية (K-mean clustering) ، ولاستخراج الميزات من الصور تم استخدام طريقة ((Gray Level co-occurrence Matrix (GLCM) . تم استخراج أحد عشر ميزة من كل صورة من صور اوراق النباتات، كما تم استخدام خوارزمية Support Vector Machine (SVM) لاجراء عملية التصنيف.

اما في النظام المقترح الثاني ، تم استخدام الشبكة العصبية الالتفافية (CNN) والتي يتم من خلالها تصنيف أمراض أوراق النبات وذلك من خلال هيكلية خاصة لهذه الخوارزمية المتكونة من عدة طبقات. في هذين النظامين المقترحين ، تم تصنيف 15 فئة ، بما في ذلك 12 فئة لأمراض النباتات المختلفة التي تم اكتشافها ، مثل البكتيريا والفطريات والفايروسات وغيرها ، و 3 فئات للأوراق الصحية من الأنواع الثلاثة المستخدمة.

أظهرت النتائج التي تم الحصول عليها من المقارنة بين النظامين المقترحين من حيث الأداء والدقة تفضيل النظام الثاني الذي اعتمد التعلم العميق واستخدام خوارزمية CNN ، على النظام الأول ، لأن معدل الدقة الإجمالي الذي تم الحصول عليه من النظام الثاني المقترح (98،029٪) في مرحلة الاختبار و (98.29٪) في مرحلة التدريب. في حين أن معدل الدقة الإجمالي الذي تم الحصول عليه من النظام المقترح الأول هو (61.79٪) في مرحلة الاختبار و (69.33٪) في مرحلة التدريب. لذا ، فإن النظام المقترح الثاني أكثر دقة وقوة في عملية اكتشاف وتصنيف أمراض أوراق النبات.

جمهورية العراق
وزارة التعليم العالي والبحث العلمي
جامعة ديالى
كليــة العلوم

الكشف عن أمراض أوراق النبات وتصنيفها بناءً على تقنيات التعلم العميق

رســالة

مقدمة الى قسم علوم الحاسوب / كلية العلوم / جامعة ديالى وهي جزء من متطلبات نيل درجة الماجستير في علوم الحاسوب

من قبل

مروان عدنان جاسم

باشراف

أ.م.د جمال مصطفى التويجري

1441هـ العراق **2020م**