ALI AL-MASHHADANI ATILIM UNIVERSITY 2020

SEGMENTATION OF DISEASED APPLE TREE LEAVES USING CONVOLUTIONAL NEURAL NETWORK

THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
ATILIM UNIVERSITY

ALI AL-MASHHADANI

A MASTER OF SCIENCE
THESIS
IN
ELECTRICAL AND ELECTRONICS ENGINEERING

AUGUST 2020

SEGMENTATION OF DISEASED APPLE TREE LEAVES USING CONVOLUTIONAL NEURAL NETWORK

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
ATILIM UNIVERSITY

BY

ALI AL-MASHHADANI

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
ELECTRICAL AND ELECTRONICS ENGINEERING

AUGUST 2020

Approval of the Graduate School of Natural and Applied Sciences, Atilim University.

_____

Prof. Dr. Ali Kara
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of **Master of Science in Electrical and Electronics Engineering, Atilim University**.

_____

Assoc. Prof. Dr. Efe Eseller
Head of Department

This is to certify that we have read the thesis "**Segmentation of Diseased Apple Tree Leaves Using Convolutional Neural Network**" submitted by "Ali Al-Mashhadani" and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

_____

Asst. Prof. Dr. Hakan TORA
Supervisor

Asst. Prof. Dr. Emre Sümer
Computer Eng. Dept., Başkent University _____

Asst. Prof. Dr. Hakan TORA
Electrical & Electronics Eng. Dep., Atilim University _____

Asst. Prof. Dr. Erhan GÖKÇAY
Software Eng. Department, Atilim University _____

Date: *9.8.2020*

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name : Ali Al-Mashhadani

Signature :

ABSTRACT

SEGMENTATION OF DISEASED APPLE TREE LEAVES USING CONVOLUTIONAL NEURAL NETWORK

Al-Mashhadani, Ali Hussein
MSc., Electrical and Electronics Engineering
Asst. Prof. Dr. Hakan TORA

August 2020, 54 pages

In the field of agriculture, the expert's eye might not be able to identify or correctly diagnose the disease at an early stage. The misdiagnosis of plant disease often leads to choosing the wrong treatment and this leads to losing the crops. Therefore, an automatic segmentation system of the diseased leaf is highly required to solve this issue. This thesis displays the prowess of deep learning in the segmentation of Plant Pathology 2020 - FGVC7 dataset that includes high-resolution coloured images of multiple apple foliar disease symptoms such as apple scab, cedar apple rust, and healthy leaves. The proposed segmentation algorithm is the semantic segmentation approach by using two different architectures U-Net and ResNet. The results of both networks have been evaluated by using Pixel Accuracy, IoU, F1-Score, and Recall metrics, and the comparison showed the efficiency of ResNet for this purpose.

Keywords: Apple Foliar Diseases, Deep Learning, Features Extraction, Semantic Segmentation, MATLAB.

ÖZ

KONVOLUTIONAL NÖRAL AĞ KULLANARAK HASTA ELMA AĞAĞI YAPRAKLARININ SEGMENTASYON

Al-Mashhadani, Ali Hussein
Yüksek lisans., Elektrik ve Elektronik Mühendisliği
Doç. Prof. Dr. Hakan Tora

Ağustos 2020, 54 sayfa

Tarım alanında, uzmanın gözü hastalığı erken bir aşamada tanımlayamayabilir veya doğru bir şekilde teşhis edemeyebilir. Bitki hastalığının yanlış teşhisi genellikle yanlış tedavinin seçilmesine ve bu da mahsulün kaybına neden olur. Bu nedenle, hastalıklı yaprağın otomatik segmentasyon sistemi bu sorunu çözmek için son derece gereklidir. Bu tez Bitki Patolojisi 2020 segmentasyonunda derin öğrenme nin cesaretini görüntüler - FGVC7 veri seti elma kabuğu gibi birden fazla elma foliar hastalığı belirtileri yüksek çözünürlüklü renkli görüntüler içeren, sedir elma pas, ve sağlıklı yapraklar. Önerilen segmentasyon algoritması, U-Net ve ResNet olmak üzere iki farklı mimari kullanılarak yapılan anlamsal segmentasyon yaklaşımıdır. Her iki ağın sonuçları Pixel Accuracy, IoU, F1-Score ve Recall ölçümleri kullanılarak değerlendirilmiş ve karşılaştırma ResNet'in bu amaca yönelik verimliliğini göstermiştir.

Anahtar Kelimeler: Elma Yaprak Hastalıkları, Derin Öğrenme, Özellik Çıkarma, Anlamsal Segmentasyon, MATLAB.

To My Parents

My Father "Hussein Abdul-Ameer"

My Mother "Suha Mohsin Ali"

ACKNOWLEDGMENTS

Before I thank anyone, I must mention God's blessing on me. He inspired me the patience and strength to complete this thesis in a satisfactory manner, otherwise, it would not have been possible for me to accomplish it.

I would like to express my sincere appreciation to my respected supervisor, Asst. Prof. Dr. Hakan Tora for his instructions, support, and help, and what he offered to me during the research work until I finish this thesis. His door was always opened to me when I need him even when the road got tough. I am without any doubt extremely proud of being one of his students.

Words can't explain how thankful I am to Dr. Hasan Fleyeh, for his motivation and invaluable suggestions throughout the research work, without his timely hints this work would not have been possible. May Allah reward him with the best reward for this.

To my strength, my patience and my companion in this life, Hiba. Thanks and gratitude, are not enough for your support during this research work. May Allah keep you to me.

Also, I am indebted to my dear colleagues the engineers Yezi Al-Obeidi, Sayef Salam, Mohammed Hamid, Othman Al-Hayali, Mohammed Naser, Ahmed Hefdhi, Amir and Ongun for their support, and help throughout my academic journey. May Allah reward them with the best.

Finally, I wish to acknowledge the support and great love of my family, my father, Hussein; my mother, Suha; and my sister, Wurood. They kept me going on throughout my years of study and through the process of researching and writing this thesis. This achievement would not have been possible without their unfailing support and continuous encouragement. Thank you.

Author

Ali Al-Mashhadani

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF SYMBOLS/ABBREVIATIONS

ADAM	-	Adaptive Moment Estimation
CNN	-	Convolutional Neural Network
DL	-	Deep Learning
DNN	-	Deep Neural Network
FN	-	False Negative
FP	-	False Positive
ML	-	Machine Learning
SGD	-	Stochastic Gradient Descent
SVM	-	Support Vector Machine
TN	-	True Negative
TNR	-	True Negative Rate
TP	-	True Positive
TPR	-	True Positive Rate

CHAPTER 1

INTRODUCTION

In the last few years there has been a growing interest in developing a modern apple industry, the traditional apple industry faces the problems of ensuring the production of apples, improving the quality and quantity of apples, changing the climate and the environment. The environment and pollution, which cannot meet the needs of sustainable agricultural development. Due to advances in sensor technology and the availability of low-cost integrated circuits, the wireless surveillance sensor network is considered a new generation of surveillance technologies. It is therefore important to study the technology of the Internet of Things in the apple orchard. Wireless sensor technology offers a cost-effective approach to continuously monitoring the infrastructure [1].

The recent researches showed the skilfulness of deep learning in several fields like classification [2,3], bounding box detection [4-6], natural language processing [7-9], machine translation [10-12], speech recognition [13,14], image generation [15,16], and semantic segmentation [17,18]. All these fantastic results make it possible to build up an automated diseased plant detection system based deep learning that detects the disease at an early stage, which could help the farmer from saving the crops.

This task requires to detect the disease in the first step. Since the leaves of the plant occupy the largest part, as well as, the most obvious, in addition to the fact that most symptoms of diseases appear on the leaf in the form of small spots. Therefore, the segmentation of these spots correctly could help in the building of the detection system.

1.1 Apple Foliar Diseasess

Apple foliar diseases can cause serious loss performance and quality. These diseases are often absent. Shortly before, during, or after harvest. The fruit is saved. No cure for treating infected fruits can cause many diseases. Prevented by cultural practice

(optional) Fungicide. But an accurate diagnosis is important to identify and adopts the best management method of prevention of future losses [19]. Several apple diseases are presented and explained below:

- **Fire Blight**

Fire epidemics are difficult to control in many parts of the United States. False mold trees have water stains, brown flowers, and brown leaves. The branches and branches of trees can turn brown or black and cause a dark brown, dark brown liquid. You may look like the hump of a shepherd where a branch falls from the top. The disease hibernates on infected trees and spreads in the spring from rain and insects.

Plant varieties such as Jon free, Liberty, Pristine and Williams Pride avoid the same inevitable varieties Beacon, Granny Smith, Jonathan, Gala and Fuji. Prior to growing, do not modify the tree in spring and use excess fertilizer to promote the most vulnerable, rapid, prolific growth of infectious diseases.

Eliminate the infection when winter is over while the trees are resting. Actively growing trees can spread the disease. Put as much as possible and get rid of winter germs. Do not put it in your property. According to Ohio State University expansion, aerosols are not recommended in modern home gardens [20].

- **Cork spot**

Cork spots are physiological disorders that can affect the visual appeal, which affects the quality of apples. Cork spots generally appear as small pits and green bumps outside of the fatal dimensions. This disorder is likely to develop in June and continue until the first stages of growth and expansion. The green spot eventually increases to 1/4 to 1/2 inches from the discoloured area of the cork apple meat. Cork spots can appear anywhere in the meat.

Fruit growers can confuse this disorder with damage from insects or pathological diseases, especially fungal infections or hail diseases. The apples affected by this disease are edible, but their unattractive appearance affects the attractiveness of the fruit and gives consumers cause for concern. Cork compression is common at York Imperial, but sometimes you can also see yummy and gold yummy. Low soil pH, easy harvests and abnormal growth of shoots are associated with an increased frequency of cork staining.

For long-term management of cork contamination, it is recommended to add agricultural limestone to the soil at the time of planting, as recommended in soil analysis. Depending on the soil analysis, limestone should be added every 3 to 5 years. Spraying calcium not only limits the soil, but also reduces the appearance of cork stains in established pear trees. However, in some situations it may not work. It is important to obtain local recommendations, as recommendations regarding frequency of use may vary with the amount of calcium used, depending on your geographic location. Do not use calcium chloride spray if the temperature exceeds 85 degrees Fahrenheit. The leaves and fruits can be damaged. Because calcium chloride is very corrosive to metals, wash the spray after use [21].

- **Powdery Mildew**

Powdery mildew is a fungal disease that affects various plants. There are several types of mold, each of which affects other plants, roses and legumes (beans, peas).

When a mushroom or plant begins to penetrate, a mold layer forms, which consists of many spores on the leaves. Spores are replaced by other plants by the wind. reduce yield and quality of fruits.

In general, the mildew spores are blown by the wind in the garden, but if the mildew was in the past, dormant spores can come from old or nearby herbaceous plant material.

Unlike many other fungal diseases, the fungus grows in hot, dry climates (15 to 27 ° C), but its distribution requires a relatively high relative humidity (i.e., humidity) around the plant.) In cold, rainy It does not diffuse regions and slows down even at temperatures above 32 ° C (90 ° F). It also affects shade plants and not direct sunlight [22].

- **Rust**

Apples and flowering crab-apples are susceptible to several rust diseases including cedar-apple rust, quince rust, and hawthorn rust. Although incited by different species of fungi in the genus Gymnosporangium, they have in common the fact that they must spend part of their life cycle on various trees and shrubs of the Junipers species such as Eastern red cedar and common juniper. Any of the rust diseases can result in serious losses if environmental conditions are favourable for disease development. Since they

are similar in appearance and life cycles, only cedar-apple rust will be discussed in detail [23].

Symptoms of cedar and apple oxide are found in apple and juniper species. Apples are made on the leaves and fruits of rust and are less common on the skin. Small yellow spots usually appear on the upper surface of the leaves within 1-2 weeks after flowering. The point increases to 1/4 inch or more, and an orange drop appears inside. Finally, the mold here produces a small black coolant. In summer, another type of fruit structure forms on the underside of the leaf and contains mold spores. The structure of this fruit is shown in a series of yellow tubes called a helmet. Some infected leaves may fall from the tree, but only a few falls completely. See Figure 1.1 and Figure 1.2, which represented the disease symptoms.

Figure 1. 1 Apple rust example 1

Figure 1. 2 Apple rust example 2

- **Apple Scab**

Apple scab is caused by the Ben Iine Charis mushroom. Missouri-sensitive apples and crabs are clearly visible to apples and the yellow leaves fall off prematurely in midsummer. Cold, wet spring weather Infections can cause a multitude of dead leaves for many years. Despite the heavy leaf fall, apple peels rarely kill trees. If the heavy leaves last for several years, the tree becomes weak and susceptible to other problems. The skin can cause cosmetic stains on the fruit. Apple fruits can still be eaten. Apple peels can also be infected with sea buckthorn and ears [24].

The first signs of scab from an apple are often not detected. This sign includes a curved blade with black circles and pointed tips at the bottom. The tips on top are velvet, olive green and sooty. Points can be covered by combining the entire sheet. Leaves turn yellow as the disease progresses. The fruit is brown, creates sunken and shabby spots. Apple peels are sometimes mistaken for cedar, apple oxide and mold. The cedar apple is pale yellow in the dark centre of the black fruiting body, which can cause a loose yellowish colour, see Figure 1.3.

Figure 1. 3 Apple scab example

The infection with apple scab begins in early spring with young and young leaves. The initial lesions appear after 10 days in the form of lighter green areas than the surrounding leaf tissue. The damage increases and becomes olive and velvety through the formation of asexual spores (conidia) (Figure 1.4 a, b). Damage to the crust that forms on young leaves is likely to go beyond 1 cm in diameter. However, resistance to the development of old leaves is generally poor or no symptoms are visible. The affected tissue can possibly deform wrinkles and leaf lesions are broken. Heavily infected leaves fall from the tree. Two or three successive stands can weaken the tree and cause increased sensitivity to other stresses such as frost damage, insects and other diseases [25].

Damage to the fetus, as a rule, is "battered" with blisters and clearly defined edges (Figure 1.4 c, d). The first symptoms that can be recognized in the fetus develop rapidly with velvety lesions that change from noggar to olive brown in the waters. Young fruit infections cause sprain and healthy tissue continues to grow (Figure 1.4 d). Heavily infected fruits at the beginning of the tree often decrease.

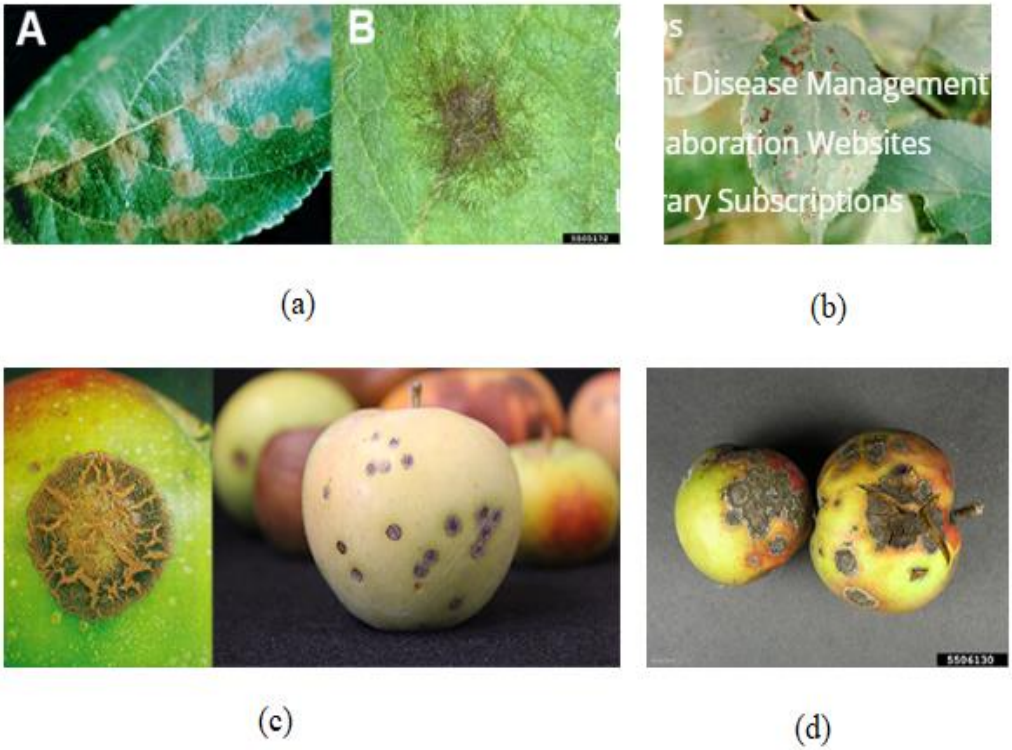

Figure 1. 4 Various apple tree leaf diseases examples

1.2 Research Contributions

The main achievements, including contributions to the field, can be summarized as follows:

- In particular, no study, to our knowledge, has applied the semantic segmentation task on apple foliar diseases dataset. However, this is the first time that the U-Net and ResNet have applied to the problem of extracting high quality features disease spots by using complex images including, different light conditions, various capturing angles, infected leaf with multiple diseases, and unbalanced data issue.
- The benefit of using the proposed networks is expected to determine disease locations, as well as extract disease area from the leaf that will contribute effectively in designing a fully automatic disease detection system based on a real-life plant leaf image.

1.3 Research Questions

- What are the diseases of apple tree leaves? this is the first question in our study which we try to investigate those diseases and decide it's really important to developed computer-based system to tackle it. This problem effected several fields in our live if it's not solved and reduced in the earliest stages such as economics, agriculture, commercial, and environmental.
- What is semantic segmentation and how we can deal with the diseased apple tree leaves as segmentation problem? The segmentation of the image in this disease is very important issue which automatically we can determine the disease period, in fast form without complex procedure and investigations.
- What is machine learning and deep learning? The main question here which technique are more suitable and presented high accuracy with this problem. Which deep learning techniques can present satisfactory results with this problem?
- What is convolution neural network and residual network? This topic explained in detail form and number of network architectures are presented. In other words, which structure is better U-Net as a convolution neural network, or ResNet as a residual neural network in semantic segmentation task. How can those networks deal with the high variations in each disease symptoms? which structure can present best performance in extraction of high dimensionality features of disease spots?
- The last question concerns the possibility of employing one of the proposed networks as a subnet to segment disease region accurately, and the extent to which it can be used to build a plant leaf disease detection model by combining with another subnet that will train in the end-to-end manner based on the extracted disease features to obtain the distinguishing of diseases categories.

1.4 Orginaztion of Thesis

Chapter 1

In chapter one, we present the introduction of the thesis including, an overview of the apple tree diseases that will be as data for our model.

Chpater 2

In chapter two, we discuse several litretures related to plant diseases detection according to their characteristics. As well as, we explain the theoretical background that contributed in developing of our model.

Chapter 3

In this chapter, we demonstrate extensively the structures used to segment the diseases of the apple tree leaves (CNN and ResNet), data augmentation process, training and testing phases of the proposed networks, and evaluation metrics used.

Chapter 4

In chapter four we present and discuss the results that obtianed from each structure, after validated by using 14 test sample each class. Furthermore, we compare our results with other segmentation methods including, k-means, region growing, texture filter, and Otsu method. In addition, we compare the performance of our proposed networks with previous studies related to plant leaf disease detection.

Chapter 5

In chapter five we illustrate the conclusion, limitation of the study, future works and future research related to the thesis.

CHAPER 2

RELATED WORK AND THEORETICAL BACKGROUND

2.1 Related Work

In the last few years, image processing techniques, traditional machine learning algorithms and deep learning architectures have shown the considerable possibility to speed up plant leaf diseases diagnosis with a low error rate. In general, the design of any plant leaf disease detection system includes two main steps. The first step is a features extraction or spots segmentation after some image enhancement methods. The other one is the classification for the extracted features. Therefore, in this section, several techniques related to plant leaf diseases identification presented and analysed.

In Lin K et al. [31], explained an effective segmentation model for Powdery Mildew on cucumber leaf images by using semantic segmentation-based convolution neural network. Firstly, the cucumber leaf images were captured using a USB camera with a resolution of 2592 × 1944 × 3. Furthermore, the disease region was annotated manually, where the white area was annotated as white and the rest as black. In addition, several image processing methods were applied to the annotated images in order to enhance the background, where an image transformed to HSV color space, extract S channel using Otsu thresholding method, as well as, the original RGB image was multiplied by a mask to obtain a black background image. In the last step, the image was resized to 512 × 512 × 3. Later on, U-Net structure was trained on a binary labelled image with the original image to segment the disease region at the pixel level. Moreover, the evaluation of 20 test sample results showed that the proposed method obtained an average of IoU, Dice and Pixel accuracy, 72.11%, 83.45% and 96.08%, respectively. The authors concluded that the proposed method outperforms other existing segmentation methods such as K-means, Random forest, and GBDT.

In L. Gao and X. Lin [32], introduced a fully automatic segmentation algorithm for medicinal plant leaf images. Initially, the Otsu thresholding method was applied to segment the foreground and complex background of the image after solving the RGB

images into a gradient magnitude image and gradient angle image. Later on, the marker-controlled watershed method was executed to obtain the binary segmentation result. Moreover, the proposed algorithm was tested under various database including the self-built one, and it showed a maximum segmentation precision rate when compared with another fully automatic image segmentation methods including deep learning FCN. According to the authors, the proposed algorithm running fast, but slightly slower than the Otsu thresholding method.

In Camargo et al. [33], introduced a method of image processing to identify the visual symptoms of plant diseases from the exploration of RGB images. The colored diseased leaf images transformed into H, I3a and I3b transformation. The I3a and I3b diversions are improved from an adjustment of the original I1 I2 I3 color transformations to meet the demands. The segmentation process of transformed images obtained by analyzing the scattering of the intensities of a histogram. The accuracy of segmentation checked after post-processing by implemented the manually comparison between both segmented leaf images and automatically segmented one. The authors concluded that this technique is suitable to identify a diseased segment even when the segment is represented by an extensive range of various intensities.

In Haiguag Wang et al. [34], presented a technique for the identification of diseases in grape and wheat leaves. In the first stage, The captured RGB images were converted into XYZ color space then transformed into L*a*b color spaces. In order to evaluate the color difference in a*b, a two-dimensional data space, Squared Euclidean distance (D) is used as a similarity distance. The distinguishing of various diseases process is obtained by features extraction of total 4 shapes, 21 color and 25 texture. Moreover, the principal component analysis (PCA) is implemented for reducing the dimensions in the featured data processing. For classification and identification of wheat and grape leaf diseases, different neural network techniques with different parameters including Back Propagation (BP) networks, Generalized Regression Networks (GRNNs) and Probabilistic Neural Networks (PNNs), Radial Basis Function (RBF) were used. The author assured that this method provides accurate and fast recognition of plant diseases.

In Qinghai He et al. [45], proposed a method to identify the diseased leaf of cotton. The extracting process of the disease is implemented using three color models.

Initially, the impact of noise on the captured images is removed by applying spatial non-linear and frequency domain filtering. Subsequently, the histogram equalization process is applied to highlight the edges of the background image and then the contrast of the image is increased. Afterwards, the improved images are transformed into the RGB, HIS, and YCbCr color model. Moreover, the percentage of damage (γ) is selected as a feature to evaluate the amount of damage due to leaf diseases or pests. The experimental results showed that YCbCr color format measured as the best color model for identifying the injured color leaf image when compared. In case of outdoor conditions, the author confirmed that the presented algorithm fails to deal with the random noise interference and leaf shadow.

In Auzi Asfarian et al. [46], suggested a method of paddy diseases identification. The proposed method included the fractal descriptors based on Fourier transforms with texture analysis. Firstly, the RGB injured lesion images were collected manually, then converted to HSV color space. After extracting the saturation components, the lighting effects were reduced by using Histogram equalization. Decisively, for classification purpose, the fractal descriptors were extracted from each lesion and given to PNN classifier. Furthermore, 5-fold cross-validation used to split the training and testing data. The author concluded that the proposed method could be combined with other methods to be used as possible features, especially when two diseases relatively involved have the same color.

In Pradnya Ravindra Narvekar et al. [47], suggested SGDM (spatial gray level dependence matrices) method used in the detection of grape leaves diseases. They used RGB images consisted of four categories under different disease conditions (i.e. Black rot, downy mildew, powdery mildew and normal). The colored images converted into HSI color space. According to the analysis in this research, the S and I components were throw down since they do not provide any useful information, while the H component is considered for further analysis. Furthermore, to generate the mask of green pixels, varying threshold-based Otsu method is used. Afterwards, the color and texture method were used to extract features and classified in different disease class. The authors concluded that the proposed method marked valuable results. Moreover, it can be developed with the use of hybrid algorithms to increase the recognition rate of classification.

In Juan F. Molina et al. [48], discussed the effective way of identifying an early blight disease on tomato leaf. In the first stage, several color descriptors were used to help in the categorization of tomato leaves (i.e. color structure descriptors using Hue-Max-Min method, scalable color descriptor and layout descriptors using YCbCr color space transform). Moreover, in the classification phase, a novel strategy based on nested leaf one out cross-validation method is used to achieve better accuracy. The individual descriptor configuration evaluation performed using an inner loop permit, while the outer loop measures performance assessment between different descriptors. The author concluded that color characterization based on color structure descriptor provided better precision than other methods.

In H. Yu et al. [57], developed a novel leaf spot attention network for apple leaf disease identification. The primary idea of the proposed network based on two subnetworks, where the first one works to transform the input image into a segmented feature map that includes, background, leaf area, and spot area. The seconed subnetwork used to classify leaf diseases depending on the extracted features by the first one. The authors concluded that the proposed method provided a high dimennsonality features of disease spots which leads to increase the accuracy rate of diseases classification process.

In D.G. Sena Jr. et al. [58], introduced a method for identifying maize plant leaf diseases at simplified lighting conditions. The colored maize images that used were captured in eight various locations and in three different light conditions. The proposed work consisted of two stages. Firstly, preprocessing and image analysis, were the original RGB images transformed to 256 grey level images by applying the additional green index, moreover, binary images were created by using the rescaling of the pixel values. Afterwards, the iterative method was applied, in order to, threshold the monochrome images. In the second stage, the classification of healthy and damaged images was implemented by dividing each image into many blocks that carried the desired features to be classified. The author concluded that the proposed algorithm achieved remarkable results with good precision.

In Shen Weizheng et al. [59], have studied the issues that existed in the image segmentation approach for analyzing the leaf spot disease. Initially, the colored RGB images were converted into HSI color space, H element is selected for segmentation

of the leaf spot to reduce the disruption of lightning changes. Later on, the Otsu thresholding method used to segment leaf regions. Moreover, the edges of the leaf disease spots observed by applying the Sobel operator. Finally, the estimating of the quotient of the diseased region and leaf areas were used for classification purpose. The author concluded that this method is quite fast and accurate to classify plant leaf spot diseases.

In Zulkifli Bin Husin et al. [60], proposed the effective method used for the detection of diseased chili leaf. Initially, the proposed method started with preprocessing the captured images to enhance the image information. Furthermore, the color information method is applied to extract the desired features and for distinguishing between the healthy and non-healthy leaf pixels in the image. Finally, in order to measure the healthiness of leaves, the histogram graphs are used to evaluate the results. The author justified that the proposed method works fast and efficient in recognition of plant chili disease.

In D. Zhihua et al. [61], introduced a method to detect cotton mite based on color features and thresholding methods. The proposed method divided into three stages. In the initial stage, diseases spots and stems extraction of the green plants. In the second stage, extract the features in the gray histogram, then thresholding the segmented image to convert it into 8-bit grayscale image. The histogram played an important role in distinguishing the diseased and non-diseased pixels, where the grayscale spots with values 255 elected as diseased pixels, on the other hand, pixels with values less than 120 depicted as non-diseased. In the third stage, area thresholding is applied to segment the binary images. The author concluded that the proposed method was not meet the requirement of the comparison between the areas with spots on the stems because the similarity in color between the mite and stem disease spotted portions.

In Eric Hillman et al. [62], defined approach of injured coffee leaf detection and severity evaluation. The images of coffee leaf were enhanced by applying Gaussian kernel and converted into CIELAB color system. As well as, average pixel intensity was applied to increase the contrast. Afterwards, the Canny edge detector was used to detect the boundary features of the threshold image. Furthermore, in order to extract leaf area only, background removal method based on luminance and color is applied, the background of the image is processed with a YUV color system to maximize leaf

injury detection. Finally, the severity of damaged leaf is evaluated with respect to the percentage of the distribution of pixels for a damaged and healthy leaf. The author concluded that the proposed method can deal with all images, furthermore, it is very fast and helpful in avoiding of the defoliation.

In Kai et al. [63], defined an approach based on a neural network to recognize maize leaves diseases. Initially, the colored RGB images were transformed into YCbCr color space. The author focused on Cb and Cr components because they are less influenced by the brightness. Afterwards, the SGLCM matrix was used to assess texture features. Moreover, the high dimensionality features were used as an input to the 3 hidden layers backpropagation neural networks with a sigmoid activation function to classify different diseases. However, the experimental results showed that the proposed method obtained high precision. The author justified that the neural network has high reliability because the error value is very low, as well as, high performance and the weight design is moderately correct.

In D.S. Guru et al. [64], explained a novel approach for tobacco leaf lesion regions extraction based on artificial neural networks. Firstly, lesion areas were segmented by using morphological operations and contrast expansion transformation with an adjustable parameter. Moreover, in order to identify and analyze the diseased category, R channel was extracted from the lesion area. Furthermore, to classify tobacco seedling leaf diseases as anthracnose or frog-eye spots, Probabilistic Neural Network (PNN) was used. According to the author, classification accuracy could be enhanced by using a combination of various color and texture features. Furthermore, the first-order statistical texture features perform better than Gray level co-occurrence matrix features.

In Wanrat Abdullakasim et al. [65], presented a method based on image analysis to detect brown spot leaf disease. In order to recognize different regions of the leaf, various color descriptors such as RGB and HIS descriptors were used. Furthermore, the distinguishing process infected region and a healthy region was implemented by employing the artificial neural network. As well as, Brier score was utilized to assess the detection ability of the ANN with more hidden layers. Consequently, the proposed algorithm correctly identified 89.92% healthy leaves and 79.23% diseased leaves, however, the proposed method includes misclassification of infected leaves. The

author concluded that this method could be developed by combining the appropriate segmentation technique, the lighting conditions and effects of infection stages.

In Tushar HAI Jaware et al. [66], suggested an effective system for detection and classification of five kinds of leaf diseases. Initially, the original colored images were transformed from the RGB color system to HIS color format and diseased region were segmented by using the K-means clustering technique. Moreover, only the green pixels were masked. As well as, the red, green and blue zero-pixel values and boundary pixels of diseased regions were totally removed. Afterwards, in the classification section, the extracted features that obtained from transformed back the diseased clusters from HIS to RGB and SGDM matrices were generated for each pixel map of the image were used to train a pre-trained neural network. According to the author, the proposed method results showed high ability to classify different disease classes.

In Revathi et al. [67], elucidated a novel method based on improved PSO feature selection technique of cotton leaf diseases detection. In the initial stage, color histogram and color descriptor were used to evaluate color variance feature. Moreover, to extract shape Skew divergence feature, the Sobel and Canny edge detection method was used. As well as, texture feature was evaluated by texture descriptor and Gabor filter. Finally, cotton leaf spot diseases such as micronutrient, verticillium wilt, leaf blight, root rot, fusarium wilt and bacterial blight were classified by using the cross-information gains depth, forward neural network (CIGDFNN) which accurately decreased the error rate.

In Weidan Zhang et al. [68], introduced an identification system for jujube leaf diseases based on image processing approach and neural networks. In the first stage, features were extracted by applying five morphological, eleven texture and nine colors. Furthermore, the qualified features were selected by using Principal Component Analysis (PCA) and step wise discriminant analysis (STEPDISC). Finally, the detection of jujube diseases was executed by using a two-layer tan sigmoid activation function model using 12 parameters. The authors justified that the classification accuracy decreases, due to the high similarity structure index between jujube diseases.

In Pranjali Vinayak Keskar et al. [69], elucidated a method of recognition and classification of leaf disease categories. The proposed method based on fuzzy feature algorithm as a feature extractor, and artificial neural networks as a classifier.

According to the author, the proposed model showed high precision in distinguishing between healthy and diseased leaves.

In Dhiman Mondal et al. [70], presented a method to detect okra leaf which diseased with a yellow vain mosaic virus based on a combination of K-means and Naive Bayes classifier. In the first stage, the K-means algorithm was used after extract the features using gray level co-occurrence. In the distinguishing stage, Naive Bayes classifier was used. Moreover, the experimental results showed that the proposed method correctly identified 87% as diseases. The author concluded that classification accuracy could be improved by using more appropriate segmentation and feature extractions techniques.

In Mokhled S. Al-Tarawneh et al. [71], defined a method to detect the diseased leaves in olive. Initially, the colored RGB images were transformed into a CIELab color system. Moreover, to segment the region of interest of a leaf, the automatic polygon method is used. Furthermore, to define the masked polygon points of diseased segments, the edging contour of the entire image is applied. Finally, in the classification phase, fuzzy c-means clustering for statistical usage is employed to locate the defect and severity. The author justified that the experimental results of the combination between the FCM and polygon auto-cropping segmentation present remarkable precision prospects.

In Sekulska- Nalewajko and Goclawski et al. [72], discussed the effective method to detect disease symptoms in cucumber and pumpkin leaves. In the initial stage, the captured RGB images were converted to HSV color system. Later on, the brightness part of the leaf is ignored after the thresholding process. Afterwards, to sperate the pixels into clusters, Fuzzy c means clustering method is applied in hue-saturation space. The author concluded that the proposed method requires too many processes to obtain the desired results. However, it achieved good results in classification manner.

2.2 Theoretical Background

2.2.1 Machine Learning

Machine learning algorithms use statistics to find patterns in large amounts of data. Here, the data contains many elements, such as numbers, words, pictures and clicks. If it can be stored digitally, it can be used in machine learning algorithms. Machine learning is a process that provides many of the services currently in use. Reference system similar to that used in Netflix, YouTube, Spotify. Search engines like Google, social networks like Facebook or Twitter. Voice support for Siri and Alexa. The list goes on. In all these cases, each platform collects as much data as possible about you: what genre you like, what link you click, react and use machine learning to make advanced training hypotheses on what you need [26]. I needed more. Or, in the case of voice assistance, it's the word that most closely matches the strange sound coming out of your mouth, see Figure 2.1. However, machine learning methods are often classified as unsupervised or supervised, as described below.

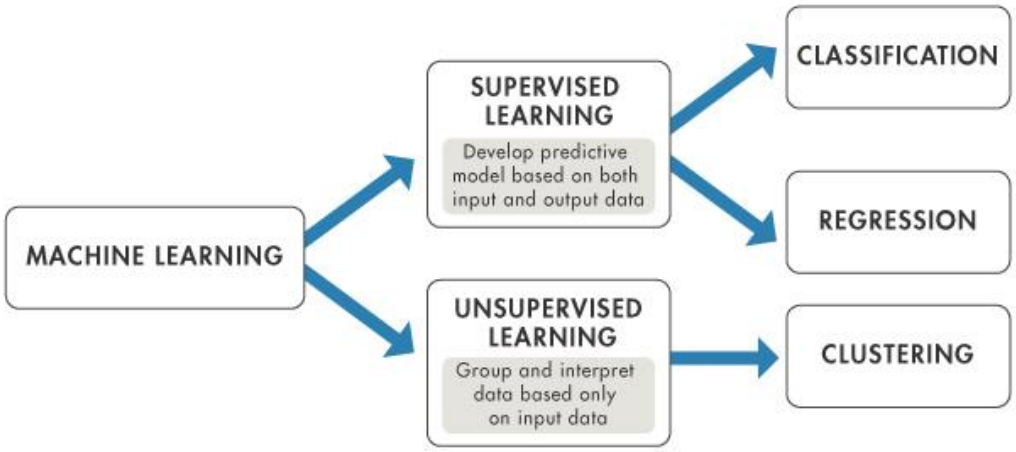

Figure 2. 1 Machine lerning structure [74]

2.2.1.1 Unsupervised Learning

The unsupervised learning algorithm is used to search for important features in large volumes of data. Here, the data contains many elements, such as numbers, words, images, and clicks. This can be done digitally, you can use it in an automatic machine learning algorithm. Automatic discovery is the process of providing most of the services currently in use. The referral system is similar to Netflix, YouTube, Spotify. Search engines like Google, Baidu social networks like Facebook and Twitter. Voice

support for Siri and Alexa, the list goes on. In all these cases, each platform collects as much data as possible about you: which genre you like, which links you click on to respond and use machine learning. Create the advanced learning hypothesis you need. We need it more [27]. Or, in the case of language support, this word is most often applied to strange sounds coming from your mouth. Several unsupervised techniques presented below:

- **Cell-based Clustering (CBF)**

This method is known in its abbreviated form by CBF. The key development idea of CBF is based on combating scalability issues that deteriorate other bottom-up approaches, such as CLIQUE and ENCLUS. The increment of cells resulting from dimensionality increase is solved in CBF by deploying an algorithm that continuously inspecting a dimension's maximum and minimum values in order to create the optimal partitioning. In addition, scalability is partially addressed by CBF, considering a large number of instances in a dataset. The problem here is tackled by the deployment of an improved data structure, filtering-based index, for storing cells, which allows faster access. Notwithstanding, CBF suffers from its sensitivity to section threshold, which determines the frequency of intervals in the same dimension, and to cell threshold which identifies the data points; minimal density in an interval. Mostly, all other advantages provided by bottom-up approaches are available with CBF, yet there is a slight deterioration in the clustering performance with CBF compared to other methods, however, the processing time is generally faster [49].

- **Density-based Optimal Clustering (DOC)**

DOC projective clustering is considered a hybrid approach, which combines both bottom-up and top-down approaches. Essentially, DOC considers a projective cluster, and it defines an optimal projective cluster in a mathematical way for this purpose. The projective cluster in DOC is established by determining a strong clustering tendency of a subset of instances C towards a subset of intervals D. In DOC, this is formed as a pair (C, D), and in order to create the aimed clustering process, DOC must identify the optimal pairs (C, D). Here, identifying the optimal pairs is done by

generating a small subset S, by any valid means, for example, random sampling. Instances in both C and S are compared and the absolute result of their difference should be less than or equal of the fixed-length, ω, which is a fixed subspace length provided by the user. This procedure is repetitively done, and both C and S are continuously obtained, using random sampling or any alternative method, until the optimal result is achieved. The minimum number of instances which can create a class, α, is also needed for the operational process of DOC. The minimum allowed density for a cluster is then determined by both parameters α and ω. It is also suggested that a trade-off should be made between the number of intervals and subspaces. For this specific purpose ano,ther parameter is used, which is usually referred to by β. This balancing process is at extreme importance for DOC, as the time complexity grows exponentially with respect to the intervals number and linearly with the cells number [50].

- **δ-Clusters**

This algorithm deploys the coherence between instances subset and attributes subset. The idea is based on calculating this coherence and using it as a distance measure for clustering purpose. The relationship they can be observed with coherent data samples is used for generating a proper cluster, as it is possible to discover instance based on their coherence. Using δ-clusters, cluster seeds are initially created, then the algorithm attempts iteratively to improve these clusters using a random shuffling technique, which tries different data samples and attributes to get a better clustering quality.

This process stops when the iterative process produces no further improvements. Two main parameters are needed for the operational process of δ-clusters algorithms, namely the size of the cluster and the total number of considered clusters. It is important to mention that the time complexity of δ-clusters depends on these two parameters. In addition, while implementing the δ-clusters method, it is recommended to maintain pre-knowledge or some sort of speculation of cluster size, as it makes difference with large datasets. Unlike other top-down approaches, δ-clusters deploys the coherence as a measure, which makes it a potential method for a number of applications, especially in the field of medicine [51].

2.2.1.2 Supervised Learning

Supervised learning is the most common area of machine learning today. Basically, the practice of new machine learning begins with a controlled learning algorithm. The first of these three messages therefore focus on supervised learning.

Controlled algorithms for machine learning are designed for learning, for example. The name "controlled" learning that teaches this type of algorithm comes from the idea that teachers are similar to controlling the whole process.

When learning a controlled learning algorithm, the training data consist of input data with correct output. During the training, the algorithm searches for data models that relate to the desired result. The learning algorithm for the control after the training takes a new hidden element and determines which designation the new element classifies according to the previous training data [28]. Several techniques are presented below:

- **Support Vector Machine (SVM)**

SVM is a discriminatory classifier that is formally defined by a separation hyperplane. In other words, with tagged training data (supervised learning), the algorithm generates an optimal hyperplane that classifies new examples. In two dimensions, this hyperplane is a line that divides the plane into two parts, each of which is on both sides [52]. Figure 2.2 illustrates support vector machine as a classifier.

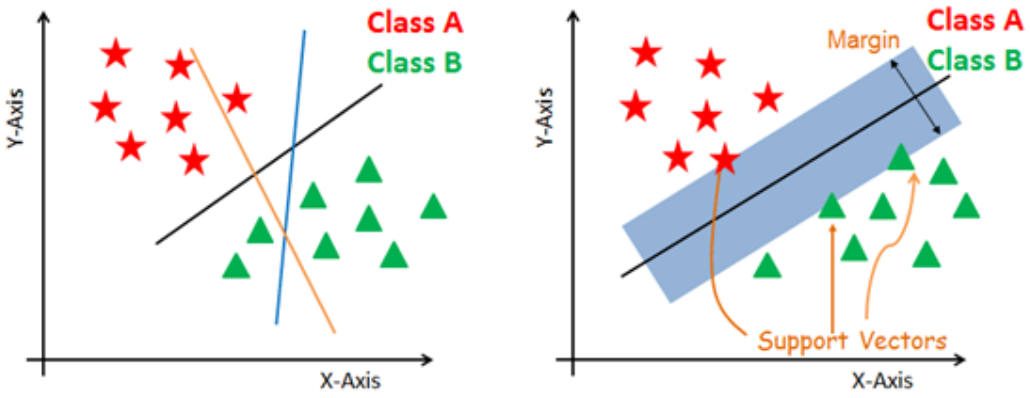

Figure 2. 2 Support vector machine

- **Naive Bayes (NB)**

The NB algorithm is a smooth classifier that compute a collection of probabilities by counting the combination and frequency values. NB classifier depends on Bayes theorem. Condition independence can be used in this classifier algorithm; in other words, it supposed that features value on a given class does not build on the values of other features [53]. NB is depending on supervised learning and it is a statistical classifier. Figure 2.3 shows the stages of Naïve Bayes and how can read data and then spilt training set data then use Naïve Bayes.

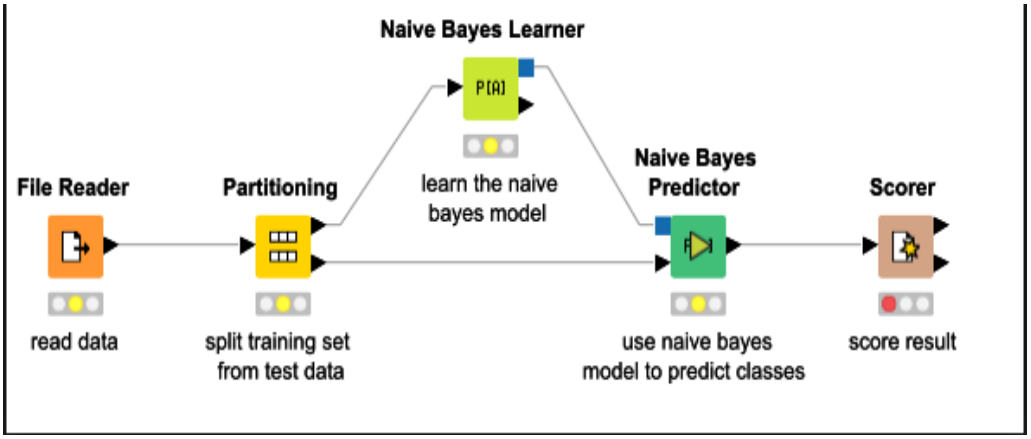

Figure 2. 3 Navie bayes

- **Neural Network (NN)**

Inspired by humans' brains, artificial neurons are the basic units of an Artificial Neural Network (ANN). These neurons are distributed over multiple layers, where the distribution of the neurons in the layer defines its operations and the position of the layer in the network defines its type. Each neuron collects its inputs from the previous layer, i.e. the outputs of the neurons in the previous layer. As shown in Figure 2.4, the output of the neuron is calculated by multiplying each input with a corresponding value, known as the weight, before summed up and passed through an activation function. This nonlinearity provided by the activation function allows the detection of more complex features, as the margins of the values that form the feature can be nonlinear. Moreover, to adjust the positioning of these margins in a more accurate way, an additional value, known as the bias value, is added to the inputs of the neurons, where the value of the bias is adjusted by the neural network depending on the need to position the margins of the detected feature [54].

Two types of computations are executed in any neural network, forward and backward passes. During training, both passes are required to update the weights and biases values, while when a prediction is required, only the forward pass is executed. In the forward pass, the execution of the computations is initiated from the input layer, where the output of each neuron in a certain layer is calculated and inputted to those in the next layer until the output of the network is calculated. The modification of the weights and biases values is known as the training of the neural network, where the optimal value per each weight and bias is calculated to achieve the required output. Several types of optimizers can be used to update these values depending on the difference between the output of the neural network and the values required to be outputted from it. Mainly, these optimizers are derived from the gradient descent algorithm, which relies on the rate of change of the output per each value to optimize it [55].

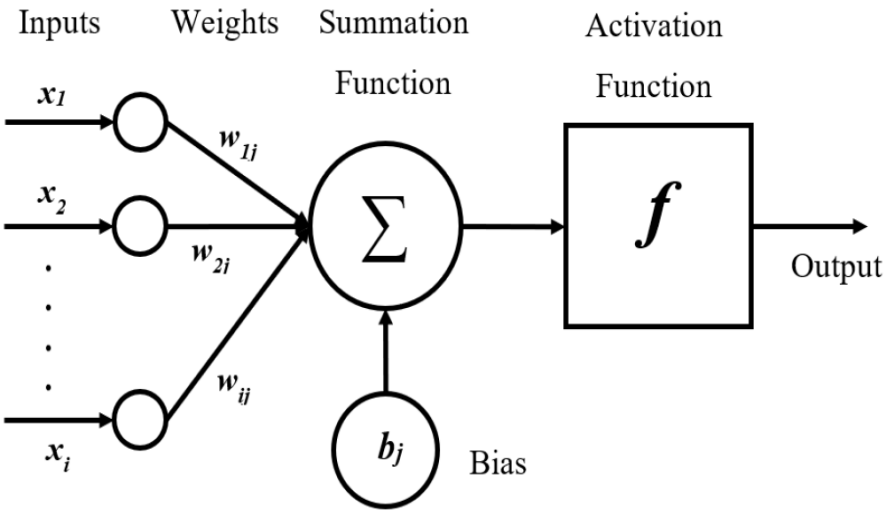

Figure 2. 4 Basic neural network structure

- **Multi-Layer Perceptron (MLP)**

Literature anatomy reveals a persistent implementation of Feed Forward NN, from among the several categories of link for 'artificial neurons'. A type of feed forward NN technicality is the MLPNN. The design of MLPNN is shown in the Figure 2.5. One of the most famous models in ANN is MLPNN.

Three Types of layers in the 'MLPNN': the first type is an input layer, the second one is a hidden layers. The third one is output layer type (several nodes in each layer). Information passes from one node to the next node. Outer nodes transform signals to the input layer. The 'output of first layer' passes to second layer over "weighted connection line". It accomplishes computations and sends the result to final layer (output) over weighted links. After that the output of second layer is send to last layer, for a final result [56].

The working steps of (MLPNN) are given in the following:

(1) Input data is given to the first layer for treatment which leads a divine output.

(2) The error is calculated by the difference between the predicted output & actual output.

(3) Back-Propagation algorithm has used to adapt the weights.

(4) For adjusting weights, it begins from weights between last layer nodes and last hidden layer nodes and works backwards through the network.

(5) When BP is finished, the operation starts again.

(6) The operation is renewed until the error between actual & predicted output to reach the least difference.

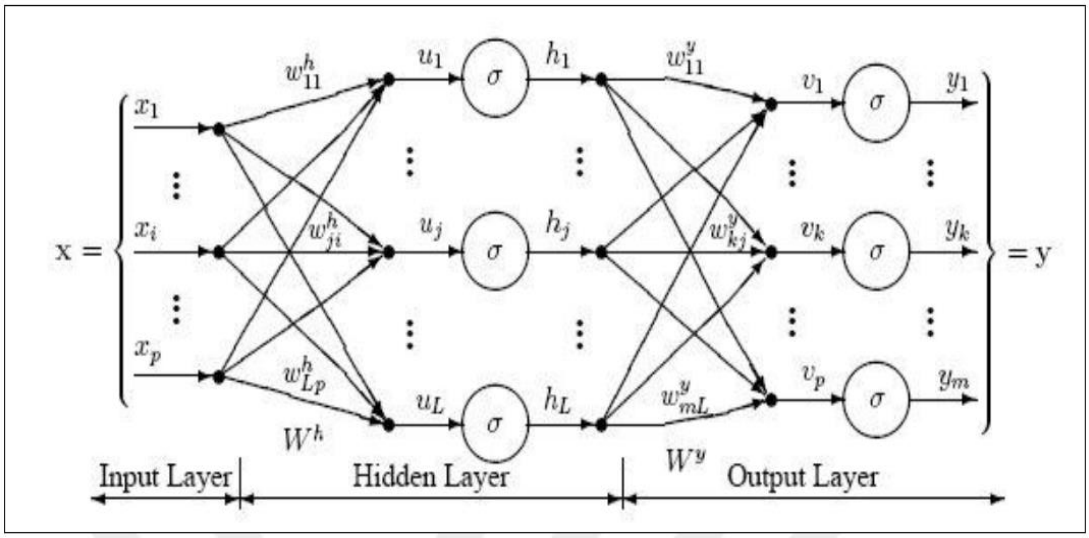

Figure 2. 5 Multi-Layer Perceptron

2.2.2 Deep Learning

Deep learning is a subfield of machine learning, it was developed using traditional flat neural networks, the biggest difference being the number of hidden layers. Hidden layers project inputs into multidimensional space. Here you can analyse the input data from different angles. Nearly Hidden layers allow more data to support the stochastic model discovered. There are several deep learning techniques such as: convolutional neural network (CNN), recurrent neural network (RNN) and Long short-term memory (LSTM).

2.2.2.1 Convolution Neural Network

Deep Neural Networks consist of layers where each layer contains neurons. Neurons in the layers are composed of weights and biases. When a neuron receives an input, it applies multiply and adds operations to the input using its weights and bias. The way those multiplication and addition is applied is based on the type of the layer. Besides, the type of the layers differs in terms of not only operations but also neuron connectivity patterns. For each different type of network, there might be a different connectivity pattern for neurons in successive layers. When a neuron produces its output, it may apply a non-linearity to its output before passing it to the next neuron. To apply a non-linearity, layers use activation functions. One purpose of using activation functions is to deactivate some of the connections in the network if that specific information is not important for the final decision of the network. Another purpose is to increase the importance of the connections that contain important information for the final decision of the network. Activation functions are one of the most important parts of the neural networks. They introduce non-linearity; thus, neural network learns and makes sense of complicated and complex functional mappings.

After one iteration of the given data through the network, the prediction of the network is obtained. The training procedure for a network consists of multiple iterations. After each iteration of the training, weight, and bias values are updated in the layers so that they can learn information from the given data. The updating procedure of weights and biases is called back-propagation. The main purpose of back-propagation is to update parameters in the network so that the predicted output of the network is close to the expected output. If the predicted output of the network is far from the expected output, then there will be high loss value based on the type of loss function. Loss value is the metric that shows how far the network is from predicting accurate results. There could be different loss functions based on the data and the problem. General working principles of a deep neural network and important factors are given above. The principles and important factors above may work differently for different types of deep neural networks.

There are many different types of deep neural networks such as Recurrent Neural Networks, Long-Short Term Memory Networks, Multilayer Perceptron's, Convolutional Neural Networks and more. The working principles of deep neural networks given above will be explained in detail in the next sections. Also, there will

be more detailed information about Convolutional Neural Networks (CNNs) and how they work as algorithms described in this work are more focused on CNNs [31]. The essential layers of CNN are presented below:

- **Convolution Layer**

Convolutional layers are used widely in deep neural networks. They are the core information extractors, and most of the computations done in deep neural networks are done on these layers. A simple convolutional layer is composed of a set of filters and a bias parameter. Filters are used to extract local information from the input such as images. At the beginning of the training, the weights of filters are set randomly. During the training process, those filters are updated and their weights are computed (learned) so that they can detect features from the input images. Extracted features from first convolutional layers are low-level information such as edges, and extracted features from later convolutional layers are high-level information such as the wheel of a car. As data flows through the network which is also called forward-pass, the filter of convolutional layers is slid across the width and height of the input volume. During this sliding process, the dot product of the filter and the current slide of the input is calculated. Resulting dot products are also called activation maps or feature maps which represent the response of the filter at every spatial position. Eventually, the network will learn to activate specific filters as it detects extractable information on the image. As shown in the Figure 2.5, let us say we have an image of size [32x32x3], where the first and second dimensions represent width and height, and the third dimension represents colour channel dimension of the image such as RGB (Red-Green-Blue). If we a have filter with dimension [4x4] (width and height), each neuron in the convolutional layer will have a weight tensor with dimension [4x4x3]. The last dimension is 3 because input has 3 channels and a different filter is needed for each channel of the input. Furthermore, if the convolutional layer consists of 32 neurons, the dimension of the layer becomes [32x4x4x3], where each parameter is trainable [37]. As mentioned earlier, the filter is slid across the input image and the output dimension is dependent on the sliding length, which is also called stride. When stride is set to 1, filter slides on one pixel at a time for each dot product operation. When the stride is set to n, filter slides on n pixels after each dot product operation. Increasing

the stride value will result in smaller output volumes as the filter skips a higher number of pixels. Another parameter for the convolutional operation is called padding. This parameter pads zeros around the border of the input to preserve the spatial size of the input volume so that the input and output width and height are the same. Both stride and padding are hyperparameters for convolutional layers, and they need to be optimized based on the problem. The sliding of convolutional filters can be applied to different dimensions based on the input. 1-dimensional (1D) convolution implies that sliding of the filter is done on 1-axis, which can be time. The output of 1D convolutional is a 1D array. 2D convolution implies that sliding of the filter is done on 2-axes (x, y), and outputs 2D matrix. 3D convolutional implies that sliding of the filter is done on 3-axes (x, y, z) and output has 3D volume. In most of the convolutional layers, 2D convolutions are applied on 3D data as there is a filter for the third dimension. On 3D data, the 2D filter is slid across the x and y-axis. Therefore, the output is also a 2D matrix with 3D volume.

As shown in the Figure 2.6, the connectivity pattern of convolutional layers also differs from fully-connected layers. Convolutional layers are locally connected layers. If the input image dimension is [32x32x3], which contains 3 colour channels, and a neuron inside the convolutional layer contains [4x4] filters, there will be a different filter for each channel in the total dimension of [4x4x3]. Each [4x4] filter inside a neuron is convolved with a single channel of the input image. Therefore, there is a local connectivity pattern in convolutional layers, instead of connecting each filter with each channel of input which is seen at fully connected layers [38].

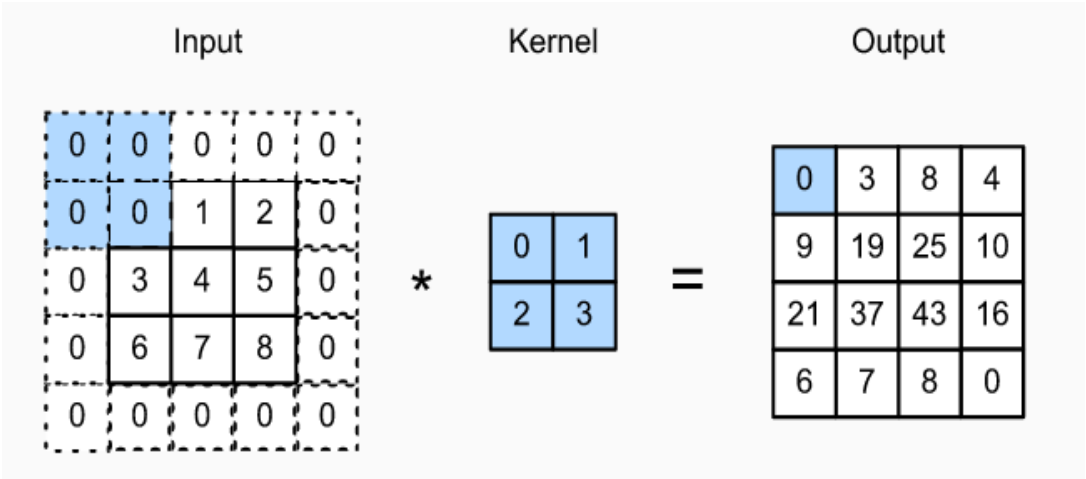

Figure 2. 6 Convolution layer

- **Pooling Layer**

The pooling layer is introduced among convolution layer in CNN architecture. The pooling is a function to decrease the longitudinal size of the features to decrease the amount of parameters and calculations in NN then control the overfitting.

Pooling layers are also widely used in convolutional neural networks. The main purpose of using pooling layers is to reduce the spatial dimension of the feature maps. Thus, the amount of data that is transferred between layers will be less and there will be less amount of communication. While pooling layers reduce the dimension of the feature maps, they also extract meaning and powerful information from small slides of the data as can be seen in the Figure 2.6.

There are several types of pooling layers such as max pooling, average pooling, and sum pooling. Max pooling layer is the one that is generally used in convolutional neural networks to reduce the dimension of the feature maps. In Figure 3.3, max-pooling layer is visualized. Let's say we have a max-pooling layer with a filter size of [2x2] and stride of 2. Then, the maximum value of each [2x2] part of the feature map is taken. Since we have stride value as 2, the filter is slid 2 pixels after each pooling operation on both axes. In Figure 2.6, input size [4x4] is reduced to [2x2] after the max-pooling layer and only important features are extracted while reducing the dimension. Similarly, sum pooling applies summation operation to the slices of input

instead of extracting maximum value. Besides, since average pooling takes an average of each slice, it can be considered as a smoothing operation [39].

Figure 2. 7 Max-pooling layer

- **Activation Function**

Activation functions are mathematical equations that determine the output of a neural network, it plays an important role in the output of the deep learning model, the computational efficiency of training, and its accuracy. Moreover, it helps in normalizing the output of each neuron to a range between 1 and 0 or between -1 and 1. The activation function is like a mathematical gate between the input of the current neuron and its output which going to the next layer. It can be simplified as a step function that turns the output on or off, controlled by the threshold value.

There are various kinds of activation functions, each one has a different work characteristic, the most commonly used one is the Rectified Linear Unit (ReLU) which is a kind of linear activation function.

ReLU function returns 0 if it receives a negative input, but if it is a positive value. See Figure 2.7. The Equation (1) represented the ReLU model.

$$f(x) = \max(0, x) \quad (1)$$

Where x is the input features, and represented the output with the positive values and all the negative values converted to zero.

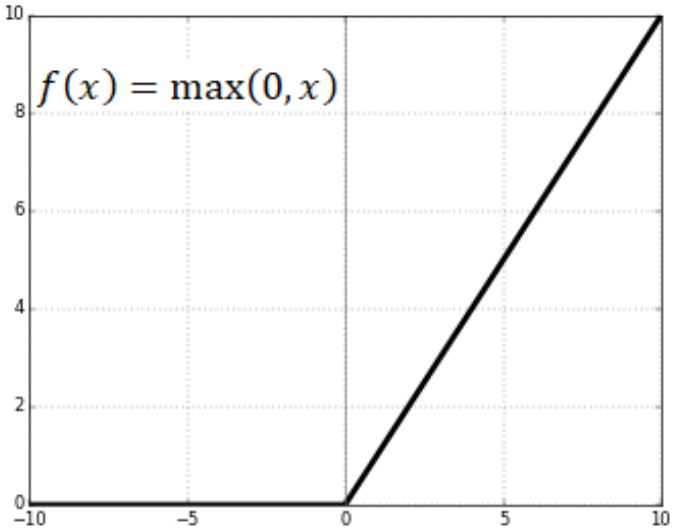

Figure 2. 8 Rectified linear unit

- **Fully Connected Layer**

Fully connected layers apply linear transformations to the given input vector before applying activation functions and have full connection to the neurons in the previous layer. All of the neurons in the fully connected layer are multiplied by neurons in the previous layer and if there is a bias in the fully connected layer, it is added to the resulting linear transformation. We can formulate the operation done by a fully connected layer as follows. Let us say the weights of the fully connected layer are denoted by W and have an input dimension of k. Then we can formulate the operation done by a fully connected layer with respect to the given input x as follows in the Equation (2) [35].

$$y = W * x + b \tag{2}$$

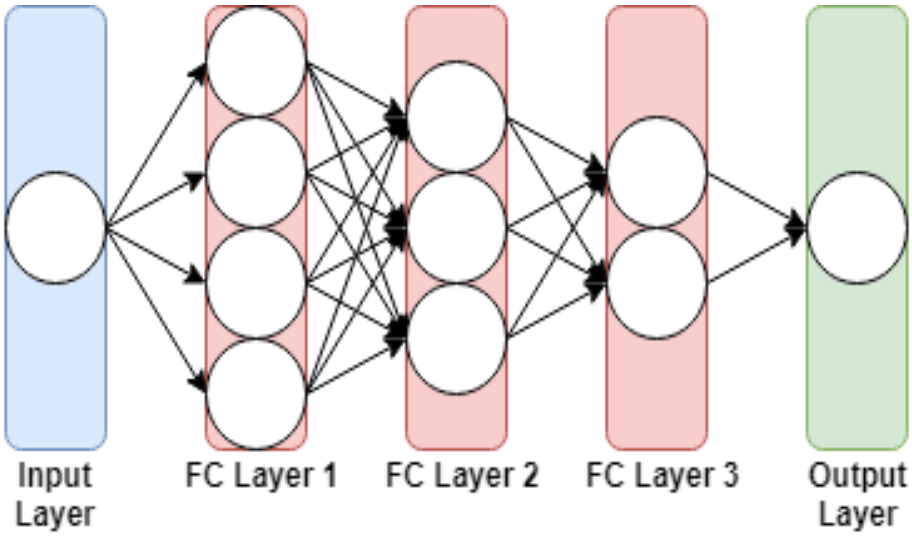

Figure 2. 9 Fully connected layers

In Figure 2.9, connectivity pattern of fully connected layers is shown. The fully connected layers with red colour are also called hidden layers. Hidden layers 1, 2, and 3 in the representation are fully connected layers and neurons in those layers apply a linear transformation to the neurons in the previous layer. For hidden layer 1, the input dimension is 1 and the output dimension is 4 as it includes 4 neurons.

Similarly, the input dimension of hidden layer 2 is 4 since the input for hidden layer 2 contains 4 neurons. The output layer is also fully connected. This is because it is connected to all of the neurons from the previous layer and outputs 1-dimensional result. In convolutional neural networks, fully connected layers are usually used at the end of the network after a series of convolutional layers.

Convolutional layers which will be discussed next are good at subtracting local information from the data and connecting that information with fully connected layers at the end of the network is good for generalizing the local information and obtaining a result with the given number of neurons [36].

- **SoftMax Classifier**

The SoftMax is probability technique which estimate the probabilities between (0,1). The model mathematically can present as shown in Equation (3):

$$SoftMax(y) = \frac{e^{xi}}{\sum_{k=1}^{k} e^{xak}} \quad (3)$$

Which the x represented the input feature that become output from fully connected layers as matrix of arrays and each array represented one example of data. The i indicate the output units and must be integer value, so i = 1, 2, ..., K. K represented the total of the number of examples. Represented the predicated labels of the pixels by the SoftMax. The accuracy of this method can be calculated easily by calculating the difference between the actual labels that are selected manually and the predicated labels by the SoftMax.

2.2.2.2 Residual Neural Networks

The primary idea of ResNet was motivated by constructs known from pyramidal cells in the cerebral cortex. However, deep residual networks (ResNet) are a deep convolution neural network which is stacked by a set of residual blocks after each layer to employ the shortcuts to jump over some layers that not trained correctly, where the input from previous layer is added directly to the output of the other layer. Therefore, this process helps to solve the gradient vanishing problems. As well as, it helps to train networks which have a large number of layers easily with low error rate. A residual block is illustrated in the Figure 2.10. Let us suppose that we try to learn an H(X) function that is the mapping from the input X of a few convolution layers to the output of them. If we add the input X to the output, we let the convolution layers to approximate F(X): = H(X) +X. Learning F(X) function is easier than learning H(X) [73].

The specification for 5 versions of ResNet with different number of layers are reported in the Table 2.1.

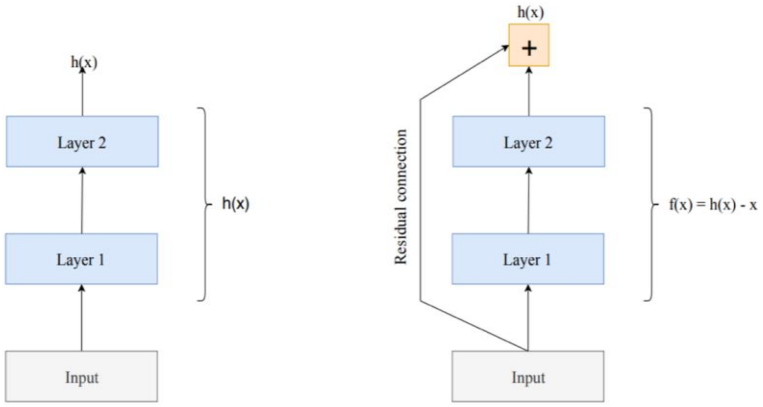

Figure 2. 10 Residual learning block

Table 2. 1 The number of the parameters of the Residual Neural Networks

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$
conv3_x	28×28	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times8$
conv4_x	14×14	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times23$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times36$
conv5_x	7×7	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

CHAPTER 3

MATERIAL AND METHODS

3.1 Proposed Methodology

In this study, we introduce an effective method to segment apple leaf spots. In the initial stage, the Plant Pathology 2020 - FGVC7 dataset is used which includes colored images with a resolution 2048 × 1365 × 3 for various diseased leaf images categories such as apple scab, cedar apple rust, multiple diseases and normal leaf.

After analyzing the data, the imbalance issue has found for each category. Therefore, all classes are setting to 91 samples which are the minimum samples number are existing in multiple diseases class. The goal of this step is to find out the ability of these proposed networks to deal with imbalanced data issue when used as a subnet for features extractor in the detection model.

Later on, ground truth images are manually generated by using the local graph cut tool in MATLAB image segmenter application. The foreground of the image includes disease spots which labelled as white pixels. On the other hand, the rest of the image does not include any important features. Therefore, it considered as a background and labelled as black pixels. The aim of this step is to train and test the model in supervised vision by applying semantic segmentation method. Afterwards, to avoid the exhaustion of the GPU memory, all original images and their ground truth are stored as pixel labelled image data after rescaling process into 256 x 256 x 3. The stored data is divided into a training set 70 %, validation set 15% and testing set 15%. As shown in the Figure 3.1 which depicts the flowchart of the proposed method.

In the next step, two different architectures of convolution neural network, U-Net and ResNet are trained and tested individually on this dataset. In other words, each structure has trained and tested on each class separately. The aim of this step is to find out which structure obtain better prediction mask, and how it behaves with the high differences in disease spots shape in each class.

Furthermore, the extracted features from each structure have classified by using SoftMax classifier. Finally, the segmentation performance for each structure has tested using 14 image samples for each category, then compared and evaluated with the true label of each class by applying four metrics, including pixel accuracy, sensitivity, IoU and F1-score metrics. As well as, we compared these four metrics with the existing image segmentation methods, including K-means, Adaptive image threshold using local first-order statistics, Otsu thresholding method, Texture filter and Active contours technique.

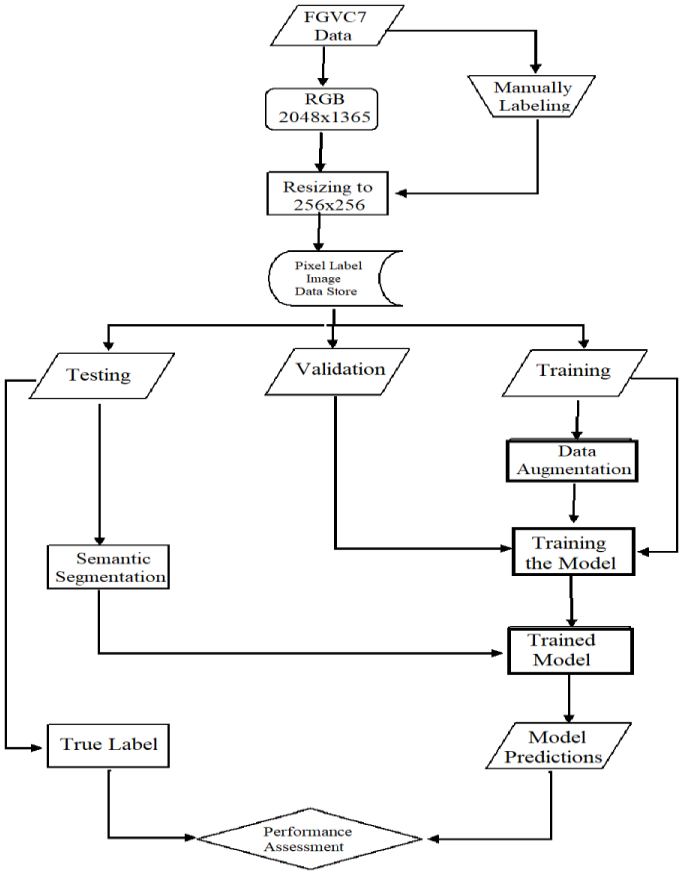

Figure 3. 1 Depicts the flowchart of the proposed method

3.2 Data Description

The data used in this study were collected from the Kaggle community for ‘Plant Pathology Challenge’; part of the Fine-Grained Visual Categorization (FGVC) workshop at CVPR 2020 (Computer Vision and Pattern Recognition).

However, the data includes high-resolution coloured images of multiple apple foliar disease symptoms such as apple scab, cedar apple rust, Alternaria leaf blotch, frogeye leaf spot, and healthy leaves. Moreover, the real-life images were captured during the 2019 growing season from commercially grown cultivars in an unsprayed apple orchard at Cornell AgriTech, Geneva, NY by using a Canon Rebel T5i DSLR and smartphones cameras under different conditions such as angle, lighting, noise and surface.

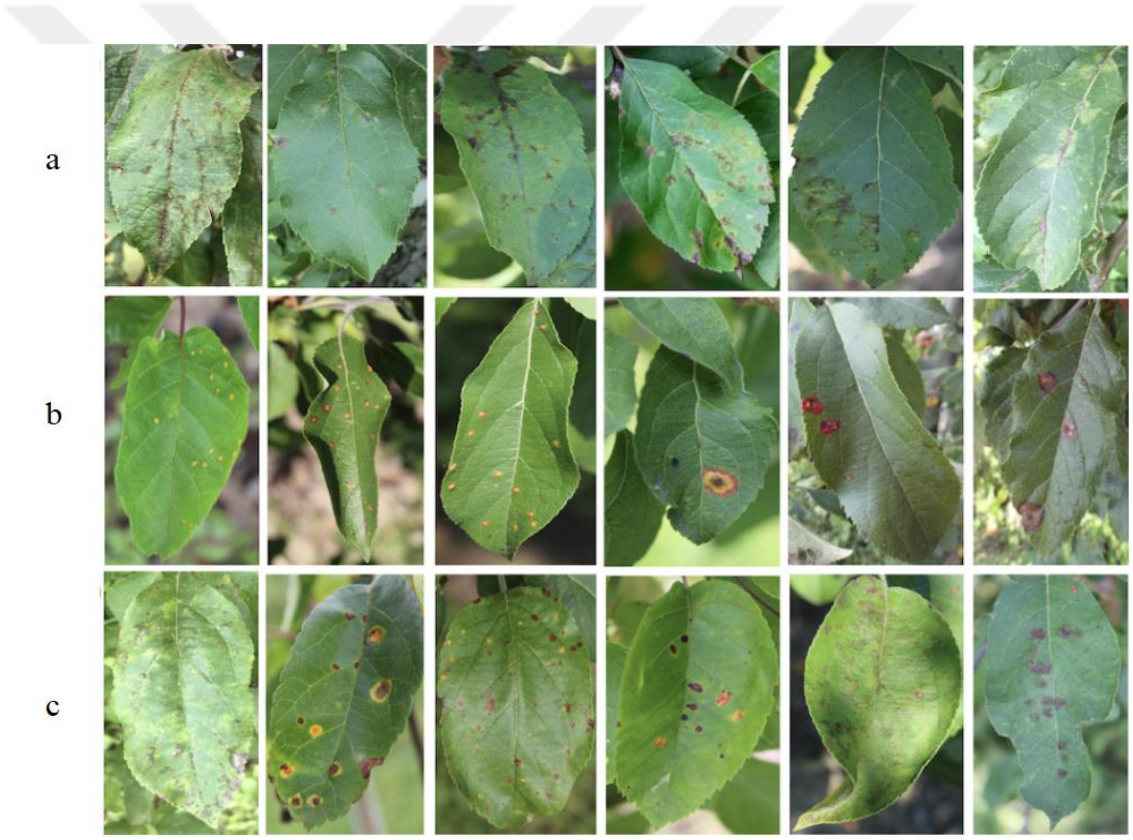

Figure 3. 2 Demonstrate the dataset samples of the three types of apple leaf diseases symptoms: a) Apple scab, b) Cedar apple rust and c) Multiple diseases in a single leaf. As well as, the differences in age of the leaf, capturing angle and illumination.

The images of apple scab, cedar apple rust and healthy were annotated manually depending on their distinctive symptoms and ease of classification. On the other hand, due to the difficulty and complexity of diagnosing the symptoms in Alternaria leaf blotch and frogeye leaf spot, and complex disease symptoms from multiple diseases

in a single leaf, the annotations of these diseases were obtained under the guidance and approval of a pathologist expert. See Figure 3.2.

Furthermore, more images were added to the dataset to increase the complexity such as images taken at different times of day, more than one leaf in the same image, multiple diseases in the same image, non-homogeneous background of images and imbalanced dataset of different disease categories. Finally, the total number of samples in this dataset is 3651, where 1821 images for training with annotations and 1830 images for testing without annotations [30].

Therefore, in our experiment, only the training set has been used, which include 91 images sample of multiple diseases leaf, 622 images sample of Cedar apple rust, 592 images sample of apple scab and 516 images sample of a healthy leaf.

3.3 Architecture Used

In this section, we explain two different deep architectures used for segmentation, U-Net and ResNet, as follow.

3.3.1 U-Net

U-Net is a fully convolutional encoder/decoder network proposed in for the medical image segmentation task. The modularity of U-Net gives the ability to combine different encoder and decoder networks. The encoder part of architecture provides high-level spatial features of images. Classical CNN architectures for classification aggregates these features on final dense layers to further classification of images. In U-Net, this encoder network followed by a decoder network which contains deconvolution layers. It has n convolutional and pooling layers in the encoder part of the network. This coupling is symmetric, decoder network in U-Net has the same number of (n) deconvolution layers to recover low-level features and create a high-level label map. Skip connections between corresponding convolution and deconvolution layers make overall architecture robust and reliable [41].

In the encoder side, convolution layers and pooling layers down-samples image at each convolution/pooling block. The decoder part uses transposed convolution

(deconvolution) operations and skips connections to recover the image created by the encoder block. In Figure 3.3 which depicts the U-Net architecture used in this work.

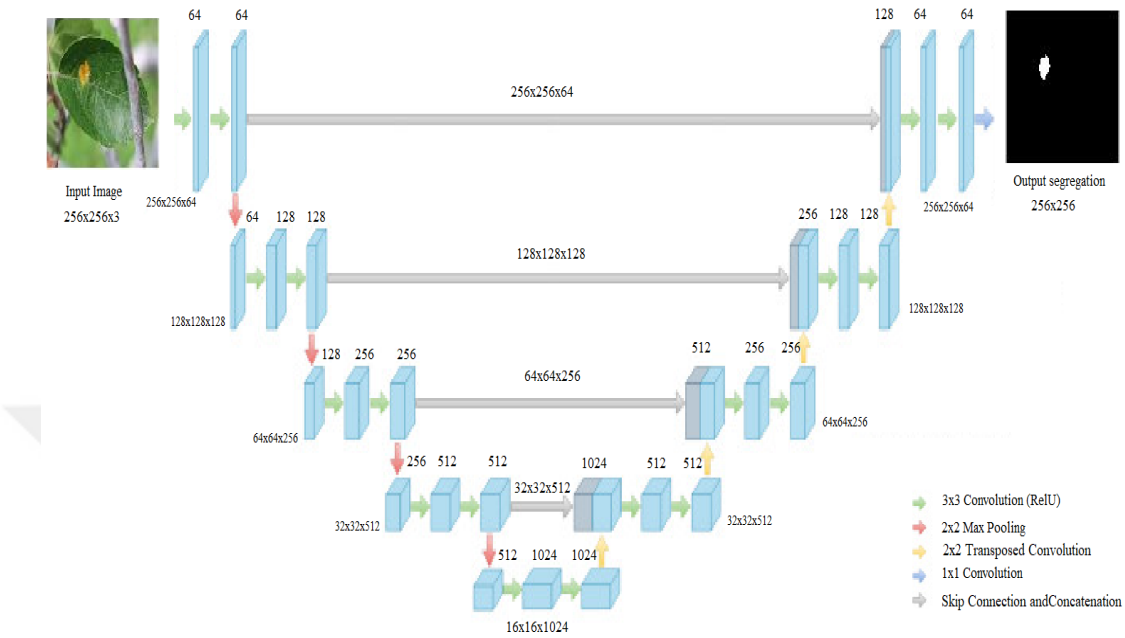

Figure 3. 3 U-Net architecture

3.3.2 ResNet

In our experiment, we use ResNet 18 which is a convolutional neural network that has 18 layers deep with an image input size of 224 by 224, this network trained on 1000 object categories, as well as it showed efficient results in semantic segmentation applications, it can be described as in the Figure 3.5. In our experiment, we modified the network input to 256 by 256 to fit the size of our images, this process leads to increase in the encoder depth of the network which increases the layers into 100, as can be seen in the Figure 3.5.

However, ResNet architecture consists of one contracting path which works to down sample the input image in each layer; this path resembles the typical architecture of a convolutional network. First convolution layer made of 7x7 feature mask with stride [2 2], and padding [3 3 3]. However, the other convolutions layer made of 3x3 kernel

size with stride [2 2], and padding [3 3 3], each followed by batch normalization, a rectified linear unit (ReLU) and a 2x2 max pooling operation with a stride 2.

Finally, after a sequence of the down sampling process and to meet the input size at the same level as the first convolution, therefore, the final layer will be a feature map of 256x256x2 which include the most important features which will pass through SoftMax and pixel classification layers to calculate the difference between the actual labels that are selected manually and the predicted labels of the network.

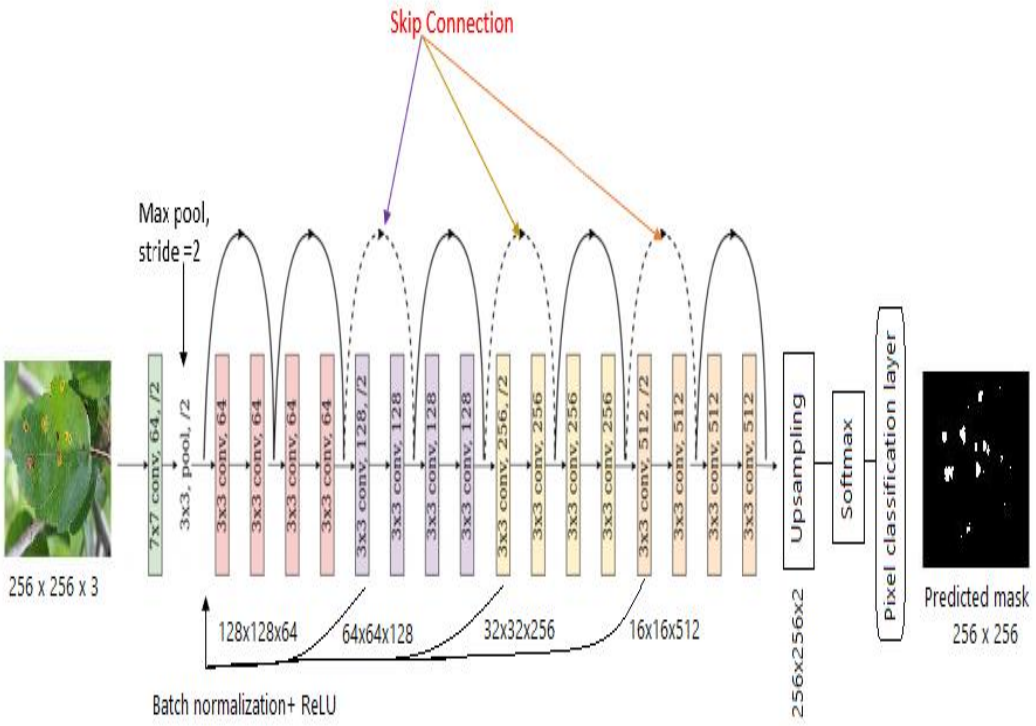

Figure 3. 4 ResNet architecture

3.4 Semantic Segmentation Concept

Image segmentation task aims to locate separate objects or boundaries of objects within an image to make further regional analysis efficient. It requires the formal definition of low-level properties such as color or texture or spatial constraints on

objects to extract the target area. It detects these formally defined regions and does not assign any label on segmented area. Semantic segmentation of the image is an essential step for the ultimate computer vision goal of image or scene understanding. Early research on semantic segmentation primarily utilizes Conditional Random Fields (CRF). CRF defines a probabilistic graphical model to estimate pixel labels.

This probability model mainly based on pixel neighborhood characteristics. Consequently, pixel-level inference in CRF considers both spatial and contextual information of the image. As in other problem domains in computer vision, deep learning dominated the research on semantic segmentation.

The invention of convolutional neural networks provides the ability to learn the spatial relations between complex features while preserving the original input structure. It relieves researchers in various fields such as natural language processing, computer vision, etc. from manually discovering and extracting those features. These features then used to represent the class of that image. In the case of semantic segmentation, rather than the class of image, classes of individual pixels need to be determined. Successive convolutional and pooling operations in CNN produce representations of strong features in the image. The characteristics of these strong features generally correspond to individual objects in an image scene. To transfer this high-level information of labels to original image plane deconvolution (transposed convolution) operation is proposed [42].

For semantic segmentation, this approach corresponds to fully convolutional deep learning architecture where output is a label map with original image size, see Figure 3.6

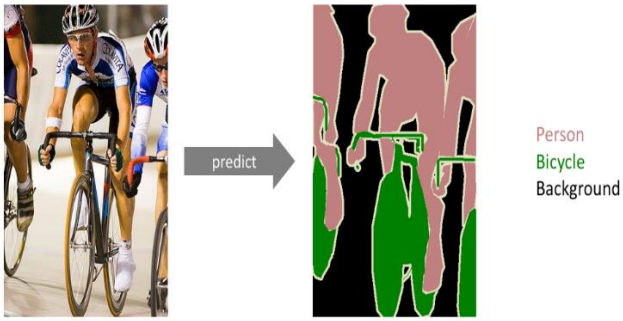

Figure 3. 5 Semantic segmentation example

3.5 Data Augmentation

In deep learning approach, data augmentation is an influential method that plays an important role to avoid overfitting issue caused by bounded training samples. However, data augmentation approximates the data probability space by manipulating input samples, such as vertical flipping, horizontal flipping, random rotation, random crop, noise disturbance. Generally, the performance of the model can be improved, as long as the quantity, quality, and verity of the data in the dataset are increased [29].

As mentioned previously in the proposed method section, all classes have set to 91 samples, because of an imbalanced data issue. However, the training set samples will be not sufficient to train the network effectively. Therefore, we increased the training set samples with their true labels to be ~256 samples for each category by applying some augmentation process, including vertical flipping, horizontal flipping and random rotation (10^o, -10^o). Three generated samples from the original training image and the corresponding labels images are shown in the Figure 3.7.

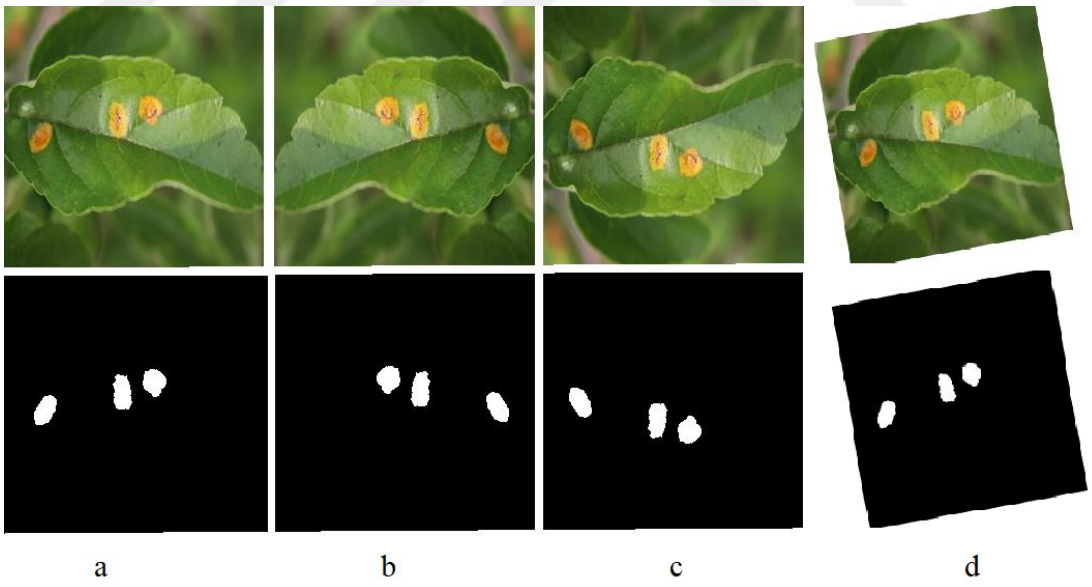

Figure 3. 6 Data augmentation samples: (a) orginal, (b) vertical flipping (c), horizontal flipping, and (d) random rotation between (10^o, -10^o).

Therefore, this process works effectively to increase the number of samples of training set randomly during the training process, which leads to increase the overall performance of the model.

3.6 Networks Training

The training experiments of both networks have implemented by using MATLAB R2020b software which installed on GT75 Titan 8RG laptop which has Intel CM246 CPU and NVIDIA GeForce GTX 1080 with 8GB GDDR5X.

However, each structure has trained using the augmented data for each class separately. Moreover, in the optimization process, to gain the best performance, several training parameters have taken into account, like tuning batch size, epochs, learning rate, optimizers like SGDM and ADAM, loss functions, and dropout. These parameters play an important role in speed up the training process and a marked improvement in predictions of any network. Therefore, the best performance of U-Net network was obtained using Adam method with a learning rate of 0.000315 with 4 mini-batch size, 50 epochs and 0.5 dropout value which help to make a noticeable speedup in the training process, where the training time was enhanced to 4-6 minutes with` a better prediction when tested. On the other hand, the training of modified ResNet 18 using Sgdm method with a learning rate of 0.06 with 2 mini-batch size, 30 epochs and 0.5 dropout value which showed better predictions than U-Net when tested with the test set.

The purpose of using these optimizers is to find the minimum point for a given loss function. In the classical GD method, weight update operation occurs after all input data fed to the network. Calculated overall training loss information for that iteration then used in weight updates. Both optimizers are described below:

- **Stochastic Gradient Descent (SGD)**

A widely-used gradient-descent based minimization method is Stochastic Gradient Descent (SGD). The difference between standard GD and stochastic one is, in SGD,

weight update occurs after every single training input fed to the network. It is also possible to update weights after a subset of training inputs fed to the network. This approach is called Mini-Batch Gradient Descent and provides more stable convergence to a minimum. In mentioned gradient descent optimization algorithms noise in training data may cause sudden changes in the gradient direction. This causes unstable changes in parameter values, thus reaching to convergence point takes more steps than optimal. In SGD this problem solved by adding an extra regularization parameter called momentum. It takes gradient information from previous iterations into consideration and enables more stable convergence to a minimum [43].

- **Adaptive Moment Estimation (ADAM)**

Adaptive Moment Estimation (Adam) is an extension of SGD that adds a regularization factor for learning rate. It calculates an adaptive learning rate for each parameter in the network with respect to gradient information for that particular iteration. This adaptation is realized by calculating first and second moments of gradients. The moving average of these two moments determines the learning rate, thus convergence speed. This averaging mechanism also provides stable convergence in the case of sparse gradients and handles divergence that fixed learning rate may cause [44].

3.7 Network Testing

In the testing phase, to verify the segmentation performance of the proposed networks, 14 samples for each category have used. Since the prediction mask of the network is an array that contains values assigned to categories that can not display as an image. Therefore, it should be converted to a grayscale image. This process leads in some images to lose some pixels from the desired area, sometimes make it bigger or smaller. However, to estimate the accuracy of the segmented parts of the test set images with their true labels, the below explained metrics including pixel accuracy, sensitivity, IoU and F1-Score have applied.

- Pixel Accuracy

Pixel accuracy is the simplest way to evaluate the performance of the model on image segmentation, basically, it measures how the model classified pixels between the predictions and the ground truth, as shown in Equation (4):

$$Pixel\ Accuracy\ = \frac{TP + TN}{(TP + TN + FP + FN)} \quad (4)$$

- Recall

Recall calculates the number of the correct boundaries that find by the model, it is also known as Sensitivity or True Positive Rate (TPR) and it can be represented in Equation (5):

$$Reacall\ = \frac{TP}{TP + FN} \quad (5)$$

- F1-Score

The F1- Score metric measures how close the point on the predicted boundary matches the point on the ground truth boundary. In other words, it calculates the harmonic mean of precision and recall. Therefore, this score takes both false negatives and false positives into account. If the numbers of false negatives and false positives are different, the result of this metric will be close to zero, see Equation (6):

$$F1\ Score\ = \frac{2 * TP}{2 * (TP + FP + FN)} \quad (6)$$

It may be concluded that the closer the F1 score to 1, the better the pixel classification performance.

- Intersection over union (IoU)

Intersection over Union (IoU) metric, also known as the Jaccard index, is a method that calculates the intersection and union between the prediction and the ground truth, and it can be represented as the Equation (7):

$$IoU = \frac{TP}{TP + FP + FN} \quad (7)$$

CHAPTER 4

EXPERIMANTAL RESULTS AND DISCUSSION

4.1 Experimental Results

In this section, we illustrate the performance of segmentation accuracy of our proposed networks and the other segmentation methods on 14 test samples, after calculating the average of the four metrics each class; however, the obtained results presented as follow:

- Segmentation accuracy of healthy leaves

Table 4. 1 Demonstrates the average of proposed networks compared with other segmentation methods tested on 14 test samples of healthy leaves.

Methods	Metrics			
	Pixel Acc. %	Recall %	IoU %	F1-Score %
ResNet	96.59	96.33	92.59	74.32
U-Net	91.91	90.31	83.15	51.04
Region Growing	88.75	88.79	70.67	50.52
K-means	50.14	56.06	27.17	19.32
Otsu thresholding	56.31	34.14	21.98	25.13
Local first-order statistics	48.87	77.44	32.79	15.71
Texture filter	52.71	18.37	11.18	22.53

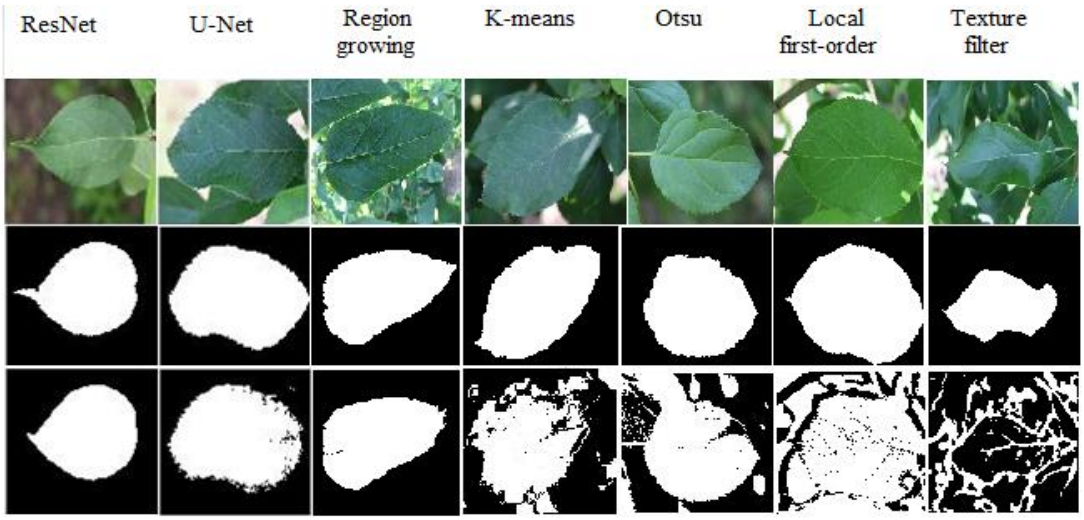

Figure 4. 1 Healthy leaf images (first row), true label images (second row), predicted mask of each method (last row).

- Segmentation accuracy of infected leaves with cedar rust

Table 4. 2 Illustrates the average of proposed networks compared with other segmentation methods tested on 14 test samples of rust leaves.

Methods	Metrics			
	Pixel Acc. %	Recall %	IoU %	F1-Score %
ResNet	99.71	78.36	76.44	87.27
U-Net	99.75	87.60	81.63	90.75
Region Growing	97	72.65	32	49.18
K-means	38.4	80.97	1.02	3.62
Otsu thresholding	53.46	81.87	1.04	4.73
Local first-order statistics	28.62	97.14	1.42	2.66
Texture filter	74.59	81.2	1.88	3.75

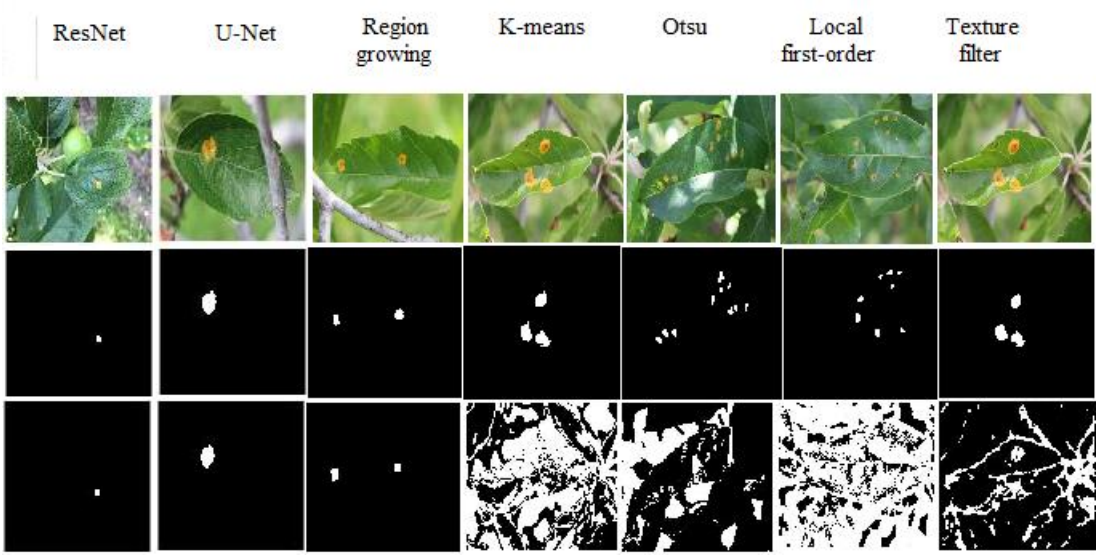

Figure 4. 2 Cedar rust leaf images (first row), true label images (second row), predicted mask of each method (last row).

- Segmentation accuracy of infected leaves with scab

Table 4. 3 Outlines the average of proposed networks compared with other segmentation methods tested on 14 test samples of scab leaves.

Methods	Metrics			
	Pixel Acc. %	Recall %	IoU %	F1-Score %
ResNet	98.19	61.79	59.43	64.78
U-Net	97.62	69.71	61.41	63.75
Region Growing	87.99	58	9.26	28.65
K-means	51.14	49.08	1.87	9.40
Otsu thresholding	60.09	50.67	1.23	7.43
Local first-order statistics	25.69	68.97	1.75	9.61
Texture filter	73.86	48.77	4.30	11.50

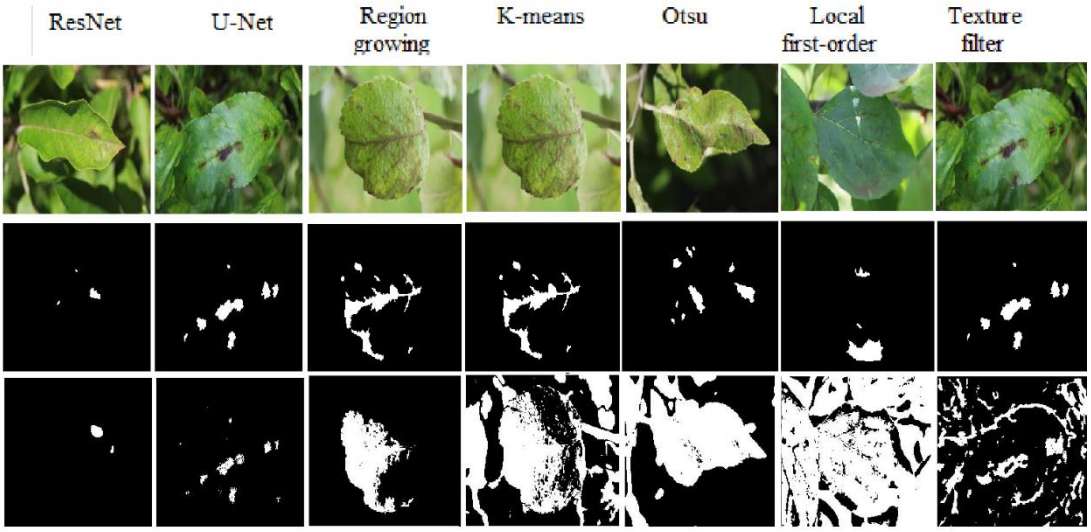

Figure 4. 3 Scab leaf images (first row), true label images (second row), predicted mask of each method (last row).

- Segmentation accuracy of multiple diseases in a single leaf

Table 4. 4 Represents the average of proposed networks compared with other segmentation methods tested on 14 test samples of multiple diseases.

Methods	Metrics			
	Pixel Acc. %	Recall %	IoU %	F1-Score %
ResNet	98.08	54.26	53.25	60.29
U-Net	98.03	59.50	57.40	60.94
Region Growing	86.09	61.97	12.62	28.04
K-means	64	49.88	2.85	9.20
Otsu thresholding	57.81	53.91	1.89	5.91
Local first-order statistics	28.78	49.82	1.42	10.20
Texture filter	72.62	63.91	4.50	9.43

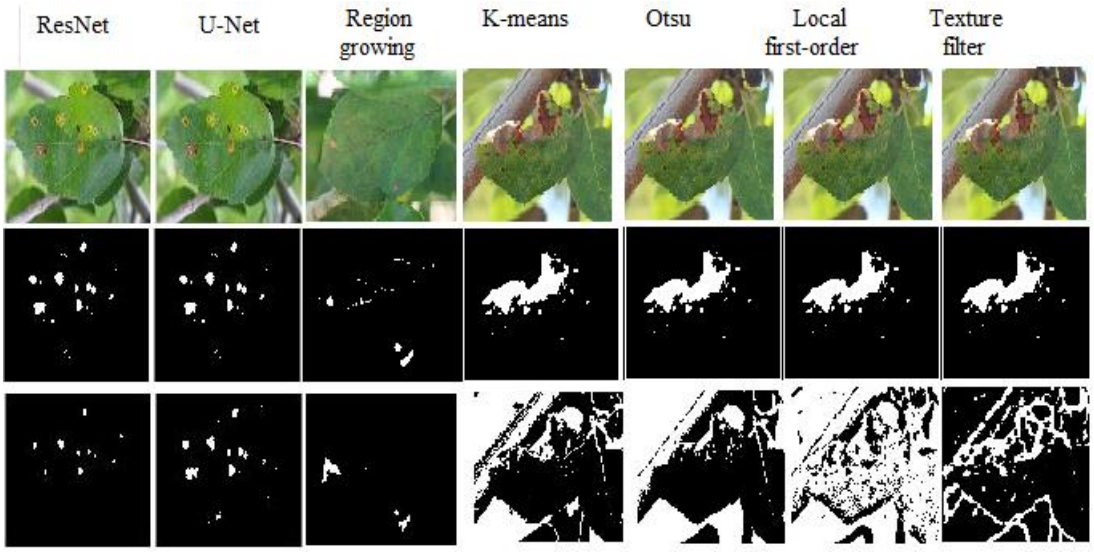

Figure 4. 4 Images of multiple diseases in single leaf (first row), true label images (second row), predicted mask of each method (last row).

Furthermore, to illustrate the results perfectly, the average obtained results for each class have calculated, as outlined in Table 4.5 which is a significant indicator for better interpret the results.

Table 4. 5 Illustrates the average obtained results of four classes for the proposed networks compared with the other segmentation methods.

Metrics	**Segmentation Method**						
	ResNet	U-Net	Region growing	K-means	Otsu Threshold	Local first-order	Texture filter
Pixel Acc.	98.14	96.82	89.95	50.92	56.91	32.99	68.44
Recall	72.68	76.78	70.35	58.99	55.14	73.34	53.06
IoU	70.42	70.89	31.13	8.22	6.53	9.34	5.46
F1-Score	71.66	66.62	39.09	10.38	10.8	9.54	11.8

In line with previous studies, the proposed method achieved distinguished results when compared with feature-based plant disease identification methods. As can be seen in the Table 4.6 which illustrates the comparison performance of those studies that meet with our proposal in extracting complex features from the images of plant leaf disease.

Table 4. 6 Performance comparison

Methods	**Evaluation metrics**				
	Pixel Acc.	IoU	F1-Score	Recall	Precision
Powdery Mildew on cucumber leaf features extraction by using U-Net [31].	96	72.11	83.45	97.34	73.30
A fully automatic segmentation algorithm for medicinal plant leaf images [32].	N/A	87.54	92.59	94.59	92.35
ResNet as FS-SubNet for apple leaf disease [57].	N/A	69	N/A	N/A	N/A
Proposed network (ResNet)	98.14	70.42	71.66	72.68	N/A
Proposed network (U-Net)	96.82	70.89	66.62	76.78	N/A

4.2 Discussion

The main concern of the study was to tackle the problem of segmenting apple foliar diseases from the leaf itself accurately based on high quality, real-life RGB images which includes a high complexity in the symptoms of the disease. To address this problem, we proposed two convolutional neural networks, U-Net and ResNet which

have used for semantic segmentation task in the field of computer vision. To our knowledge, this is the first study to deal with the data used in this experiment, however, the analysis of the result tables and the investigation of the above-illustrated figures indicate that our proposed networks outperform the other segmentation methods in prediction of disease regions almost in all cases, although the high variations in symptoms in each infected leaf, and this is evident in terms of Pixel accuracy, IoU accuracy, F1-Score accuracy, and Recall when validated in 14 test images of each class.

In addition, it has been found that a good match between ResNet and U-Net precision; the only difference between them is that ResNet is relatively good in the balance between precision and recall, and this is can be noticed in the results of F1-Score in healthy leaf case which highlighted in the Table 4.1, as well as, in other cases that include more complex symptoms such as cedar rust, scab and multiple diseases, both networks have proven their ability in recognizing disease location as outlined in the Tables (4.2, 4.3, and 4.4).

Further, some non-disease areas had been misidentified as disease areas, this situation is agreeable, because, in disease detection, the disease regions are not supposed to be missed by the system.

In addition, compared to some studies related to features extraction of plant leaf disease, the proposed networks could relieve researchers from using manually segmenting of the complex features in the image and using a complicated analytical procedure.

Furthermore, compared with some existing methods based on deep learning for feature extraction, a similar pattern of results was obtained in [31,32] and [57] which demonstrated in the Table 4.6, where the proposed networks showed a remarkable performance to segment the small regions of disease spot in all cases, although the complexity of the images used which have a high variations in lighting, age of the leaf, capturing angle and complex backgrounds, while in the mentioned studies they used their own data set to validate their models, in addition to the several image enhancement techniques were used to improve the results.

CHAPTER 5

CONCLUSION

In this research, we have addressed the issue of extracting high dimensionality disease features by presenting two convolutional neural networks which have proven a remarkable segmentation performance in almost all disease cases. However, the motivation of this study was related to the fact that foliar plant diseases symptoms exist only in the leaf area, and the background has no useful information concerning leaf diseases.

Besides, it is possible to conclude that both networks perform well in segmenting disease areas, but we emphasize on using ResNet given its ability to balance between precision and recall.

However, as with the majority of studies, the design of the current study is subject to the limitation of manually labelling process of disease region of the ground truth images which not undergo to a pathologist confirmation this may make our results subordinated to error.

The findings suggest that this approach could also be useful for building a detection system by combining ResNet as a feature extractor subnetwork to extract difference diseases spots from the leaf with another subnetwork which will be trained in an end-to-end manner on those features to distinguish the disease categories.

Further study of the issue is still required to prove the performance of the proposed networks including, providing more samples for each case to train and validate the networks and providing of high GPU memory that undoubtedly extends more space for trying a large size of images and batch sizes during training process which leads to introduce more accurate features.

REFERENCES

[1] Maroua Nouri, Nathalie Gorretta, Pierre Vaysse, Michel Giraud, Christian Germain, Barna Keresztes, Jean-Michel Roger, Near infrared hyperspectral dataset of healthy and infected apple tree leaves images for the early detection of apple scab disease, Data in Brief, Volume 16, 2018, Pages 967-971, ISSN 2352-3409, https://doi.org/10.1016/j.dib.2017.12.043.

[2] A. Krizhevsky, I. Sutskever and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in Advances in Neural Information Processing Systems, pp. 1097-1105, 2012.

[3] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-scale Image Recognition," International Conference on Learning Representations, pp. 196- 201, 2015.

[4] R. Girshick, "Fast R-CNN," in Proceedings of the IEEE International Conference on Computer Vision, pp. 1440- 1448, 2015.

[5] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 580-587, 2014.

[6] C. Szegedy, A. Toshev and D. Erhan, "Deep Neural Networks for Object Detection," in Advances in Neural Information Processing Systems, pp. 2553-2561, 2013.

[7] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed Representations of Words and Phrases and Their Compositionality," in Advances in Neural Information Processing Systems, pp. 3111-3119, 2013.

[8] A. Kumar, O. Irsoy, J. Su, J. Bradbury, R. English, B. Pierce, P. Ondruska, I. Gulrajani, and R. Socher, "Ask me anything: Dynamic Memory Networks for Natural Language Processing," Computing Research Repository, abs/1506.07285, 2015.

[9] A. Bordes, X. Glorot, J. Weston, and Y. Bengio, "Joint Learning of Words and Meaning Representations for Open-Text Semantic Parsing," in Artificial Intelligence and Statistics, pp. 127-135, 2012.

[10] M. Luong, I. Sutskever, Q. V. Le, O. Vinyals, and W. Zaremba, "Addressing the rare word problem in neural machine translation," in Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, pp. 11-19, 2015.

[11] R. Sennrich, B. Haddow and A. Birch, "Neural Machine Translation of Rare Words with Subword Units," Annual Meeting of the Association for Computational Linguistics, pp. 210-221, 2016.

[12] M. Luong, H. Pham and C. D. Manning, "Effective Approaches to Attention-based Neural Machine Translation,", Empirical Methods in Natural Language Processing, pp. 109-114, 2015.

[13] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Banjouke, P. Nguyen, and T. N. Sainath, "Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups," IEEE Signal Processing Magazine, vol. 29, pp. 82-97, 2012.

[14] A. Graves, A. Mohamed and G. Hinton, "Speech Recognition with Deep Recurrent Neural Networks," IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), pp. 6645-6649, 2013.

[15] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative Adversarial Nets," in Advances in neural Information Processing Systems, pp. 2672-2680, 2014.

[16] A. Radford, L. Metz and S. Chintala, "Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks," International Conference on Learning Representations, pp. 134-145, 2015.

[17] J. Long, E. Shelhamer and T. Darrell, "Fully Convolutional Networks for Semantic Segmentation," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3431-3440, 2015.

[18] L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, " DeepLab: Semantic Image Segmentation with Convolutional Nets, Atrous Convolution, and Fully Connected CRFs," in International Conference on Learning Representations, pp. 431-440, 2014.

[19] Dennis H. Greer, Photosynthetic responses to CO2 at different leaf temperatures in leaves of apple trees (Malus domestica) grown in orchard conditions with different levels of soil nitrogen, Environmental and Experimental Botany, Volume 155, 2018, Pages 56-65, ISSN 0098-8472, https://doi.org/10.1016/j.envexpbot.2018.06.014.

[20] Helena Šircelj, Michael Tausz, Dieter Grill, Franc Batič, Biochemical responses in leaves of two apple tree cultivars subjected to progressing drought, Journal of Plant Physiology, Volume 162, Issue 12, 2005, Pages 1308-1318, ISSN 0176-1617, https://doi.org/10.1016/j.jplph.2005.01.018.

[21] Xujun Ye, Shiori Abe, Shuhuai Zhang, Hiroyuki Yoshimura, Rapid and non-destructive assessment of nutritional status in apple trees using a new smartphone-based wireless crop scanner system, Computers and Electronics in Agriculture, Volume 173, 2020, 105417, ISSN 0168-1699, https://doi.org/10.1016/j.compag.2020.105417.

[22] Cailing Guo, Gang Liu, Weijie Zhang, Juan Feng, Apple tree canopy leaf spatial location automated extraction based on point cloud data, Computers and Electronics in Agriculture, Volume 166, 2019, 104975, ISSN 0168-1699, https://doi.org/10.1016/j.compag.2019.104975.

[23] Ziqiang Liu, Guodong Jia, Xinxiao Yu, Water uptake and WUE of Apple tree-Corn Agroforestry in the Loess hilly region of China, Agricultural Water Management, Volume 234, 2020, 106138, ISSN 0378-3774, https://doi.org/10.1016/j.agwat.2020.106138.

[24] Martin Mészáros, Hana Bělíková, Patrik Čonka, Jan Náměstek, Effect of hail nets and fertilization management on the nutritional status, growth and production of apple trees, Scientia Horticulturae, Volume 255, 2019, Pages 134-144, ISSN 0304-4238, https://doi.org/10.1016/j.scienta.2019.04.079.

[25] Ying Duan, Mengxia Zhang, Jin Gao, Pengmin Li, Vasilij Goltsev, Fengwang Ma, Thermotolerance of apple tree leaves probed by chlorophyll a fluorescence and modulated 820nm reflection during seasonal shift, Journal of Photochemistry and Photobiology B: Biology, Volume 152, Part B, 2015, Pages 347-356, ISSN 1011-1344, https://doi.org/10.1016/j.jphotobiol.2015.08.010.

[26] Takahiko Furuya, Ryutarou Ohbuchi, Transcoding across 3D Shape Representations for Unsupervised Learning of 3D Shape Feature, Pattern Recognition Letters, 2020, ISSN 0167-8655, https://doi.org/10.1016/j.patrec.2020.07.012..

[27] Amin Shahraki, Amir Taherkordi, Øystein Haugen, Frank Eliassen, Clustering Objectives in Wireless Sensor Networks: A Survey and Research Direction Analysis, Computer Networks, 2020, 107376, ISSN 1389-1286, https://doi.org/10.1016/j.comnet.2020.107376.

[28] Ming-Hao Du, Xiu-Ying Zheng, Xiang-Jian Kong, La-Sheng Long, Lan-Sun Zheng, Synthetic Protocol for Assembling Giant Heterometallic Hydroxide Clusters from Building Blocks: Rational Design and Efficient Synthesis, Matter, 2020, ISSN 2590-2385, https://doi.org/10.1016/j.matt.2020.06.020.

[29] Zheng, Qinghe, et al. "A Full Stage Data Augmentation Method in Deep Convolutional Neural Network for Natural Image Classification." Discrete Dynamics in Nature and Society, vol. 2020, 2020, pp. 1–11., doi:10.1155/2020/4706576.

[30] Thapa, Ranjita, et al. "The Plant Pathology 2020 Challenge Dataset to Classify Foliar Disease of Apples." ArXiv.org, 24 Apr. 2020, arxiv.org/abs/2004.11958.

[31] Lin, Ke, et al. "Deep Learning-Based Segmentation and Quantification of Cucumber Powdery Mildew Using Convolutional Neural Network." Frontiers in Plant Science, vol. 10, 2019, doi:10.3389/fpls.2019.00155.

[32] Gao, Liwen, and Xiaohua Lin. "Fully Automatic Segmentation Method for Medicinal Plant Leaf Images in Complex Background." Computers and Electronics in Agriculture, vol. 164, 2019, p. 104924., doi:10.1016/j.compag.2019.104924.

[33] Camargo, A., and J.s. Smith. "An Image-Processing Based Algorithm to Automatically Identify Plant Disease Visual Symptoms." Biosystems Engineering, vol. 102, no. 1, 2009, pp. 9–21., doi:10.1016/j.biosystemseng.2008.09.030.

[34] Wang, Haiguang, et al. "Image Recognition of Plant Diseases Based on Principal Component Analysis and Neural Networks." 2012 8th International Conference on Natural Computation, 2012, doi:10.1109/icnc.2012.6234701.

[35] Kaiyu Zhang, Jinglong Chen, Tianci Zhang, Zitong Zhou, A Compact Convolutional Neural Network Augmented with Multiscale Feature Extraction of

Acquired Monitoring Data for Mechanical Intelligent Fault Diagnosis, Journal of Manufacturing Systems, Volume 55, 2020, Pages 273-284, ISSN 0278-6125, https://doi.org/10.1016/j.jmsy.2020.04.016.

[36] Jialin Tang, Qinglang Su, Binghua Su, Simon Fong, Wei Cao, Xueyuan Gong, Parallel ensemble learning of convolutional neural networks and local binary patterns for face recognition, Computer Methods and Programs in Biomedicine, Volume 197, 2020, 105622, ISSN 0169-2607, https://doi.org/10.1016/j.cmpb.2020.105622.

[37] M.V. Valueva, N.N. Nagornov, P.A. Lyakhov, G.V. Valuev, N.I. Chervyakov, Application of the residue number system to reduce hardware costs of the convolutional neural network implementation, Mathematics and Computers in Simulation, Volume 177, 2020, Pages 232-243, ISSN 0378-4754, https://doi.org/10.1016/j.matcom.2020.04.031.

[38] Hui Yin, Yuanhao Gong, Guoping Qiu, Fast and Efficient Implementation of Image Filtering using a Side Window Convolutional Neural Network, Signal Processing, 2020, 107717, ISSN 0165-1684, https://doi.org/10.1016/j.sigpro.2020.107717.

[39] Marcus Stoffel, Franz Bamer, Bernd Markert, Deep convolutional neural networks in structural dynamics under consideration of viscoplastic material behaviour, Mechanics Research Communications, 2020, 103565, ISSN 0093-6413, https://doi.org/10.1016/j.mechrescom.2020.103565.

[40] Xiang Li, Wei Zhang, Hui Ma, Zhong Luo, Xu Li, Partial transfer learning in machinery cross-domain fault diagnostics using class-weighted adversarial networks, Neural Networks, Volume 129, 2020, Pages 313-322, ISSN 0893-6080, https://doi.org/10.1016/j.neunet.2020.06.014.

[41] Song LI, Geoffrey K.F. TSO, Kaijian HE, Bottleneck feature supervised U-Net for pixel-wise liver and tumor segmentation, Expert Systems with Applications, Volume 145, 2020, 113131, ISSN 0957-4174, https://doi.org/10.1016/j.eswa.2019.113131.

[42] Shijie Hao, Yuan Zhou, Yanrong Guo, A Brief Survey on Semantic Segmentation with Deep Learning, Neurocomputing, Volume 406, 2020, Pages 302-321, ISSN 0925-2312, https://doi.org/10.1016/j.neucom.2019.11.118.

[43] Imen Chakroun, Tom Haber, Thomas J. Ashby, SW-SGD: The Sliding Window Stochastic Gradient Descent Algorithm, Procedia Computer Science, Volume 108, 2017, Pages 2318-2322, ISSN 1877-0509, https://doi.org/10.1016/j.procs.2017.05.082.

[44] Tongming Qu, Y.T. Feng, Min Wang, Shengqiang Jiang. (2020) Calibration of parallel bond parameters in bonded particle models via physics-informed adaptive moment optimisation, Powder Technology, Volume 366, Pages 527-536, ISSN 0032-5910, https://doi.org/10.1016/j.powtec.2020.02.077.

[45] He, Q., Ma, B., Qu, D., Zhang, Q., Hou, X., & Zhao, J. (2013). Cotton Pests and Diseases Detection Based on Image Processing. *TELKOMNIKA Indonesian Journal Of Electrical Engineering*, *11*(6). https://doi.org/10.11591/telkomnika.v11i6.2721.

[46] Asfarian, A., Herdiyeni, Y., Rauf, A., & Mutaqin, K. (2013). Paddy diseases identification with texture analysis using fractal descriptors based on fourier spectrum. *2013 International Conference On Computer, Control, Informatics And Its Applications (IC3INA)*. https://doi.org/10.1109/ic3ina.2013.6819152.

[47] P. Narvekar, et al. (2014). Grape Leaf Disease Detection, Classification and Analysis by Using Spatial Gray-Level Dependence Matrices. *International Journal of Advance Engineering and Research Development*, vol. 1, no. 09. https://doi:10.21090/ijaerd.01099.

[48] Molina, J., Gil, R., Bojaca, C., Gomez, F., & Franco, H. (2014). Automatic detection of early blight infection on tomato crops using a color based classification strategy. *2014 XIX Symposium On Image, Signal Processing And Artificial Vision*. https://doi.org/10.1109/stsiva.2014.7010166.

[49] Ros, F., Guillaume, S., Hajji, M., & Riad, R. (2020). KdMutual: A novel clustering algorithm combining mutual neighboring and hierarchical approaches using a new selection criterion. *Knowledge-Based Systems*, *204*, 106220. https://doi.org/10.1016/j.knosys.2020.106220.

[50] Qin Xu, Qiang Zhang, Jinpei Liu, Bin Luo, Efficient synthetical clustering validity indexes for hierarchical clustering, Expert Systems with Applications, Volume 151, 2020, 113367, ISSN 0957-4174, https://doi.org/10.1016/j.eswa.2020.113367.

[51] Guangpeng Wang, Xueyan Liu, Qingmin Yang, Symptom clusters and quality of life in Chinese patients with heart failure, Collegian, 2020, ISSN 1322-7696, https://doi.org/10.1016/j.colegn.2019.12.001.

[52] Maamar Ali Saud ALTobi, Geraint Bevan, Peter Wallace, David Harrison, K.P. Ramachandran, Fault diagnosis of a centrifugal pump using MLP-GABP and SVM with CWT, Engineering Science and Technology, an International Journal, Volume 22, Issue 3, 2019, Pages 854-861, ISSN 2215-0986, https://doi.org/10.1016/j.jestch.2019.01.005.

[53] Fei Wang, Quan Wang, Feiping Nie, Zhongheng Li, Weizhong Yu, Fuji Ren, A linear multivariate binary decision tree classifier based on K-means splitting, Pattern Recognition, Volume 107, 2020, 107521, ISSN 0031-3203, https://doi.org/10.1016/j.patcog.2020.107521.

[54] Jin, P., Lu, L., Tang, Y., & Karniadakis, G. (2020). Quantifying the generalization error in deep learning in terms of data distribution and neural network smoothness. *Neural Networks*, *130*, 85-99. https://doi.org/10.1016/j.neunet.2020.06.024.

[55] Kim, H., Park, J., Kim, D., & Lee, J. (2020). Multilabel naïve Bayes classification considering label dependence. *Pattern Recognition Letters*, *136*, 279-285. https://doi.org/10.1016/j.patrec.2020.06.021.

[56] Torabi Rad, M., Viardin, A., Schmitz, G., & Apel, M. (2020). Theory-training deep neural networks for an alloy solidification benchmark problem. *Computational Materials Science*, *180*, 109687. https://doi.org/10.1016/j.commatsci.2020.109687.

[57] Yu, H., & Son, C. (2020). Leaf Spot Attention Network for Apple Leaf Disease Identification. *2020 IEEE/CVF Conference On Computer Vision And Pattern Recognition Workshops (CVPRW)*. https://doi.org/10.1109/cvprw50498.2020.00034.

[58] Sena Jr, D., Pinto, F., Queiroz, D., & Viana, P. (2003). Fall Armyworm Damaged Maize Plant Identification using Digital Images. *Biosystems Engineering*, *85*(4), 449-454. https://doi.org/10.1016/s1537-5110(03)00098-9.

[59] Weizheng, S., Yachun, W., Zhanliang, C., & Hongda, W. (2008). Grading Method of Leaf Spot Disease Based on Image Processing. *2008 International Conference On Computer Science And Software Engineering*. https://doi.org/10.1109/csse.2008.1649.

[60] Husin, Z., Shakaff, A., Aziz, A., & Farook, R. (2012). Feasibility Study on Plant Chili Disease Detection Using Image Processing Techniques. *2012 Third International Conference On Intelligent Systems Modelling And Simulation*. https://doi.org/10.1109/isms.2012.33.

[61] ZHIHUA, D., HUAN, W., YINMAO, S., & YUNPENG, W. (2013). Jatit.org. Retrieved 27 August 2020, from http://www.jatit.org/volumes/Vol48No1/83Vol48No1.pdf.

[62] Hitimana, E., & Gwun, O. (2014). Automatic Estimation of Live Coffee Leaf Infection Based on Image Processing Techniques. *Computer Science & Information Technology (CS & IT)*. https://doi.org/10.5121/csit.2014.4221.

[63] Kai, S., Zhikun, L., Hang, S., & Chunhong, G. (2011). A Research of Maize Disease Image Recognition of Corn Based on BP Networks. *2011 Third International Conference On Measuring Technology And Mechatronics Automation*. https://doi.org/10.1109/icmtma.2011.66.

[64] Guru, D. S., Mallikarjuna, P. B., & Manjunath, S. (2011). Segmentation and classification of tobacco seedling diseases. *Proceedings of the Fourth Annual ACM Bangalore Conference on - COMPUTE 11*, 1-5. https://doi:10.1145/1980422.1980454.

[65] Mondal, D., Chakraborty, A., Kole, D. K., & Majumder, D. D. (2015). Detection and classification technique of Yellow Vein Mosaic Virus disease in okra leaf images using leaf vein extraction and Naive Bayesian classifier. *2015 International*

Conference on Soft Computing Techniques and Implementations (ICSCTI). https://doi:10.1109/icscti.2015.7489626.

[66]

[67] Revathi P, Hemalatha M (2014) Cotton leaf spot diseases detection utilizing feature selection with skew divergence method. International Journal of scientific engineering and technology 3(1):22–30.

[68] Zhang W, Guifa T, Chunshan W (2013) Identification of jujube trees diseases using neural network. Optik International Journal for Light and Electron Optics 124(11):1034–1037.

[69] Keskar PV, Masare SM, Kadam MS, Deoghare SM (2013) Leaf disease detection and diagnosis. Int J Emerg Trends Electr Electron 2(2):28–31.

[70] Mondal D, Kole DK (2015) Detection and classification technique of yellow vein mosaic virus disease in okra leaf images using leaf vein extraction and Naive Bayesian classifier. In: IEEE international conference on soft computing techniques and implementations (ICSCTI), Faridabad, 8-10 October. pp. 166–171. https://doi:10.1109/ICSCTI.2015.7489626.

[71] Al-Tarawneh MS (2013) An empirical investigation of olive leaf spot disease using auto-cropping segmentation and fuzzy c-means classification. World Appl Sci J 23(9):1207–1211. https://doi:10.5829/idosi.wasj.2013.23.09.1000.

[72] Sekulska-Nalewajko J, Goclawski J (2011) A semi-automatic method for the discrimination of diseased regions in detached leaf images using fuzzy c-means clustering. In: Seventh IEEE international conference on perspective technologies and methods in MEMS design (MEMSTECH), Polyana, 11-14 May. pp. 172– 175.

[73] Alif, M. A., Ahmed, S., & Hasan, M. A. (2017). Isolated Bangla handwritten character recognition with convolutional neural network. *2017 20th International*

Conference of Computer and Information Technology (ICCIT). https://doi:10.1109/iccitechn.2017.8281823.

[74] Machine Learning in MATLAB. (2020). Retrieved from https://www.mathworks.com/help/stats/machine-learning-in-matlab.html.