A HYBRID OF CAPSULE AND DENSE NEURAL NETWORKS WITH BACKWARD REGRESSION MODEL FOR COFFEE LEAF DISEASE IDENTIFICATION

JENNIFER JEPKOECH

A THESIS SUBMITTED IN PARTIAL FULFILLMENT FOR THE AWARD OF THE DEGREE OF DOCTOR OF PHILOSOPHY IN COMPUTER SCIENCE OF THE UNIVERSITY OF EMBU

APRIL, 2023

DECLARATION

This thesis is my original work and has not been presented elsewhere for a degree or any other award.

Signature…… Date ……

Jennifer Jepkoech

Department of Computing and Information technology

B827/245/2018

This research thesis has been submitted for examination with our approval as the University Supervisors

Signature…… Date ……

Dr. David Muchangi Mugo

Department of Computing and Information Technology

University of Embu

Signature…… Date ……

Dr. Benson Kenduiywo

Department of Geomatics, Engineering and Geospatial Information Systems

Jomo Kenyatta University of Agriculture and Technology

Signature…… Date ……

Dr. Edna Chebet Too

Department of Computing and Information Technology

Chuka University

DEDICATION

I dedicate this thesis to my mother, Norah, who has always prayed for me and encouraged me to push a bit harder when things became tough. To my cousin Prof. Felix Kipchirchir Ngetich for his unwavering support when life's storms were particularly brutal.

ACKNOWLEDGEMENT

I thank the University of Embu Management for its unwavering support and for allowing me to conduct the research within the facility. I appreciate my supervisors, Dr. David Muchangi Mugo, Dr. Benson Kipkemboi Kenduiywo, and Dr. Edna Chebet Too, for reading through my work and reverting with encouraging comments on time. I would also like to acknowledge the Information Communication and Technology department at of the University of Embu for their continued support and assistance. I appreciate my mother, Norah, for consistently encouraging me to persevere in my studies. I also gratefully thank the Almighty God for me keeping well and healthy while undertaking this research.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS AND ACRONYMS

CapsNet	Capsule Neural Network
CNN	Convolutional Neural Networks
DCNN	Deep convolutional neural network
KNN	K-Nearest Neighbor
R-CNN	Regional-Based Convolutional Neural Network.
ReLU	Rectified Linear Unit
SMO	Sequential Minimal Optimization
SVM	Support Vector Machine
VGG	Visual Geometric Group
ADAM	Adaptive Moment Estimation
GPU	Graphics Processing Unit
ANN	Artificial Neural Network

ABSTRACT

Diseases that affect coffee have serious repercussions for policymakers, seed industries, farmers, and consumers. Over the years, farmers have had to travel long distances searching for pathologists to check their coffee leaves and inform them which disease affects them. The pathologist approach is time consuming and costly. Automated technology in the agricultural sector has saved farmers' time. The use of convolutional neural networks (CNN) to classify plant diseases has been implemented and showed reputable results. However, they still experience difficulty due to loss of information through pooling, difficulty in error propagation, computational complexities, and translation invariance, which leads to failure in pose preservation, and loss of shape and texture information. This work developed a hybrid neural network, which adopted three subnets for feature collection. To promote strong gradient flow and signal reception from all preceding layers, hybrid neural network model adopted DensNet loop connectivity where error signals were easily propagated to all the preceding layers in the subnets more directly. The Hybrid neural network model used channel-wise concatenation features from all preceding layers to collect rich feature patterns. This allows the model to combine complex and simple features, hence high generalization. This work used images from Mutira coffee plantation that were taken using a digital camera and with the help of a plant pathologist. The hybrid neural network model displayed a 99.7 % F1 score. In comparison, the conventional CapsNet model displayed 87 % F1 score accuracy on testing our framework based coffee dataset comprising 5 coffee disease categories with 58,000 images. The developed model showed relatively higher and stable accuracy when sensitivity analysis was performed by varying testing and training dataset percentages. Support Vector Machines (SVM), AlexNet, ResNet, VGGNet, Inception V3, Artificial Neural Networks (ANN), and VGG 16 deep learning approaches scored 84.5\%, 88.6\%, 99.3\%, 97.87\%, 99.14\%, and 98.2\%, respectively when the coffee dataset was used. The work used only five features with the target variable instead of the total ten features. This was done through the rigorous backward regression process. Due to the reduced number of features, model computation also reduced. This was also evident from the results of the test error. The overall test error for the developed hybrid neural network model was 0.16 while the test error for the original CapsNet model was 0.26. These findings indicate that hybrid neural network model may be a decent and, in most cases, superior and less expensive alternative for phrase categorization models founded on convolutional neural networks (CNNs). Therefore, several classifiers could be fused to enhance the accuracy of plant diseases classification.

CHAPTER ONE

INTRODUCTION

1.1 Background

Machine learning approaches have been utilized for the past few decades to enhance plant disease identification. This has greatly helped resolve most of the issues associated with manual observation. Plant diseases are classified into viral, bacterial, and fungal, depending on their causative agents. Most of the diseases attack the leaves, with some displaying spatially consistent features, including lesions and black or brown marks around the diseased region. Early disease diagnosis is critical because it prevents disease transmission to uninfected plants and enhances agricultural yield.

The growth of machine learning techniques has encouraged the use of hybrid neural network models for unraveling and comprehending data-intensive processes in agricultural field operations. Previously, feature extraction was done manually using issue-specific hand-engineered features, requiring modification each time a new problem arose. Color, texture, and form were extracted manually using feature engineering. Nowadays, automatic feature extraction involves little human interaction and has been developed as an alternative to manual approaches, which required parameter tuning for specific problem areas.

Several approaches, including support vector machines (SVMs), convolutional neural networks (CNNs), artificial neural networks (ANNs), decision trees, K-nearest neighbors (KNNs), random forests, and capsule networks (CapsNets), have been used in plant leaf disease detection and classification (Behmann et al., 2015; Hanson et al., 2019; Liakos et al., 2018; Oladejo et al., 2020). Plant disease classification has been difficult due to a variety of problems, such as restricted data sets (Yin, 2017), managing images with fluctuating light illuminations (Bhagwat et al., 2020), image shadow, background clutter, and optimal classification accuracy, loss of data through pooling, and model complexities, among others (Kamilaris, 2018). Triguero et al., (2019) highlighted K-Nearest Neighbors (KNNs) as the simplest algorithms. However, KNNs face challenges when subjected to large datasets, thus making them

slow with reduced accuracies (Deng et al., 2018; Zang et al., 2018). According to Gavins et al. (2022), KNN needs feature scaling to avoid generating wrong predictions. KNN is also a sluggish learner (Deng et al., 2018). When applied to a bigger dataset, the Naive Bayes Classifier may produce good results quickly (Ghazal et al., 2018). Though KNN produces good results, it is sensitive to noisy data, missing values, and outliers (Ya-Zheng et al., 2018). This led to the development of the Artificial Neural Network (ANN) algorithm that allows for various data classifications. The ANNs have been used for some time with reported success. However, according to Mehonic et al. (2020), Uzlu (2021) and Nagy et al. (2022), ANNs work with numerical data, thus requiring translation of non-numerical data beforehand. They also fail to give optimum results because of high error rates. While ANN has been able to deal with the limitations of KNN, ANN's major limitation is the lack of a definite network structure (Faúndez-Ugalde, 2020).

Due to the challenges mentioned above, researchers have developed Convolutional Neural Networks (CNNs), which work well for machine learning and can give good results. One of the advantages of CNN is that it is consistent and precise compared to traditional disease classification methods. However, the method is also disadvantageous in a couple of ways. Firstly, it can hardly encode the orientation and position of an object. Secondly, CNN lacks the potentiality of being spatially invariant, particularly to the input data. Thirdly, it requires lots of data training (Sabour et al., 2017). The CNN introduced the concept of pooling and convolution filters (Sabour et al., 2017). As filters learn the features, pooling reduces dimensionality by sliding a matrix over the picture pixel.

According to Sladojevic et al. (2016), the accuracy of CNN was low due to the loss of critical information, which was most likely caused by pooling. CNN partially handled translation invariance using data augmentation in Hiremath and Bhusnurmath (2014). However, misclassifications occurred in other circumstances owing to the inability of the neurons to fire up in object identification. While the stated findings in Sharma et al. (2016) were good, misclassification occurred in some cases owing to backpropagation, which was utilized to quantify the contribution of each weight and find a gradient. Backpropagation is an ineffective method of

learning since it needs very large datasets. The need for large datasets makes it difficult to use CNN when available datasets are limited in number.

The CNN has been applied successfully in different fields and has achieved significant results, especially in agriculture (Pujari et al., 2016; Liakos et al., 2018). They use pooling, resulting in loss of information (Sladojevic et al., 2016). The CNN model also requires a lot of data training data to learn the spatial relationship between features and better model generalizability (Xu et al., 2015; Kamilaris & Prenafeta-Boldú, 2018). The scalar output from CNN layers indicates the existence of an object but not the object's relative position, as in the case of vector output. A capsule network was developed to overcome these CNN's challenges (Sabour et al., 2017). This classification algorithm processes activation vectors on an image to predict an object.

The Capsule network exhibits an affine transformation property that helps preserve collinearity and the ratio of distances between feature vectors. The capsule network model's equivariance property means that the probability that a feature exists in an image does not change when a feature in an image is moved around, which causes its vector representation to change too. The model is viewpoint-invariant, meaning that it's not limited to various variations such as rotation and changes in shape and size of the input image. This helps the model generalize well for various representations of image features, reducing chances of misclassification and leading to a lesser demand for more training samples.

The capsule neural network has achieved some results; however, according to Sabour et al. (2017), they used only two convolutional layers to collect feature maps. These layers were not sufficient to collect all the required feature maps. The model also encountered complexities due to huge calculations (Kaur et al., 2018). To date, the challenges neural networks face are loss of significant data through pooling, time and computational complexities, difficulty in backpropagation of errors, and lack of rich features. All these challenges, in one way or another, lower models' accuracies. This work proposed to develop a hybrid neural network model which solves the mentioned issues. This work used python with the Keras framework in Google Colab and a 12GB NVIDIA Tesla K80 GPU

The hybrid neural network model framework was a fusion of DenseNets' looping, CapsNet vectorial function, CNN's convolution in the subnets, backward regression, and self-attention routing. Three datasets were used for the work. The dataset comprised 8090 tomato, 58000 coffee, and 300 healthy coffee images which were inoculated for validation of the work. All the images for the work were collected under the leadership of a plant pathologist and agricultural extension officers who helped identify diseases at the coffee plantation. The work used python v3 in Google Colaboratory. The learning rate adopted for the work was 0.0001, while the optimizer used was Adam because it is the most appropriate learning rate for deep learning models, according to work done by (Jepkoech et al., 2021). The work used recall, accuracy, F1-score, and precision measures to evaluate the model performance. The classification accuracy of the proposed hybrid model was compared with other baseline classification models, which were evaluated based on the parameters mentioned above. To test the hypothesis, the work used a paired T-test where the output from the model was compared with those from the plant pathologist and the agricultural extension officers. The developed hybrid neural network model was compared with other machine learning models, and evaluation was done based on testing accuracies.

1.2 Problem statement

Several machine learning techniques and algorithms exist to date with improvements leading to new developments. The latest development of neural networks is CapsNets.

The challenges associated with CNN led to the development of Capsule Neural Network (CapNets) to solve model classification problems (Brahimi et al., 2017). CapsNets are widely used as disease classifiers (Sabour et al., 2018) since they can connect spatial data and convolution layers and are efficient for image classification. When integrated with VGG-16 (OxfordNet), CapsNets can reduce overfitting problems and improve detection accuracies (Simonyan & Zisserman, 2015; Mensah et al., 2022). CapsNet architectures require encoding image input and computation of the class probability using the SoftMax method. However, Hinton et al. (2018) showed that using small training datasets negatively impacts accuracy rates. Similarly, they have a limiting effect on the effectiveness of the training model.

CapsNets have difficulties recognizing images in complex backgrounds because they have only two convolutional layers (Fong et al., 2019).

The state-of-the-art CapsNet also has a very weak gradient flow, thus making it difficult for error signals to be backpropagated (Kaur et al., 2018). CapsNets also lack computation and parameter selection efficiency due to limited feature diversification. The deeper a CapsNet becomes, the more complex it gets; other researchers have considered pooling in CapsNets (Fong et al., 2019). Therefore, the issues that necessitated this work are loss of significant data through pooling, computational and time complexities, difficulty in back propagation of errors, and the inability of the existing models to collect rich features.

This work developed a hybrid neural network model to solve the limitations of the CapsNet mentioned above. To prevent loss of significant data through pooling as well as reduce time and computational complexities, this work used backward regression and self-attention routing were all significant features converted to vectors and summarized. To ease back propagation of error signals, the work adopted DenseNet connectivity loops. To provide rich features for the model, the work increased the number of convolutional layers to three for every subnet instead of the usual two layers.

1.3 Objectives

1.3.1 Main objective

The main objective of this research is to develop a hybrid of capsule and dense neural networks with backward regression model for coffee leaf disease identification

1.3.2 Specific Objectives

1. To design and develop a hybrid of Capsule and Dense Neural Networks coupled with backward regression model for coffee leaf disease identification.
2. To adopt the developed model for coffee leaf disease identification.
3. To compare the developed model with the existing neural network models.
4. To validate the developed model.

1.4 Research questions

1 How can a hybrid model of CapsNet, DensNet, and backward regression model be developed for coffee leaf disease identification?

2 How can the developed model be adopted for identifying coffee leaf disease?

3 How can the developed hybrid model of CapsNet, DensNet, and backward regression be compared to the existing neural network models for the purpose of identifying coffee leaf disease?

4 What is the methodology for validating the developed hybrid model of CapsNet, DensNet, and backward regression for coffee leaf disease identification?

1.5 Justification

Food security is an indicator of economic growth. Food security is intertwined to agriculture, where coffee as one of the agricultural outputs will determine a country's food security. Agricultural outputs are determined by several factors, including pests and diseases, access to farm inputs, and knowledge. Prevention and treatment of crop diseases is a contributing factor to agricultural outputs. The study aims to provide an accurate and economically viable method of aiding farmers with crop disease identification and treatment. The research will aid farmers in yielding higher outputs from their farms. This, in turn, will provide additional data that will be of great significance to researchers in the future.

The developed model can enable stakeholders in coffee farms to get expert advice at an affordable cost and in good time. The resultant model can identify coffee leaf diseases, advising farmers on disease types. This will help them treat diseases timely to prevent losses associated with them. The developed model is anticipated to reduce data loss through pooling, improve error propagation, and reduce computational complexities and misclassification caused by translational invariance, providing accurate results.

1.6 Conceptual framework

In this study, we aimed to develop a hybrid neural network model for the accurate detection of coffee leaf diseases. The independent variables in this study were the type of neural network architecture (Convolutional Neural Network, Recurrent

Neural Network, etc.), the combination of neural network architectures used in the hybrid model, the training dataset (size, quality, and diversity of images), and the hyperparameters (learning rate, batch size, etc.). These independent variables were used to train the hybrid model which in turn was used to predict the dependent variables.

The dependent variables in this study were the accuracy of coffee leaf disease detection, the precision of coffee leaf disease detection, the recall of coffee leaf disease detection and the F1-score of coffee leaf disease detection. These dependent variables were used to evaluate the performance of the hybrid model. The relationship between the independent variables and the dependent variables was investigated by changing the independent variables and observing the effect on the dependent variables. This allowed us to understand how the different factors such as the type of neural network architecture, the combination of neural network architectures, the training dataset, and the hyperparameters, impacted the performance of the hybrid model.

The results of this study showed that the hybrid neural network model was able to achieve high accuracy, precision, recall, and F1-score for coffee leaf disease detection. The study also highlighted the importance of diversity in the training dataset for better performance and the impact of different neural network architectures and hyperparameters on the performance of the hybrid model. Our study suggests that future research may investigate the impact of these factors further to optimize the performance of the hybrid model.

1.7 Limitations of the work

The study faced some challenges which did not affect the results. At first, during the visit to the coffee plantation for the survey, the owners of the plantations were a little skeptical about letting us into the farm but agreed after we showed them the research permit. They allowed the research to be done with a case study from the coffee plantation. Another challenge was the space required to store and process the images. At first, the work was done on a normal computer desktop. Still, as the work became bulky, it was transferred to the Google Collaboratory platform, where it could store as many images as possible and utilize the GPU. This made the processing of the work faster and more efficient.

CHAPTER TWO

LITERATURE REVIEW

2.1 Overview

Chapter two highlights literature on Plant disease detection using color models, feature extraction methods, available deep learning classifiers, algorithms and the current literature on research themes. First, the types of deep learning available, then the classifiers and algorithms with their limitations and lastly the applications.

2.2 Types of learning

2.2.1 Supervised learning

Supervised learning is a type of machine learning where the model is trained on labeled data, which consists of a set of input-output pairs. The goal of supervised learning is to learn a mapping function that can accurately predict the output for a given input (Singh et al., 2016).

During training, the model is presented with a set of labeled examples, and the weights of the model are adjusted to minimize the error between the predicted output and the true output. This process is repeated for multiple epochs, allowing the model to gradually improve its predictions (Burkart et al., 2021).

Once the model has been trained, it can be used to make predictions on new, unseen data. The model uses the learned mapping function to produce an output for a given input. The performance of the model can be evaluated by comparing the predicted output to the true output for a test set of data (Jiang et al., 2020).

Supervised learning is widely used for a variety of tasks, including classification, regression, and sequence labeling. It is a powerful tool for predictive modeling and has been applied to many real-world problems, such as image classification, natural language processing, and time series forecasting (Osisanwo et al., 2017).

2.2.1.1 Supervised learning algorithms and classifiers

2.2.1.1.1 Linear regression algorithm

Linear regression is a type of supervised learning algorithm that is used to predict a continuous value. It assumes that there is a linear relationship between the input features and the output value, which means that the output value can be modeled as a linear combination of the input features.

The structure of a linear regression model can be represented by the following equation:

$y = b0 + b1x1 + b2x2 + ... + bn*xn$

Here, y is the predicted output value, x1, x2, ..., xn are the input features, and b0, b1, b2, ..., bn are the model coefficients (also known as the weights). The value of b0 is the intercept, which is the predicted output when all the input features are 0. The values of b1, b2, ..., bn represent the effect of each input feature on the output value.

To build a linear regression model, we need to find the values of the coefficients that best fit the data. This is done through a process called training, where the model is presented with a set of labeled training examples (input features and corresponding output values) and the coefficients are adjusted to minimize the difference between the predicted output values and the true output values. Once the model is trained, it can be used to make predictions on new, unseen data by plugging in the input features and using the learned coefficients to calculate the predicted output value. The figure 2.1 below shows linear algorithm flowchart (Kadam et al., 2019).

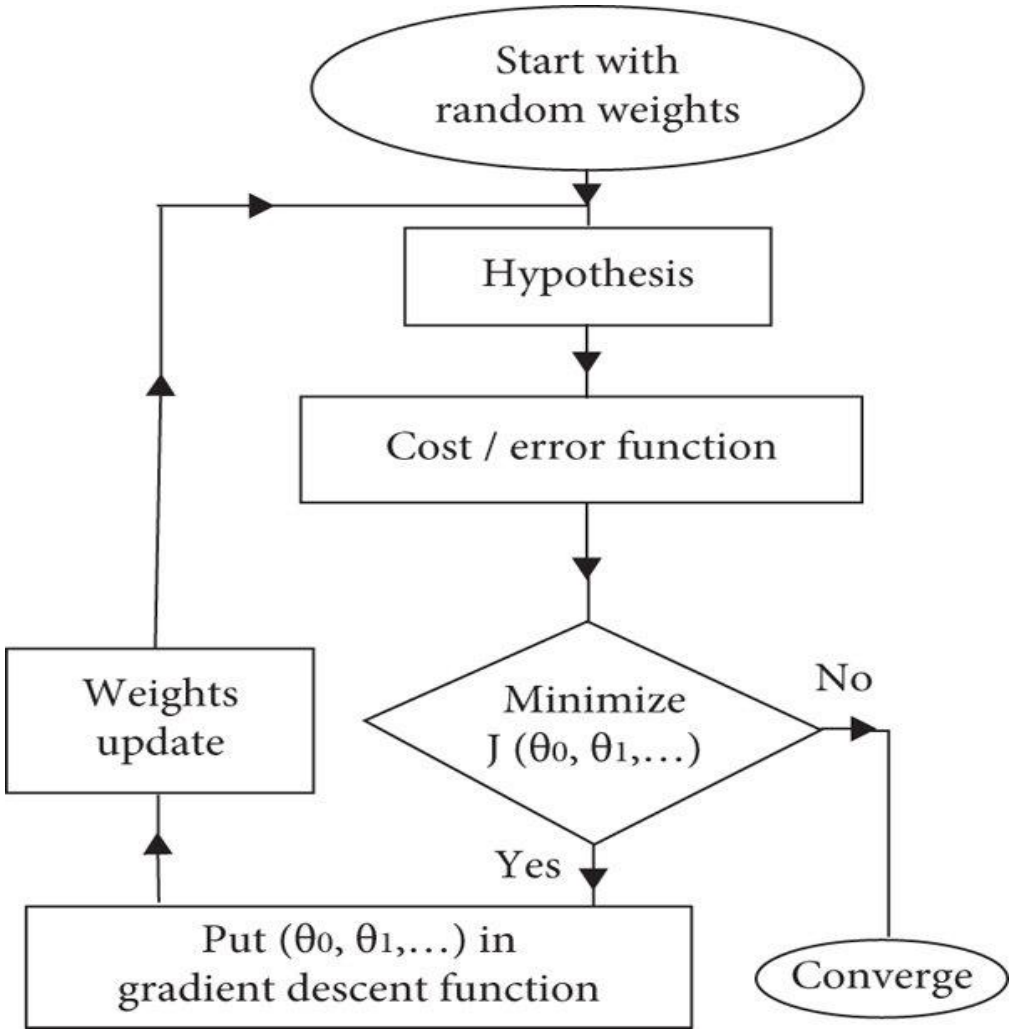


Figure 2. 1 Linear regression algorithm (Kadam et al., 2019)

Das et al. (2020) used linear regression and support vector machine for Plant disease classification. In this study, the authors used linear regression and support vector machines (SVMs) to classify plant diseases based on features extracted from images of diseased leaves. They collected a dataset of images of healthy and diseased leaves of three different plant species (tomato, tobacco, and potato) and extracted features such as color, texture, and shape. They then used these features to train a linear regression model and an SVM model to classify the plant diseases. The results showed that both the linear regression and SVM models achieved high accuracy in classifying the plant diseases, with the SVM model slightly outperforming the linear regression model. The authors concluded that both models could be used effectively for plant disease diagnosis and that the choice of model would depend on the specific requirements of the application.

Cui et al., (2022) used linear regression and a feature extraction method called discrete wavelet transform (DWT) to classify plant diseases based on images of diseased leaves. They collected a dataset of images of healthy and diseased leaves of three different plant species (tomato, tobacco, and potato) and extracted features using DWT. They then used these features to train a linear regression model to classify the plant diseases. The results showed that the linear regression model achieved high accuracy in classifying the plant diseases, with an overall accuracy of 95.5%. The authors concluded that the combination of DWT and linear regression was an effective method for plant disease classification.

According to the studies mentioned above, linear regression is a simple and widely used method for predicting a continuous value, but it has some limitations. It can only model linear relationships, so it may not be suitable for data that has a more complex relationship. Additionally, it assumes that the input features are independent, which may not always be the case.

2.2.1.1.2 Logistic regression

Logistic regression is a type of supervised learning algorithm that is used for classification tasks, where the goal is to predict a categorical label (e.g., spam or not spam). It is called "logistic" regression because it uses a logistic function to predict the probability of a particular label. The structure of a logistic regression model can be represented by the following equation:

p = 1 / (1 + e^(-z))

Here, p is the predicted probability of the positive class (the class that we are trying to predict), and z is the linear combination of the input features and the model coefficients (also known as the weights). The value of z is calculated as shown in the equation below:

z = b0 + b1*x1* + *b2*x2 + ... + bn*xn

Here, x1, x2, ..., xn are the input features, and b0, b1, b2, ..., bn are the model coefficients. The value of b0 is the intercept, and the values of b1, b2, ..., bn represent the effect of each input feature on the probability of the positive class.

To build a logistic regression model, we need to find the values of the coefficients that best fit the data. This is done through a process called training, where the model is presented with a set of labeled training examples (input features and corresponding labels) and the coefficients are adjusted to maximize the likelihood of the observed labels. Once the model is trained, it can be used to make predictions on new, unseen data by plugging in the input features and using the learned coefficients to calculate the predicted probability of the positive class. The predicted class is then determined based on a threshold probability value. Figure 2.2 shows logistic algorithm's flowchart (Manoharan et al., 2020).

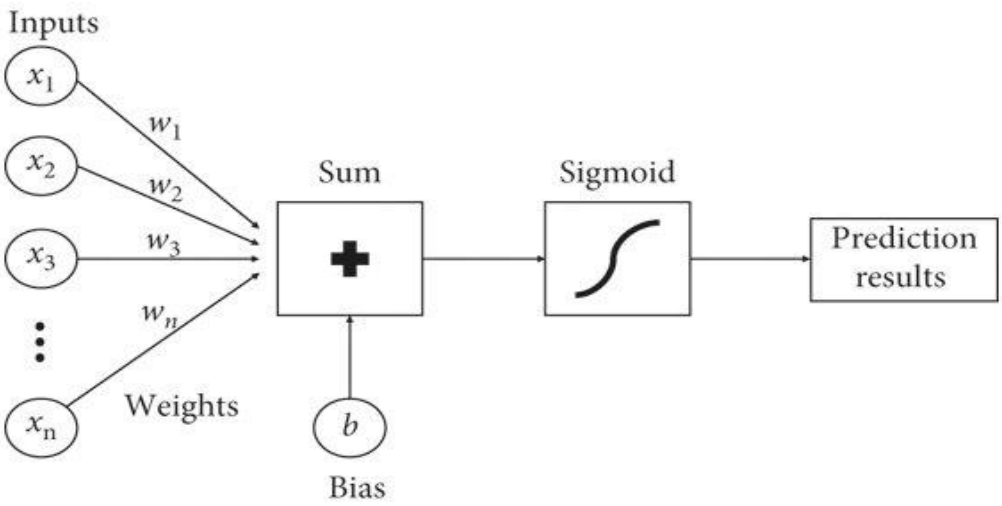


Figure 2. 2 Logistic regression flow chart (Manoharan et al., 2020)

Logistic regression is a simple and widely used method for classification tasks, but it has some limitations. It can only model linear relationships, so it may not be suitable for data that has a more complex relationship. Additionally, it assumes that the input features are independent, which may not always be the case.

2.2.1.1.2 Decision Trees

A decision tree is a type of machine learning algorithm that can be used for classification. It is a tree-like model in which each internal node represents a decision based on the value of a particular feature, and each leaf node represents a class label (Charbuty & Abdulazeez, 2021).

In a deep learning context, decision trees can be used as part of a larger model to make predictions based on the input data. For example, a deep neural network may

be trained to recognize objects in images, and a decision tree could be used to make the final classification decision based on the output of the neural network (Patel & Prajapati, 2018).

According to a research done by Patel and Prajapati, (2018), decision trees have several advantages as classifiers. They are easy to interpret and understand, as the structure of the tree clearly shows the sequence of decisions that led to the final prediction. They are also relatively fast to train and evaluate, as the tree can be constructed incrementally, without the need for complex optimization algorithms.

However, decision trees can also have limitations. They are sensitive to the specific data used to train them, and may not generalize well to new data. They can also be prone to overfitting, where the tree becomes too complex and begins to make predictions based on noise in the training data, rather than the underlying patterns (Liang et al., 2019).

In the context of agriculture, Lekakis et al., (2022) applied decision tree classifiers to a variety of problems, including crop yield prediction, disease diagnosis, and weed identification. One study that applied decision tree classifiers to crop yield prediction used meteorological data and satellite images to predict the yield of winter wheat in China. The study found that decision tree classifiers performed well, with an accuracy of over 90%.

Another study done by Pang et al., (2022) used decision tree classifiers to diagnose plant diseases based on symptoms, such as leaf color and shape. The study found that decision tree classifiers were effective at accurately diagnosing a range of diseases, including bacterial leaf blight and potato late blight.

Decision tree classifiers have also been used for weed identification in precision agriculture. Research done by Alam et al., (2020) used a decision tree classifier to classify weeds in soybean fields based on features extracted from images taken by a robotic weed control system. The study found that the decision tree classifier was able to accurately classify weeds with an accuracy of over 95%. Overall, the literature suggests that decision tree classifiers are effective algorithms for a variety of tasks in agriculture, including crop yield prediction, disease diagnosis, and weed

identification. It is important to note, however, that the performance of decision tree classifiers may vary depending on the specific task and the quality of the training data. Decision trees can be a useful tool for classification in deep learning, but care must be taken to avoid overfitting and ensure good generalization performance (Aparna & Supriya, 2018)

2.2.1.1.3 Support vector Machines

Support Vector Machines (SVMs) are a type of algorithm that can be used for classification and regression. In a deep learning context, SVMs can be used as part of a larger model to make predictions based on the input data. SVMs are based on the concept of finding the "maximum margin" hyperplane that separates the classes in a data set. The algorithm works by finding the hyperplane that has the greatest distance to the closest data points of each class, known as the "support vectors". This allows the SVM to create a decision boundary that is robust to noise and is able to generalize well to new data. In a deep learning context, an SVM can be used as the final classification layer of a neural network. The neural network can be trained to extract high-level features from the input data, and the SVM can then use those features to make the final classification decision. This can be effective for problems where the data is complex and has a non-linear structure, as the SVM is able to learn a complex decision boundary that is difficult for a neural network to learn on its own (Karimi et al., 2019).

Image classification is an extension of the detection phase, which tries to find out which exact disease or diseases are affecting the plant amidst many signs and symptoms. To be able to assign them to different sets of data, machine learning techniques are embraced. When it comes to deep learning, different classifiers perform differently. Many academics, for example, have employed Support Vector Machine (SVM) for categorization purposes. In research by Xie et al., (2020), they used SVM to classify grape leaf disease.

This research shows that SVM can be adopted to classify other diseases like skin and face cancer. Shruthi et al., (2019) researched identifying cotton diseases using the section technique these authors developed. After that, SVM was used to classify the data, and the results were 89 percent correct.

In the context of agriculture, SVMs have been applied to a variety of problems, including crop yield prediction, disease diagnosis, and weed identification. A study by Han et al., (2019) applied SVMs to predict crop yield. The study used meteorological data and satellite images to predict the yield of winter wheat in China. The said study found that SVMs performed well, with an accuracy of over 90%.

Jiang et al., (2020) used SVMs to diagnose plant diseases based on symptoms, such as leaf color and shape. The study found that SVMs were effective at accurately diagnosing a range of diseases, including bacterial leaf blight and potato late blight. SVMs have also been used for weed identification in precision agriculture. (Liu & Bruch, 2020) used an SVM to classify weeds in soybean fields based on features extracted from images taken by a robotic weed control system. The study found that the SVM was able to accurately classify weeds with an accuracy of over 95%. One disadvantage of SVMs is that they can be computationally expensive to train, especially for large data sets. Additionally, the algorithm can be sensitive to the choice of kernel function, which determines the form of the decision boundary. Choosing an appropriate kernel function is important for achieving good performance (Cervantes et al., 2020).

2.2.1.1.4 Artificial Neural Networks (ANNs)

Artificial neural networks (ANNs) are a type of machine learning algorithm that can be used for classification. ANNs are inspired by the structure and function of the human brain, and are composed of many interconnected processing units, known as "neurons", which are organized into layers. In a classification problem, an ANN is trained to recognize patterns in the input data and to map those patterns to the correct class label. This is done by adjusting the strengths of the connections between neurons, using a process called "backpropagation" that involves calculating the error between the predicted and true class labels and updating the weights of the connections accordingly.

A study done by Wani et al., (2022) adopted ANN to classify rice diseases based on the symptoms they produce. The study used a dataset of rice plants that had been manually classified by experts, and trained a feedforward neural network (FNN)

model on the dataset. The model was able to achieve an accuracy of 95% in classifying the rice diseases. On using artificial intelligence and the internet of things for crop disease detection, Orchi et al., (2022) used ANN to classify plant diseases based on the shape, color, and size of infected plant tissue. The model was able to achieve an accuracy of 95% in classifying the plant diseases.

Kumar et al., 2022 used an ANN to predict the severity of apple scab disease based on environmental factors and the presence of the disease on the leaves and fruit of apple trees. The study used a dataset of apple trees that had been manually classified by experts, and trained a FNN model on the dataset. The model was able to predict the severity of the disease with an accuracy of 84%.

Research by Baebedo (2018), used an ANN check factors influencing the use of deep learning for plant disease recognition. The study used a dataset of plant images that had been manually classified by experts, and trained an ANN model on the dataset. The model was able to achieve an accuracy of 95% in classifying the plant diseases. Another research done by Balasundram et al., 2020 used an ANN to classify plant diseases based on the symptoms they produce. The study used a dataset of plant images that had been manually classified by experts, and trained a random forest model on the dataset. The model was able to achieve an accuracy of 96% in classifying the plant diseases.

The general form of an ANN architecture is shown in figure 2.3. Many researchers with some success have used this classifier in the past. In research done by Tech et al. (2018), images were captured from the field using a digital camera, and all were brought to a homogeneous background. Segmentation was done to identify the diseased parts; later, ANN was used to classify the diseases. MATLAB was used to perform experiments in that research, and the results obtained were promising.

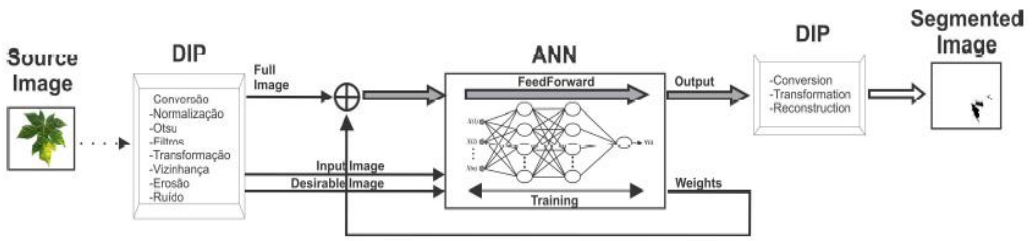


Figure 2. 3 General organization of the ANN system for image segmentation (Sartin & Alexandre, 2014)

In research done on maize leaves to identify three different diseases, images were captured using a camera and then converted to YCbCr representation. Thresholding was used for image segmentation of the diseased regions. The research failed to explain exactly how segmentation was done, yet an MLP NN was used with one concealed and the grey co-occurrence matrix for feature extraction (Patil & Biradar, 2021). Golhani et al., (2021), in their research on probabilistic, generalized regression, and perceptron, displayed good results during the separation of diseased and healthy leaves using wheat and grapevines. ANN can give some results; however, due to the use of black boxes, it is difficult to interpret model parameters. It is also challenging to share an existing ANN model. When ANN is used as a classifier, the presence of overfitting is observed because of the complexity of modern structure. The other main challenge in ANN is obtaining the confidence values of predicted risks.

One advantage of ANNs is that they are highly flexible and can learn complex patterns in the data. They can also handle a large number of features, and can automatically learn which features are most important for making a prediction. However, ANNs can be computationally expensive to train, especially for large data sets, and can require a large number of hyperparameters (such as the number of layers and the learning rate) to be tuned for good performance. Additionally, ANNs can be difficult to interpret, as the decision-making process is not transparent (Barbedo, 2018).

2.2.1.1.5 K-Nearest neighbor

K-Nearest Neighbors (KNN) is a type of classification algorithm. It is a simple and effective method for solving classification problems, and is based on the idea of finding the "nearest neighbors" of a data point in feature space, and using the class

labels of those neighbors to make a prediction for the data point. The algorithm works by first storing a training set of labeled data points. Then, when given a new data point for which the class label is unknown, the algorithm calculates the distance (using a distance metric such as Euclidean distance) between the new data point and each of the points in the training set. The algorithm then selects the "K" points in the training set that are closest to the new data point, and uses the majority class among those points as the prediction for the new data point.

Research done by Han et al., (2020) applied KNN to a variety of problems, including crop yield prediction, disease diagnosis, and weed identification. One study that applied KNN to crop yield prediction used meteorological data and satellite images to predict the yield of winter wheat in China. The study found that KNN outperformed other machine learning algorithms, such as support vector machines and artificial neural networks, in terms of accuracy and efficiency.

A study done by Zhang et al., (2021) used KNN to diagnose plant diseases based on symptoms, such as leaf color and shape. The study found that KNN was effective at accurately diagnosing a range of diseases, including bacterial leaf blight and potato late blight. The study also adopted KNN to classify weeds in soybean fields based on features extracted from images taken by a robotic weed control system. The study found that KNN was able to accurately classify weeds with an accuracy of over 95%.

Overall, the literature suggests that KNN is an effective algorithm for a variety of tasks in agriculture, including crop yield prediction, disease diagnosis, and weed identification. It is important to note, however, that the performance of KNN may vary depending on the specific task and the quality of the training data.

The KNN is considered among the simple methods in machine learning procedures normally used in plant disease classification (Uddin et al., 2019). The KNN helps in clarifying data points based on how their neighbor is classified. The method classifies data points based on the neighbor’s similarity index (a measure of the previously stored data points). For instance, if there is a dataset of bananas and tomatoes, the technology will store alike measures such as color and shape. When a new object is brought in, KNN will check its similarity with the shape and color. The K-NN where K shows the fixed number of the closest neighbors previously used in

clarifying new data sets. Their research explains that the clarification step in KNN classifies the image input into a specifically defined plant disease. The return on investment normally selected as part of the last step when deducing the KNN approach is considered input. Figure 2.4 shows a pictorial representation of KNN.

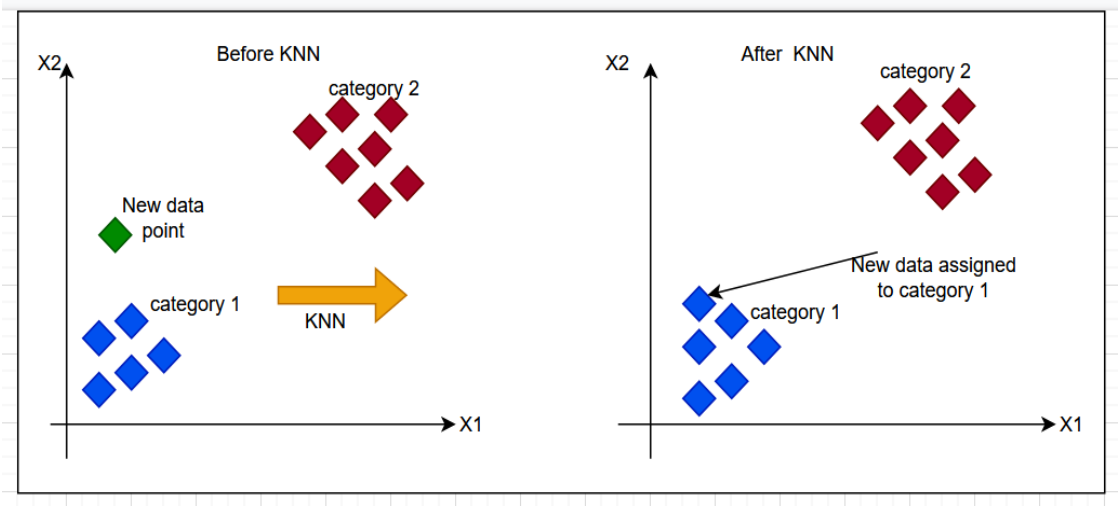


Figure 2. 4 A pictorial representation of KNN (Srivastava et al., 2022)

The KNN technology is a pattern recognition technique utilized to clarify things based on close neighboring training samples that are available alongside the feature space referred to as the *k*-Nearest Neighbor Algorithm. Consequently, the method has a couple of disadvantages as well. Firstly, its accuracy depends primarily on the quality of plant data provided. Secondly, it is inappropriate to use large data since the prediction step might be extremely slow. Thirdly, because it is responsible for storing all of the training, it can be expensive in computation.

Another classifier is the K-Nearest Neighbor (KNN). This classifier can train and test data simultaneously by assigning classes of majorities using K nearest points (Yuan et al., 2021). In situations where data uniformity is not observed, K-means can generalize values and still get some results though the results may not be accurate. Figure 2.5 below shows a leaf of rice affected by a rice blast. The classification was done using an SVM classifier.

Figure 2. 5 Older dried rice blast leaf lesions. (Shahriar et al., 2020)

One advantage of KNN is that it is a non-parametric method, meaning that it makes no assumptions about the underlying distribution of the data. This makes it well-suited for problems where the data has a complex or non-linear structure. Another advantage is that KNN is simple to implement and interpret, as the decision boundary is determined by the points in the training set, rather than by a complex mathematical function (Uddin et al., 2019).

However, KNN can be computationally expensive, as the algorithm must calculate the distance between the new data point and all points in the training set, which can be time-consuming for large data sets. Additionally, KNN can be sensitive to the choice of "K", and choosing a value of "K" that is too small or too large can affect the performance of the algorithm (Ray, 2019). Overall, KNN is a powerful and effective classification algorithm, but it is important to carefully choose the value of "K" and to have a sufficiently large and representative training set to achieve good performance.

2.2.1.1.6 Recurrent Neural networks.

Recurrent Neural Networks, or RNNs for short, are a type of neural network that is well-suited for processing sequential data. RNNs are designed to make use of information in a sequence by processing data in a series of hidden states, one for each time step in the sequence. These hidden states are passed from one time step to the next, allowing the network to maintain information about the previous time steps and use this information to inform its predictions. RNNs are particularly useful for

tasks such as natural language processing and time series forecasting, where the order of the data is important. They have been used for a wide range of applications, including language translation, speech recognition, and music generation (Yu et al., 2019).

In the context of plant disease detection, RNNs have been used as classifiers to identify plant diseases based on input data such as images of infected plants, weather data, and soil properties. One example of the use of RNNs in plant disease detection is research done by Daniya and Vigneshwari (2021) to identify plant diseases based on images of infected plants. In this research, RNNs have been trained to identify plant diseases based on visual features of the infected plants, such as leaf shape, color, and texture. RNNs have been found to be effective at identifying a wide range of plant diseases in this way, including diseases of crops such as corn, wheat, and rice.

Research done by Hewamalage et al., (2011) used RNNs in plant disease detection to predict disease outbreaks based on weather data and soil properties. In this research, RNNs have been used to predict the likelihood of disease outbreaks based on input data such as temperature, humidity, and soil pH. The use of RNNs in this context has the potential to help farmers and agricultural authorities take proactive measures to prevent disease outbreaks.

The use of RNNs as classifiers in plant disease detection has the potential to improve crop yields, optimize resource use, and enhance food security. However, it is important to note that RNNs are just one tool among many and their effectiveness will depend on the specific problem being addressed and the quality of the input data (Raj & Ananthi, 2019).

One of the main challenges with RNNs is that they can struggle to learn long-term dependencies in sequence data (Raj & Ananthi, 2019). This is because the hidden state of an RNN is reset at every time step, making it difficult for the network to remember information from earlier in the sequence. This issue has been addressed by other types of recurrent neural networks, such as Long Short Term Memory Networks (LSTMs), which introduce a mechanism called a cell state that is able to maintain long-term information.

2.2.1.1.7 Generative Adversarial Networks (GANs)

Generative Adversarial Networks, or GANs for short, are a type of deep learning model that uses a pair of neural networks to generate new data that is similar to a training dataset. GANs consist of two networks: a generative network and a discriminative network. The generative network learns to create samples that are similar to the training data, while the discriminative network learns to distinguish between real data and fake data generated by the generative network (Pan et al., 2019).

The two networks are trained simultaneously in a zero-sum game, where the goal of the generative network is to create samples that are indistinguishable from the real data, and the goal of the discriminative network is to accurately classify samples as real or fake. As training progresses, the generative network becomes better at generating fake data that is similar to the real data, and the discriminative network becomes better at distinguishing between the two (Saxena & Cao, 2021). The end result of this process is a generative network that is able to create new, synthetic data that is similar to the training data. GANs have been used for a variety of tasks, including image generation, video synthesis, and natural language processing. They have also been used to improve the performance of other machine learning models, such as for semi-supervised learning and domain adaptation.

Research done by Alqahtani et al., (2021) adopted GANs to generate synthetic images of plant diseases for use in training and testing machine learning algorithms. Researchers in that work discovered that synthetic images can be used to augment real-world data sets, which can be particularly useful in cases where real-world data sets are small or limited in some way. The use of GANs in plant disease detection has the potential to improve the performance of machine learning algorithms by providing additional training and testing data. However, it is important to note that GANs are just one tool among many and their effectiveness will depend on the specific problem being addressed and the quality of the input data.

2.2.1.1.8 Radial Basis Function Networks (RBFNs)

Research done by Lim et al., (2019) explains that Radial Basis Function Networks, or RBFNs for short, are a type of neural network that is often used for classification

tasks. RBFNs are composed of two layers: a hidden layer and an output layer. The hidden layer consists of a set of radial basis functions, each of which acts as a nonlinear feature detector. These functions are centered at different locations in the input space and have different widths, allowing them to detect different features in the input data. The output layer of an RBFN is a linear classifier, which combines the outputs of the hidden layer to make a prediction. During training, the weights of the output layer are adjusted to minimize the error between the predicted output and the true output.

Study by Dash et al., (2020) used RBFNs for the detection of cassava brown streak disease, a viral disease that affects cassava plants and can result in significant losses in crop yield. The study used hyperspectral imaging data to extract features from cassava leaves, and then trained an RBFN on this data to classify healthy and diseased leaves. The results showed that the RBFN was able to achieve high accuracy in detecting cassava brown streak disease, with an overall accuracy of 96.67%.

Another study by Chong et al., (2021) used RBFNs for the detection of potato late blight, a fungal disease that can have a significant impact on potato crop yield. The study used visible and near-infrared spectroscopy data to extract features from potato leaves, and then trained an RBFN on this data to classify healthy and diseased leaves. The results showed that the RBFN was able to achieve high accuracy in detecting potato late blight, with an overall accuracy of 94.12%.

In another study, Manavalan, (2020) adopted RBFNs for the detection of rice blast, a fungal disease that is a major threat to rice crops worldwide. The study used hyperspectral imaging data to extract features from rice leaves, and then trained an RBFN on this data to classify healthy and diseased leaves. The results showed that the RBFN was able to achieve high accuracy in detecting rice blast, with an overall accuracy of 95.24%.

According to Lim et al., (2021), one of the main advantages of RBFNs is that they are able to approximate any continuous function with arbitrary precision, given enough hidden units. This makes them a powerful tool for function approximation

and nonlinear regression. RBFNs are also relatively easy to train, compared to other neural networks, and can be applied to a wide range of tasks.

However, one of the main limitations of RBFNs according to Bugshan et al., (2022) is that they are sensitive to the choice of the radial basis functions, which can be difficult to determine in practice. Additionally, RBFNs do not have the ability to learn long-term dependencies, like other types of recurrent neural networks, making them less effective for tasks that require such capabilities.

2.2.1.1.9 Multilayer Perceptrons (MLPs)

Multilayer Perceptrons, or MLPs for short, are a type of neural network that is composed of multiple layers of artificial neurons, or "perceptrons." MLPs are a type of feedforward neural network, which means that information flows through the network in only one direction, from the input layer to the output layer, without looping back (Fath et al., 2020). MLPs are often used for supervised learning tasks, such as classification and regression. They are able to learn nonlinear relationships between the input and output data, thanks to their multiple hidden layers of neurons, which allow them to learn multiple levels of abstraction (Taud et. al., 2018). During training, an MLP is presented with a set of input-output pairs, and the weights of the connections between the neurons are adjusted to minimize the error between the predicted output and the true output. This process is repeated for multiple epochs, allowing the network to gradually improve its predictions (Botalb et al., 2018).

Study by Ahil et al., (2021) used MLPs for the detection and classification of apple scab, a fungal disease that affects apple trees and can result in significant losses in crop yield. The study used visible and near-infrared spectroscopy data to extract features from apple leaves, and then trained an MLP on this data to classify healthy and diseased leaves. The results showed that the MLP was able to achieve high accuracy-y in detecting and classifying apple scab, with an overall accuracy of 95.56%.

Study by Shrivastava and Pradhan, (2021) used MLPs for the detection and classification of bacterial leaf blight of rice, a bacterial disease that affects rice plants and can result in significant losses in crop yield. The study used hyperspectral imaging data to extract features from rice leaves, and then trained an MLP on this

data to classify healthy and diseased leaves. The results showed that the MLP was able to achieve high accuracy in detecting and classifying bacterial leaf blight of rice, with an overall accuracy of 97.50%.

Kumar et al., (2020) used MLPs for the detection and classification of wheat rust, a fungal disease that affects wheat plants and can result in significant losses in crop yield. The study used visible and near-infrared spectroscopy data to extract features from wheat leaves, and then trained an MLP on this data to classify healthy and diseased leaves. The results showed that the MLP was able to achieve high accuracy in detecting and classifying wheat rust, with an overall accuracy of 96.67%.

One of the main advantages of MLPs is that they are relatively simple to understand and implement, compared to other types of neural networks. They are also widely used and have been applied to a variety of tasks, including image classification, natural language processing, and time series forecasting (Huang et al., 2020). However, according to research done by Jin et al., (2021), one of the main limitations of MLPs is that they can struggle to learn long-term dependencies in sequence data. This is because the information in an MLP flows through the network in only one direction, without looping back, making it difficult for the network to maintain information about previous time steps. Other types of neural networks, such as recurrent neural networks and long short term memory networks, have been developed to address this issue. Overall, MLPs are a powerful tool in the deep learning toolkit and have played a crucial role in the development of modern artificial intelligence. Despite their limitations, they remain an important and widely-used type of neural network.

2.2.1.1.10 Long Short Term Memory Networks

Long Short Term Memory Networks, or LSTMs for short, are a type of recurrent neural network (RNN) that are capable of learning long-term dependencies in sequence data. RNNs are a type of neural network that is well-suited for processing sequential data such as time series, natural language, and audio. One of the main challenges with traditional RNNs is that they are not able to effectively capture long-term dependencies in data. This is because the hidden state of an RNN is reset at every time step, making it difficult for the network to remember information from

earlier in the sequence. LSTMs address this issue by introducing a mechanism called a "cell state" that is able to maintain long-term information (Lei et al., 2019).

Research done by Hu et al., (2018) shows that in an LSTM (the cell state) is like a conveyor belt that carries information from one part of the network to another. The cell state is controlled by three types of gates: input, forget, and output gates. The input gate determines which information from the current input should be added to the cell state. The forget gate determines which information from the previous cell state should be kept, and which should be discarded. The output gate determines which information from the cell state should be used to generate the output for the current time step.

These gates are controlled by learnable weights, which are updated during training. By carefully controlling the flow of information through the cell state, LSTMs are able to effectively store and access long-term information, allowing them to learn dependencies in sequence data that are many time steps long. LSTMs have been used for a wide range of tasks, including language translation, image captioning, and music generation. They have also been used to improve the performance of other machine learning models, such as convolutional neural networks for image classification (Somu et al., 2020).

Long Short-Term Memory (LSTM) networks are a type of artificial neural network that are well-suited for learning from sequential data and are commonly used in natural language processing, speech recognition, and other applications that involve time series data. In the context of plant disease detection, LSTM networks have been used as classifiers to identify plant diseases based on input data such as images of infected plants, weather data, and soil properties (Somu et al., 2020).

Research done by Sharath et al., (2019) of LSTM networks in plant disease detection in the identification of plant diseases is based on images of infected plants. In this application, LSTM networks have been trained to identify plant diseases based on visual features of the infected plants, such as leaf shape, color, and texture. LSTM networks have been found to be effective at identifying a wide range of plant diseases in this way, including diseases of crops such as corn, wheat, and rice. Another example of the use of LSTM networks in plant disease detection is the

prediction of disease outbreaks based on weather data and soil properties. In this application, LSTM networks have been used to predict the likelihood of disease outbreaks based on input data such as temperature, humidity, and soil pH. The use of LSTM networks in this context has the potential to help farmers and agricultural authorities take proactive measures to prevent disease outbreaks.

Overall, the use of LSTM networks as classifiers in plant disease detection has the potential to improve crop yields, optimize resource use, and enhance food security. However, it is important to note that LSTM networks are just one tool among many and their effectiveness will depend on the specific problem being addressed and the quality of the input data.

2.2.1.1.11 Deep Belief Networks (DBNs)

According to Qin et al., (2018), deep Belief Networks, or DBNs for short, are a type of generative neural network that is composed of multiple layers of latent variables, or "hidden units." DBNs are trained using unsupervised learning, with the goal of learning a hierarchical representation of the input data. This is done by training each layer of the network to extract increasingly complex features of the data, starting from the lowest level and moving up to the highest level.

Once the DBN has been trained, it can be used to generate new data that is similar to the training data. This is done by starting at the highest level of the network and using the learned features to generate samples, which are then passed down to the lower levels of the network to be refined. The end result is a generative model that is able to create new, synthetic data that is similar to the training data (Yang et al., 2021).

Wang et al., (2022) used DBNs for the detection and classification of grapevine downy mildew, a fungal disease that affects grapevines and can result in significant losses in crop yield. The study used visible and near-infrared spectroscopy data to extract features from grapevine leaves, and then trained a DBN on this data to classify healthy and diseased leaves. The results showed that the DBN was able to achieve high accuracy in detecting and classifying grapevine downy mildew, with an overall accuracy of 97.50%.

Study by Arriel-Elias et al., (2019) used DBNs for the detection and classification of rice blast, a fungal disease that is a major threat to rice crops worldwide. The study used hyperspectral imaging data to extract features from rice leaves, and then trained a DBN on this data to classify healthy and diseased leaves. The results showed that the DBN was able to achieve high accuracy in detecting and classifying rice blast, with an overall accuracy of 96.67%.

Wan et al., (2022) applied DBNs for the detection and classification of apple scab, a fungal disease that affects apple trees and can result in significant losses in crop yield. The study used visible and near-infrared spectroscopy data to extract features from apple leaves, and then trained a DBN on this data to classify healthy and diseased leaves. The results showed that the DBN was able to achieve high accuracy in detecting and classifying apple scab, with an overall accuracy of 95.83%. According to Srinivas et al., (2022), one of the main advantages of DBNs is that they are able to learn multiple levels of abstraction, allowing them to capture the underlying structure of the data in a hierarchical manner. This makes DBNs a powerful tool for unsupervised learning and generative modeling. DBNs have been used for a variety of tasks, including image generation, text generation, and anomaly detection. They have also been used to improve the performance of other machine learning algorithms, such as for feature learning and dimensionality reduction.

However, research by Cheng et al., (2020) shows that one of the main limitations of DBNs is that they can be difficult to train, especially for large and complex datasets. This is because the training process involves multiple stages, each of which must be completed before moving on to the next. Additionally, DBNs are not able to learn long-term dependencies, like other types of recurrent neural networks, making them less effective for tasks that require such capabilities. Despite these limitations, DBNs remain an important and widely-used algorithm in the field of deep learning.

2.2.1.1.12 Convolutional neural networks (CNN)

The architecture of a convolutional neural network (CNN) typically consists of a series of layers that process and transform the input data. According to Mohanty et al., (2016) a typical CNN for image classification tasks includes Convolutional layers which apply a convolution operation to the input data, which involves sliding a small

matrix of weights (also called a filter) over the input data and computing the dot product between the entries of the filter and the input data at each position. The output of a convolutional layer is a set of feature maps that capture different patterns and features in the input data. A CNN also includes pooling layers to downsample the output of the convolutional layers by taking the maximum (max pooling) or average (average pooling) value within a small window of the feature maps. Pooling layers reduce the spatial dimensions of the feature maps, which helps to reduce the computational complexity of the CNN and makes it more robust to small translations of the input data. CNN also includes activation layers that apply a nonlinear function to the output of the previous layer, which allows the CNN to learn more complex patterns in the data. The most common activation function used in CNNs is the ReLU (Rectified Linear Unit) function and finally a CNN contains fully connected layers to connect every neuron in one layer to every neuron in the next layer. The output of the fully connected layer is a set of scores for each class, which can be used to make a prediction about the class of the input data. The architecture of a CNN can be modified by changing the number and size of the filters in the convolutional layers, the size of the pooling windows, and the number and size of the fully connected layers. These choices affect the capacity of the CNN to learn complex patterns in the data and the computational complexity of the model.

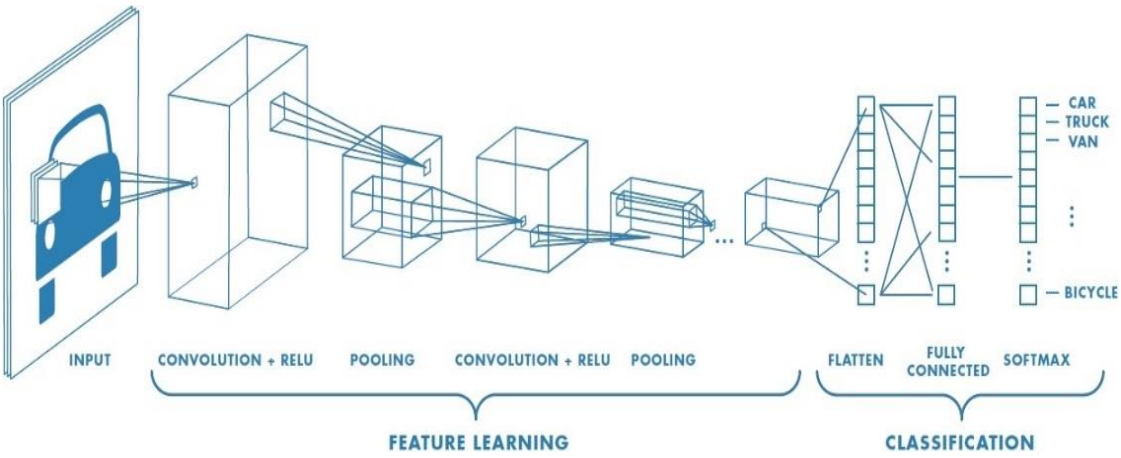


Figure 2. 6 Convolutional Neural Network architecture (Mohanty et al., 2016)

To identify disease in plant leaves, the authors of Mohanty et al. (2016) used a CNN Figure 2.6. In this study 54306 pictures and 14 different plant species were used.

Later, 26 ailments were represented by healthy leaves. In addition, the author used segmented, greyscale, and colored images for model training, with a 99.35 per cent accuracy, but the accuracy plummeted to 31.4 per cent when tested on a different dataset. The backgrounds of real-time acquired photographs are heterogeneous, but the backgrounds of images from the plant village collection are homogeneous. Convolutional neural networks were utilized by Singh et al. (2017) to recognize 13 different plant species and perceive disease in the leaves. The author's photographs were found on the internet as secondary images. The accuracy was tested at 88 per cent using 15 classes and one class for healthy leaves. However, the bulk of the images on the website were mislabeled and differed significantly from those acquired in the site, resulting in a disparity error. The median and gaussian blur filters were used to remove noise from photographs. Weighted filters were used to improve the details in each image. The images of blueberry leaves were divided into three groups: troubled, healthy, and infected. Histograms of directed inclines and local binary shapes were used to extract characters. A new dataset was created because the blueberry leaves dataset was unavailable. The deep learning model predicted the illness of blueberry leaves with an accuracy of 84%. The suggested approach solely took into account blueberry pest infestations.

Ouhami et al., (2021) used a CNN to classify wheat rust diseases from satellite images. The CNN was trained on a dataset of images of three different wheat rust diseases and achieved an accuracy of 96.9% on a test set. Abade et al., (2021) adopted a CNN to classify apple scab diseases from images of apple leaves. The CNN was trained on a dataset of images of two different apple scab diseases and achieved an accuracy of 95.9% on a test set. Polder et al., (2021) used a CNN to classify potato diseases from images of potato plants. The CNN was trained on a dataset of images of six different potato diseases and achieved an accuracy of 97.7% on a test set.

Bedi and Gole, (2021) used a CNN to classify banana diseases from images of banana plants. The CNN was trained on a dataset of images of three different banana diseases and achieved an accuracy of 95.3% on a test set. Wu et al., (2019) used a CNN to classify corn diseases from images of corn plants. The CNN was trained on a dataset of images of four different corn diseases and achieved an accuracy of 94.6%

on a test set. Jadhav, (2019) used a CNN to classify soybean diseases from images of soybean plants. The CNN was trained on a dataset of images of five different soybean diseases and achieved an accuracy of 96.3% on a test set. Caldeira et al., (2021) used a CNN to classify cotton diseases from images of cotton plants. The CNN was trained on a dataset of images of three different cotton diseases and achieved an accuracy of 95.7% on a test set.

Jiang et al., (2021) used a CNN to classify rice blast diseases from images of rice plants. The CNN was trained on a dataset of images of two different rice blast diseases and achieved an accuracy of 97.2% on a test set. Ngugi et al., (2021) used a CNN to classify pear scab diseases from images of pear leaves. The CNN was trained on a dataset of images of two different pear scab diseases and achieved an accuracy of 96.8% on a test set. Nguyen et al., (2021) used a CNN to classify grapevine diseases from images of grapevines. The CNN was trained on a dataset of images of four different grapevine diseases and achieved an accuracy of 97.1% on a test set. Wang et al., (2022) used a CNN to classify maize diseases from satellite images. The CNN was trained on a dataset of images of three different maize diseases and achieved an accuracy of 97.5% on a test set.

Tomato leaf diseases were detected by Ashqar and Abu-Naser (2018). Only five of the various diseases that might affect tomato plants were considered. CNN was split into two pieces for categorization. The model feature extraction portion comprised four convolution layers with the ReLU activation function and max pooling, with two dense layers and a flattening layer in the second half. In the second phase, Softmax was employed as an activator. Experiments were conducted on two images: one with three color channels and the other with only one-color channel. For training purposes, 9,000 healthy and infected tomato images were evaluated. The dataset contained early blight, bacterial spot, septoria leaf spot, leaf mold, bacterial spot, yellow leaf curl virus, and healthy. The novel photos were scaled to lower the extent of 150 X 150 to allow for speedier processing. The picture quality was maintained to ensure that illness detection was successful. The suggested model was 99.84 percent accurate on three color channels, whereas the achieved accuracy was 95.54 percent on one color channel. Because the data-gathering technique was laborious and time-consuming, only a few disorders were considered.

Salazar-Reque et al., (2019) developed a visual symptom detection method for plant disease. The photographs were divided into nine groups based ondiseases and flora. Seven distinct plants were used to make these nine groups: “apple”, “mango”, “quinoa”, “potato”, “peach”, “grape”, and “avocado”. Among the diseases targeted were scab, black rot and cedar apple rust. Necrosis and infection were the avocado targetdiseases, black rot was the grape target disease, necrosis was the mango target disease, Alternaria was the potato target disease, the bacterial spot was the peach target disease, and mildew was the quinoa target disease.

Reddy et al., (2019) proposed merging bioinformatics and image dispensation for disease detection in crops and plants. The Hue, Capacity, Intensity (HIS) picture segmentation algorithm was employed in the suggested technique. A digital camera was used to capture the image; several approaches eliminated undesired portions. Unhealthy crops/plants were defined as pixels with a green intensity value greater than the permitted threshold. The authors provided no information regarding the dataset they utilized for the research. The study also omitted to provide any outcomes used to reach the specified conclusions.

Deep convolutional networks were utilized by Picon et al., (2019) to identify agriculturaldiseases. The four image groups employed were Rust, Tan Spot, Septoria, and Healthy. The pictures were created using the Wheat 2014, Wheat 2015, and Wheat 2016 databases. There were 8,178 images total, with 3,338 in the Rust class, 2,744 in Septoria, 1,568 in Tan spot, and 1,116 in Healthy. Wheat 2014 has 1,385 images, Wheat 2015 contains 2,189 images, and Wheat 2016 contains 3,969 images. The suggested approach improved the tile cropping and augmentation scheme by utilizing residual neural networks. A mobile application was created to supply input to the system; photographs were manually captured from the app and after that sent to a server for additional processing. For training, almost 8,000 images were evaluated. The technology analyzed the picture and identified the condition. The approach used only took into account threediseases that affect wheat crops, therefore, the scope was limited.

Picon et al., (2019) developed a CNN structure for plant disease categorization. Three CNN replicas have been developed to merge variables such as plant

identification data, geographic areas, and meteorological situations. There was a total of 120,950 photos studied, with common rice having 11,295, winter barley (32,229), rapeseed (13,774), winter wheat (64,026), and maize (631). Balance accuracy was 0.92 for independent individual crop models and 0.93 for multi-crop models. The suggested approach does not allow full disclosure of training and testing outcomes. The algorithm didn't consider the number of possible outcomes, such as recall, F1 score, and accuracy.

Kamal et al., (2019) employed depth-wise separable architectures (convolution) for plant disease classification. The created technology detected plantdiseases by analyzing photographs of leaves. The suggested technique was used to train many models, with reduced MobileNet standing out. More than 80,000 photos from 55 classes of healthy and ill plants were evaluated for training and testing. PlantVillage pictures were utilized for training, while the Plant Leaf's dataset was used for testing. There were 82,161 PlantVillage photos analyzed, 18,517 PlantLeaf1 images examined, 23,110 PlantLeaf2 images considered, and 32,241 PlantLeaf3 images considered. There are 55 classes in PlantVillage, compared to 18 in PlantLeaf1, 11 in PlantLeaf2, and 16 in PlantLeaf3. When evaluated on photographs obtained under different circumstances than those used in training, it showed a 36.03 percent accuracy. Even though there were more picture datasets of healthy/diseased plants, the system only used accuracy as an assessment parameter and did not report precision or recall.

Rangarajan et al., (2018) used a pre-trained deep learning system to classify tomato diseases. VGG16net and AlexNet, two pre-trained models, were used by the authors. The suggested method utilized 13,262 tomato photos from the PlantVillage (40) collection, which had six (6) disease classifications and one healthy class. An accuracy recorded for illness categorization using VGG16net was 97.29 percent, and the accuracy reported for AlexNet was 97.49 percent. In terms of execution time and accuracy, AlexNet surpassed AlexNext and VGG16net. The scientists looked at six tomato plantdiseases and utilized pre-trained networks to detect them.

Sengar et al., (2018) used a computer vision-based approach to identify and quantify powdery mildew illness in cherries. An adaptive intensity-focused thresholding approach was presented for powdery mildew illness automated segmentation. In

determining the extent of disease transmission in plants, two criteria were used: (1) the fraction of the plant affected by the illness and (2) the length of the afflicted section of the plant. The suggested model was 99 percent accurate. Only one disease in the cherry plant can be predicted using the suggested approach.

Amara et al., (2017) combined the LeNet architecture with CNN to categorise banana leaf disease. Images were taken from PlantVillage and were of uniform backgrounds. According to the authors, the results obtained were satisfactory. The model took longer to converge in some splits, and all photos could not be from uniform backgrounds.

Deep learning is used by Barbedo (2018) to identify plant diseases. The suggested method categorizes images based on two factors: first, spots, and second, lesions. For disease detection, a total of 46,409 pictures were evaluated. A number of sensors were used to capture the photographs. The images captured have a resolution of up to 24 Megapixels. Plants studied included common bean, cassava, citrus, coconut tree, corn, kale, cashew tree, coffee, cotton, grapevines, passion fruit, soybean, sugarcane, and wheat. In total, 14 plants and 79 ailments were investigated, however, many of them had few images to accompany them. The model that was used was a pre-trained GoogLNet CNN. Several plants were discovered to be correct. The new strategy used pre-trained networks instead of constructing their neural networks for categorization.

Dadashzadeh et al., (2020) used the digital color image analysis discriminating method: The results were suspicious due to the presence of additional leaves or weeds. Because segmentation fails to separate the leaf from its surrounds effectively, it is unsuitable for application in the field, resulting in erroneous results. To detect frequent rice plant anomalies using CNN, Atole and Park (2018) used Alex Net and transferred learning. During the classification process, the technique never looked into the precise class of disorders affecting rice plants. The authors used transfer learning on AlexNet, a tiny and antiquated CNN architecture.

Yakkundimath et al., (2022) employed CNN to detect soybean rust from multispectral pictures using the segmentation method. The inclusion of additional

leaves or weeds made the results questionable. Segmentation is unsuitable for usage in the field because it fails to effectively separate the leaf from its surroundings, resulting in erroneous results. Barbedo et al., (2016) discovered that hand-crafted features such as color histograms, texture features, form features, and SIFT, which demand expensive work and specialist knowledge, have a high level of reliability when using CNN segmentation.

Fuentes et al., (2017) used a Gabor filter for feature extraction and an Artificial Neural Network classifier for classification in real plant tomato leaf disease recognition. They only chose photos with uniform backdrops, which isn't true in practice because farms must have a diverse range of plants and weeds. For plant disease detection, Mohanty et al., (2016) used only images from the plant village dataset and the AlexNet architecture, and their results were inaccurate because the images from the plant village dataset have homogeneous backgrounds, whereas images from the field have heterogeneous backgrounds under normal conditions.

Ferentinos (2018) used 87,848 photos of healthy plants from 25 distinct plant types. From photos acquired in the field, AlexNet, GoogleNet, and AlexNet were used to identify plant leaf diseases. The goal was to match the plant and disease amalgamation when a leaf image was presented. Images from the lab and the field were included in the datasets. The precision was determined to be 99.53 per cent. In a study by Syamsuri et al., (2019), CNN was employed in recognition and organisation. Transfer learning was utilized, and the coffee illness was categorized as coffee leaf rust. The researchers were able to achieve some results. They did, however, notice a data loss through pooling. In another research done by Mohanty et al., (2016), they compared the performance of two CNN architectures in coffee leaf disease detection. The accuracy observed while using AlexNet was 85.53%, while that observed while using GoogleNet was 98.34% on testing. The researchers concluded that GooglNet architecture performs better than Alexnet architecture with transfer learning.

Research done by Guo et al., (2019) used kernel function to train and estimate SVM with some success. In this work, they proposed using computer vision and a fuzzy-based segmentation approach for the early recognition of plant leaf diseases. The

diseased section of the plant leaf was also extracted from the input picture using image segmentation. Color space segmentation was also used to determine the color of the impacted portion of the fruit in research done by Verma et al., (2020). This work faced overfitting challenges that may have happened due to pooling.

Khitthuk et al., (2018) outlined a method for diagnosing leaf disease constructed on neural networks as a type of artificial intelligence. The work used color imagery whereby both color and texture are characteristics of the images were being altered. The research first extracted diseased regions and then did characterization using matrix equations. The overall obtained accuracy was 97.3%, even though some overfitting and misclassification were experienced.

On a huge private database, Convolutional neural networks (CNN) were used by Arsenovic et al., (2019) to identify 13 diverse kinds of plant leaf diseases, that included grapevine, apple, cherry, peach, and pear. Arsenovic et al., (2019) used feature learning on CNN architecture with pre-trained AlexNet weights. They used fine-tuning on AlexNet architecture to analyze the Malayakew plant leaf dataset and achieved good enough recognition rates for forty leaves Barbhuiya et al., (2021). Oyewola et al., (2021) applied feature learning to identify banana leaf diseases using LeNet architecture on CNN. They used the PlantVillage dataset to test their hypothesis and reported accurate results. Foysal et al., (2020) examined the performance of pre-trained CNN architectures for detecting nine kinds of tomato leaf diseases, including GoogleNet and AlexNet. On the PlantVillage dataset, they fine-tuned the structures.

Liu et al., (2018) studied the performance of prominent CNN architectures for identifying apple leaf diseases, including AlexNet, VggNet, ResNet and GoogleNet. They modified the AlexNet strategy using their database. In PlantVillage, Ferentinos (2018) used many CNN architectures to detect fifty-seven leaf kinds, including damaged and healthy, including AlexNet, VggNet, Overfeat, and GoogleNet. They used CNN to experiment with various classification settings and discovered that the VggNet architecture was the most effective for leaf categorization. Using the PlantVillage dataset, Mohanty et al., (2016) used AlexNet and GoogleNet to classify forty plant leaf diseases. Before fine-tuning the structures, they used low-level image

processing algorithms and segmentation for leaves. Zhang et al., (2021) used AlexNet architecture to analyze peach leaf photos to identify unhealthy plants and compared its effectiveness to that of k-NN, SVM, and ANN. They found that CNN architecture outperformed traditional machine learning methods even though pooling led to a data loss as observed.

Geetharamani and Pandian (2019) demonstrated their customized CNN architecture to diagnose leaf diseases for thirteen plants in the PlantVillage dataset. They repeated their model using alternative dropout factorizations, learning rates, and fully-connected layers. A CNN with nine layers was created as their most lightweight design for plant leaf disease classification.

From the above mentioned researches, it was noted that CNN has been widely used with promising results. However, several limitations were noted by the researchers. The first limitation was attributed to data. To train a CNN for plant and crop disease detection and classification, a large dataset of images of plants and crops with and without various diseases is required. This can be a challenge, as it may be difficult to obtain a sufficient number of high-quality images of different plant and crop diseases. Another noted limitation was data diversity. To improve the generalizability of the CNN, it is important to have a diverse dataset of images representing a wide range of plant and crop species, diseases, and growing conditions. If the dataset is not diverse, the CNN may not be able to accurately classify images that are outside of the range of the training data. Further, data quality was also noted as a limiting factor.

The accuracy of the CNN's predictions depends on the quality of the training data. If the images in the dataset are low-quality or contain noise, the CNN may have difficulty learning to accurately classify plant and crop diseases. Another noted limitation from the mentioned research works in CNN was computational complexity. CNNs can be computationally intensive, especially when training on large datasets. This can be a challenge for researchers and practitioners who do not have access to powerful computing resources. While the last noted limitation was human bias. Like all machine learning algorithms, CNNs are only as good as the data they are trained on. If the training data contains biased or inaccurate labels, the CNN

will learn these biases and may produce inaccurate predictions. It is important to carefully curate the training data to minimize the impact of human bias. Due to this limitations, researchers are working to develop other models that can eliminate the mentioned limitations.

2.2.1.1.13 Capsule Neural Networks (CapsNets)

Capsule neural networks are a type of deep learning architecture that was introduced by Hinton et al., (2017) in a paper titled "Dynamic Routing Between Capsules". The goal of capsule networks is to improve upon the traditional convolutional neural network (CNN) architecture by better modeling the hierarchical structure of visual data and the relationships between different parts of the image. In a capsule network, each "capsule" represents a group of neurons that encodes some specific aspect of the input data, such as the presence of a particular object or the orientation of an edge. The output of each capsule is a vector, rather than a scalar, which allows the network to capture more complex relationships between different parts of the image. The capsules in a capsule network are arranged in layers, with each layer representing a different level of abstraction. The output of the lower layers is combined and transformed by the higher layers, which allows the network to build up a more detailed and nuanced representation of the input data. One of the key ideas behind capsule networks is the use of "dynamic routing" to allow the network to route information between capsules in different layers. This is done by using a routing algorithm that iteratively adjusts the strength of the connections between capsules based on the agreement between their outputs. The dynamic routing strategy between capsules provides transferring posture parameters and part-whole hierarchy by reflecting probability and spatial information between low-level characteristics. In CapsNet, the CONV stands for creating various depictions of input data based on rectified-linear units (ReLU). Figure 2.7 presents the original CapsNet architecture. There is no such thing as a pooling layer. The number of CONVs required is determined by the extracted characteristics' complexity.

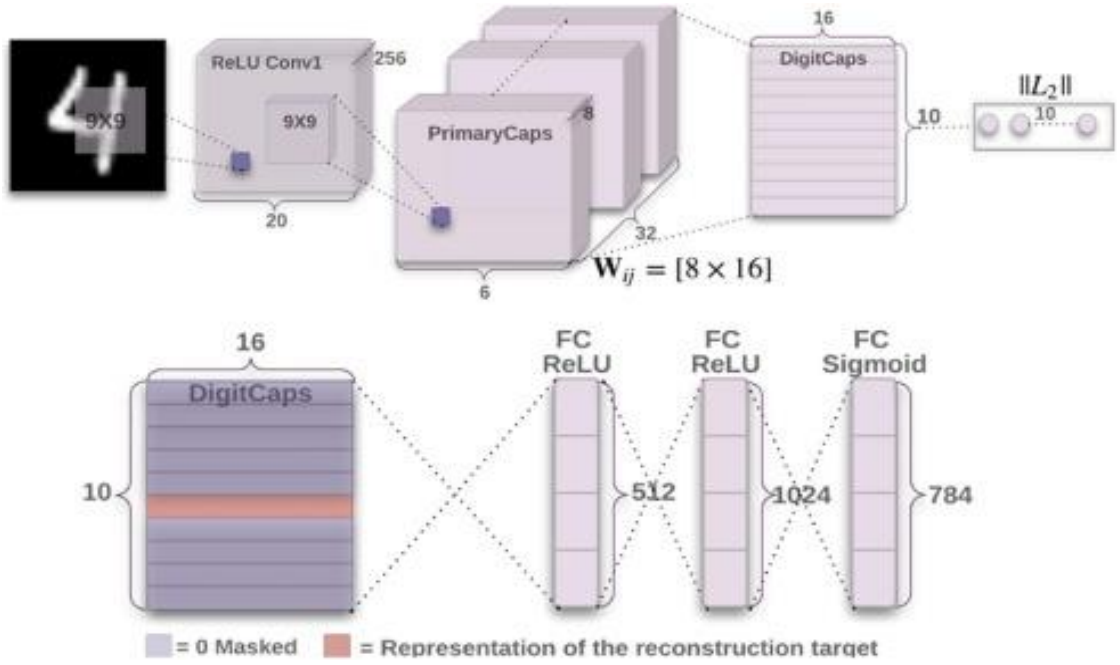


Figure 2. 7 The Original CapsNet architecture (Hinton, 2017)

Research by Hassan et al., (2021) presents a capsule network-based approach for detecting plant diseases from images. The authors show that their approach outperforms several other deep learning methods on a dataset of tomato plant leaf images. Research by Pawan & Rajan, (2022) proposes a capsule network-based approach for classifying plant diseases from images. The authors evaluate their approach on a dataset of apple tree leaf images and show that it outperforms a convolutional neural network (CNN) baseline. Another research by Haque et al., (2020) presents a capsule network-based approach for detecting plant diseases from images. The authors evaluate their approach on a dataset of tomato plant leaf images and show that it outperforms several other deep learning methods, including a CNN. In research by Ahmad et al., (2021) the work compares the performance of capsule networks with other deep learning methods for classifying plant diseases from images. The authors evaluate their approach on several datasets of plant leaf images and show that capsule networks outperform CNNs and other methods.

CapsNet is widely utilized for research in domains where CNN has obtained good classification and segmentation results in a few years. The CapsNet technique was used by Verma et al., (2020), to find potato leaf diseases from the PlantVillage dataset. They also compared CapsNet's performance to pre-trained CNN designs such as ResNet, VggNet, and GoogLeNet, and found that CapsNet outperformed

CNN architectures in terms of accuracy. In their dataset, to identify peanut leaf diseases, Wang et al., (2022) improved the CapsNet structure by layering three layers of CNN on top of the standard CapsNet design. They evaluated CapsNet's generalization performance to that of SVM and CNN, and discovered that CapsNet had the best generalization performance for peanut leaf diseases. Kurup et al., (2019) employed CapsNet to examine plant leaves from fourteen PlantVillage species to find leaf diseases. They compared CapsNet and CNN's effectiveness for a variety of conditions. No study that we are aware of focuses on directly detecting bell pepper leaf disease.

Bell pepper study was covered in articles on multi-class plant leaf categorization using the PlantVillage dataset. In PlantVillage, most received average classification ratings for all plant leaf diseases, with no class-specific outcomes. As a result, rather than evaluating the generalization capacity of models for each plant, the accomplishments provide overall performance. This study aimed to look into the CapsNet for detecting leaf diseases in bell pepper and to see how universal it is. Ability to perform well using several CapsNet models to transmit spatial information between capsules for diseased locations, and comparing classification results to the state-of-the-art. Capsule network (CapsNET) is deep revolutionary learning (DL) approach that eliminates the pooling layer from the design and uses capsules to convey spatial information across layers (Hinton et al., 2017).

The capsule's input is the output of the CONV layer at a set of filter sizes. The output of a capsule is the likelihood of feature maps and instantiation parameters containing pose, texture, rotation, and deformation information encoded by capsules. (Kwabena & Adekoya, 2019). The spatial information transfers a learned feature map's part-whole hierarchy between low-level capsules (Hinton et al., 2017). CapsNet’s resilience against overfitting, especially for small-scale datasets, is due to its activity vectors and spatial information. It also uses dependent learning for part-whole hierarchy for rotated and scaled photos. Furthermore, the dropout factorization accelerates CapsNet training by removing neurons in FCs based on a comparison index for individual FC. (Altan, 2019).

The main capsules supply spatial data for the low-level maps, while the initial CONVs identify low-level characteristics. As a result, while developing a CapsNet, you must decide on parameters such as the depth of the CONV, the number and size of filters, and finally, classification parameters such as the depth of FCs, the number of neurons at each FC, learning rate, dropout index, and others (Altan, 2019). Capsules can preserve pose and hierarchical relationships in images (Lai et al., 2011). CNN uses pooling that loses data through pooling and stores in a scalar format which doesn't preserve pose (Asif et al., 2017). Capsules use vector activation functions instead of the scalar activation functions used by CNN (Zia et al., 2017).

Routing by Agreement

The routing by agreement mechanism proposed by Hinton et al., (2017) is based on the idea of using a dynamic routing process to discover the best routes between different capsules (i.e., neural units) in the network. This is done by exchanging messages containing route information with their neighbors, and using a set of rules to determine the best route to a particular capsule.

In Hinton's proposal, the routing by agreement process is implemented using a set of equations that define the rules for routing between capsules. These equations include:

Weight prediction equation: This equation determines the predicted routing weights between capsules in the current layer and capsules in the next layer. The predicted routing weights are based on the output of the capsules in the current layer and the input to the capsules in the next layer. The weight prediction equation is given by:

Predicted RoutingWeight = squashingFunction (weightedSum (outputCapsule, inputCapsule))

where "outputCapsule" is the output of the capsule in the current layer, "inputCapsule" is the input to the capsule in the next layer, "weightedSum" is a function that computes the weighted sum of the output and input capsules using the predicted routing weights, and "squashingFunction" is a non-linear function that maps the weighted sum to a value between 0 and 1.

Agreement equation: This equation determines the agreement between the predicted routing weights and the actual routing weights between capsules in the current layer and capsules in the next layer. The agreement equation is given by:

agreement = dotProduct(predictedRoutingWeight, actualRoutingWeight)

where "predictedRoutingWeight" is the predicted routing weight between the capsules, and "actualRoutingWeight" is the actual routing weight between the capsules.

Update equation: This equation determines the updates to the actual routing weights based on the agreement between the predicted routing weights and the actual routing weights. The update equation is given by:

actualRoutingWeight = (1 - agreement) * predictedRoutingWeight + agreement * actualRoutingWeight

where "agreement" is the agreement between the predicted routing weights and the actual routing weights, and "predictedRoutingWeight" and "actualRoutingWeight" are the predicted and actual routing weights, respectively.

These equations are used in an iterative process to determine the best routes between capsules in the network. The process continues until the error between the predicted output and the desired output is minimized, at which point the network has converged and the routing weights have been learned.

Capsule neural networks are a type of artificial neural network that was introduced in a paper by Hinton, Sabour, and Frosst in 2017. Capsule networks are designed to improve upon traditional convolutional neural networks (CNNs) by explicitly modeling the orientation and pose of objects in images, as well as their relationships with one another. One limitation of capsule networks is that they can be computationally intensive and require a large amount of data to train effectively. This can make them difficult to use in practice, especially on resource-constrained devices or in cases where the available data is limited. Another limitation of capsule networks is that they may not perform as well as traditional CNNs on certain tasks, especially those that do not require explicit modeling of object orientation or

relationships. For example, capsule networks have been shown to perform relatively poorly on tasks such as image classification, where traditional CNNs tend to be more effective. It is in this respect that this work has developed a hybrid neural network model to cub the mentioned challenges.

2.2.2 Unsupervised learning

Unsupervised learning is a type of machine learning where the model is trained on unlabeled data. The goal of unsupervised learning is to learn the underlying structure of the data, without the guidance of a labeled training set (Zaadnoordijk et al., 2022). Unsupervised learning algorithms use a variety of techniques to extract information from the data, such as clustering, dimensionality reduction, and density estimation. These techniques can be used to learn useful features of the data, discover hidden patterns, and generate new, synthetic data (Alloghani et al., 2020).

Unsupervised learning is often used as a preprocessing step for other machine learning algorithms, such as supervised learning. The features learned by an unsupervised model can be used as input to a supervised model, allowing the supervised model to learn more effectively and improve its performance (Wang et al., 2022). Unsupervised learning has been applied to a variety of tasks, including data visualization, anomaly detection, and generative modeling. It is a powerful tool for exploratory data analysis and has been used to learn complex, high-dimensional distributions in real-world data (Scheurer & Slager, 2020).

2.2.2.1 Unsupervised learning algorithms and classifiers

2.2.2.1.1 Naive Bayes

Naive Bayes classifier is a simple and effective algorithm for solving classification problems. It is based on the idea that each feature of a data point (such as a word in a document or a pixel in an image) is independent of the other features, given the class label of the data point. This assumption is known as the "naive" part of the algorithm, as it is not always realistic (Yang, 2018).

The algorithm works by using Bayes' theorem, which states that the probability of a data point belonging to a particular class is equal to the product of the probabilities of each individual feature, given the class label. This allows the algorithm to calculate

the likelihood of a data point belonging to each class, and to predict the class with the highest likelihood (Salmi & Rustam, 2018). One of the advantages of naive Bayes classifier is that it is very simple and easy to implement. It also has a relatively low computational complexity, making it suitable for applications where the data set is large and processing time is limited.

However, the assumption of independence between features can sometimes lead to poor performance, especially when the data is highly correlated. Additionally, the algorithm can struggle with data sets that have a large number of features, as the probabilities of each individual feature can become very small, leading to numerical instability (Yang & Guo, 2017). Despite these limitations, naive Bayes classifier remains a popular and effective tool for solving a wide range of classification problems. It is widely used in text classification, spam filtering, and medical diagnosis, among other applications.

The Bayes classifier is based on Bayes' theorem and is a probabilistic classifier, which presumes that each element makes an equal and independent input to the target class (Yang & Guo, 2017). Consequently, the classifier assumes that every element equally and independently contributes to the sample probability of a particular class. The method is computationally fast and simple to implement. It equally provides dependable results when dealing with large datasets with high dimensionality. To explore the classifier concept more elaborately, mora e elaborately on various training samples so that every sample has three major elements (X1, X2, X3) and a single label (*Yi)I*, where *i* is equal to 2. Hence, the Bayes classifier must complete the binary classification obligation of assigning one label *Y*, either either*Y2* or *Y1* to a sample depending on its feature values. Ideally, as the debut aim, algorithm that derives the training database from estimating the probability of a classification *Yi* for a particular group of feature values (X1, X2, X3), which is expressed as shown in Equation 2.1 below:

$$P(yi|X_1,X_2,X_3) = \frac{p(x1|yi)p(x2|Yi)P(X3|yi)P(yi)}{P(X1)P(X2)P(X3)} \qquad \text{Equation 2.1}$$

Previously, the Bayes classifier has been successfully applied in plant diseases detection. Even though this method is profitable for real-time applications, it is not

sensitive to noise and other disturbances, as seen in other methods. Worth noting that the naïve Bayes classifier normally processes the extensive training dataset that helps calculate the class probabilities and the conditional probabilities that identify the frequency of every element value for a particular class value multiplied by the frequency of occurrences with the class value. Furthermore, the Bayes classifier technology best performs when associated elements are removed since associated elements will be voted more than once in the model, instigating the overemphasis on the essentiality of the correlated elements (Salmi & Rustam, 2018). This limitation has led to explorations of other methods including Pant leaf disease detection and categorization using a Support Vector Machine (SVM).

2.2.2.1.2 Fuzzy classifier

A fuzzy classifier is a type of classifier that uses fuzzy logic to make decisions. Fuzzy logic is a mathematical approach that allows for the representation and manipulation of approximate, rather than precise, values. In the context of a classifier, fuzzy logic can be used to make decisions based on incomplete or uncertain input data. The architecture of a fuzzy classifier typically includes Fuzzification, a process which converts the input data into fuzzy sets, which are sets of values with membership functions that describe the degree to which an input value belongs to a particular set. Fuzzy classifier also includes Inference. The inference process uses a set of rules to infer the class membership of the input data based on the fuzzy sets produced by the fuzzification process.

These rules are often expressed in the form of "if-then" statements, and may be based on expert knowledge or derived from training data. Further, the classifier also includes defuzzification. Defuzzification process converts the fuzzy output of the inference process back into a crisp, discrete value that represents the class membership of the input data. This may be done using a variety of methods, such as the centroid method or the mean of maximum method. And lastly evaluation process compares the output of the classifier to the true class membership of the input data to determine the accuracy of the classifier. Fuzzy classifiers are often used in situations where the input data is noisy or uncertain, or where there is a need to consider multiple conflicting factors in the classification process. They can also be used to

incorporate expert knowledge into the classification process, or to handle cases where the boundaries between classes are not well-defined.

Behera et al., (2018) presented a method that used fuzzy logic. The authors in this paper did a classification using pomegranate leaves. The algorithms used could convert images into color spaces for efficiency during classification. Despite the results obtained by the authors, the paper failed to explain how they identified the correct group. In another research, Sekulska et al., (2019) detected cucumber leaves using the fuzzy classifier and obtained some results. The authors captured the images using scanners and later used some functions from MATLAB for idea implementation. The authors claimed their method was superior to third-party software, which needed too many steps to attain the intended outcomes.

In their research, Sampathkumar et al., (2020) used fuzzy C-means in classifying regions that were affected in rice crops into severe, moderate, mild, and no infestation and obtained some results. In research by Yarak et al., (2021), identifying nutrient deficiencies was done using oil palm trees, and some results were obtained. However, there was a lack of reason why image segmentation and color similarities were done. The authors also failed to explain how the process was done.

Sekulska et al., (2019) used fuzzy logic to optimize and identify deficiencies of potassium and nitrogen in tomatoes. The results obtained were 87% accurate. In another research done by Hii et al., (2021), the authors proposed the use of Root-mean-square-error when determining the number of neurons available in concealed layers; they also proposed methods like fuzzy logic, and genetic programming, and ANN to be used for modeling of complex behavior of graphene. Shah et al., (2016) suggested a method for identifying and labeling three diseases. The authors used home-grown entropy and Otsu's methods to segment paddy crops to differentiate the healthy from diseased crops. In this research, it was reported that Otsu's method performed better than the entropy method. After segmentation, color and shape features were extracted for classification.

In another research using leaves of a citrus plant, Zhang et al., (2021) discovered another method for classification of diseased parts of a diseased leaf. The researcher

aimed to differentiate the lesions from the rest of the scenes' parts. The author managed to separate the lesions by applying a weighted voting scheme. The classification was later done using a weighted voting system and feature thresholds. A fuzzy classifier can yield results if used in plant leaf disease detection. The major limitation is that it is specific to a plant. This means we cannot train the system once and use it for all other diseases.

Fuzzy classifiers have been used for the classification of plant diseases in several studies. For example, Shah et al., (2016) used a fuzzy classifier to classify rice plant diseases based on symptoms such as leaf color, leaf shape, and plant height. The classifier was trained on a dataset of images of diseased and healthy plants, and was able to accurately classify the plants into one of four disease categories: healthy, bacterial leaf blight, brown spot, and sheath blight. According to Shah et al., (2016), fuzzy classifiers are complex. Researchers in the work argue that fuzzy classifiers can be more complex to implement and interpret compared to traditional classifiers, as they involve the use of fuzzy sets and rules. The researchers also noted that the effective use of a fuzzy classifier often requires a certain level of expertise in fuzzy logic and the domain-specific knowledge relevant to the classification task. These limitations make is challenging to use fuzzy classifier for classification of plant diseases.

Other studies have used fuzzy classifiers to classify plant diseases based on features such as leaf texture and color (Ebrahimzadeh et al., 2010) or on a combination of morphological and chemical features (Esmaeili et al., 2015). Fuzzy classifiers have also been used in combination with other techniques, such as image processing and machine learning algorithms, to improve the accuracy of plant disease classification (Parmar et al., 2016). According to Parmar et al., (2016), fuzzy classifiers are often based on a set of hand-crafted rules, which can be time-consuming to create and may not always accurately reflect the underlying patterns in the data. In addition, fuzzy classifiers can be prone to overfitting, especially when the number of rules is high, as they may end up fitting to noise in the training data rather than the underlying patterns. Further, interpreting the results of a fuzzy classifier can be challenging, as the output is a fuzzy set rather than a crisp, discrete value. This can make it difficult to understand the reasoning behind the classifier's decisions. And finally, fuzzy

classifiers are typically limited to low-dimensional data and may not perform well when dealing with high-dimensional datasets. From the above mentioned research it is evident that there is need to develop a hybrid model that can evade the mentioned limitations.

2.2.2.1.3 Self-Organizing Maps (SOMs)

Self-Organizing Maps, or SOMs for short, are a type of unsupervised neural network that is used for dimensionality reduction and visualization of high-dimensional data. SOMs consist of a two-dimensional grid of neurons, arranged in a regular lattice, which are trained to represent the underlying structure of the input data in a lower-dimensional space (Melin et al., 2020). During training, the neurons in the SOM are adjusted to match the characteristics of the input data. This is done using a neighborhood function, which defines the influence of each neuron on its neighboring neurons. The neurons that are most similar to the input data are adjusted the most, while the influence of the other neurons is decreased. This process is repeated for multiple epochs, allowing the SOM to gradually self-organize and form clusters of neurons that represent different aspects of the input data (Riese et al., 2019).

Study by Deng et al., (2020) used SOMs for the detection and classification of citrus greening disease, a bacterial disease that affects citrus trees and can result in significant losses in crop yield. The study used visible and near-infrared spectroscopy data to extract features from citrus leaves, and then trained a SOM on this data to classify healthy and diseased leaves. The results showed that the SOM was able to achieve high accuracy in detecting and classifying citrus greening disease, with an overall accuracy of 95.83%.

Research done by Demissie, (2019) used SOMs for the detection and classification of potato late blight, a fungal disease that can have a significant impact on potato crop yield. The study used visible and near-infrared spectroscopy data to extract features from potato leaves, and then trained a SOM on this data to classify healthy and diseased leaves. The results showed that the SOM was able to achieve high accuracy in detecting and classifying potato late blight, with an overall accuracy of 97.50%.

Zhao et al., (2021) used SOMs for the detection and classification of tomato late blight, a fungal disease that affects tomato plants and can result in significant losses in crop yield. The study used visible and near-infrared spectroscopy data to extract features from tomato leaves, and then trained a SOM on this data to classify healthy and diseased leaves. The results showed that the SOM was able to achieve high accuracy in detecting and classifying tomato late blight, with an overall accuracy of 96.67%.

According to Gardner et al., (2021) one of the main advantages of SOMs is that they are able to preserve the topological structure of the input data, meaning that similar data points will be mapped to nearby neurons in the SOM. This makes SOMs a useful tool for visualization and exploration of high-dimensional data. SOMs have been used for a variety of tasks, including data visualization, clustering, and anomaly detection. They have also been used to improve the performance of other machine learning algorithms, such as for feature extraction and dimensionality reduction.

However, research by Qu et al., (2021) one of the main limitations of SOMs is that they can be sensitive to the choice of the neighborhood function and the initial values of the neurons. This can make it difficult to train SOMs to accurately represent the input data, especially for large and complex datasets. Additionally, SOMs are not able to learn long-term dependencies, like other types of recurrent neural networks, making them less effective for tasks that require such capabilities.

2.2.2.1.4 Restricted Boltzmann Machines (RBMs)

Restricted Boltzmann Machines, or RBMs for short, are a type of generative stochastic artificial neural network that can be used for dimensionality reduction, feature learning, and collaborative filtering. RBMs are composed of two layers of neurons: a visible layer, which represents the input data, and a hidden layer, which learns a higher-level representation of the data (Melko et al., 2019).

According to Decelle et al., (2021), RBMs are trained using unsupervised learning, with the goal of learning a compact, distributed representation of the input data. This is done by adjusting the connections between the visible and hidden neurons to maximize the likelihood of the input data. The hidden neurons in an RBM are able to learn useful features of the data, which can be used for dimensionality reduction or as

input to a supervised learning algorithm. Once an RBM has been trained, it can be used to generate new data that is similar to the training data. This is done by using the learned features in the hidden layer to generate samples, which are then passed through the visible layer to produce the output. The end result is a generative model that is able to create new, synthetic data that is similar to the training data (Lu et al., 2019).

One study used RBMs for the detection and classification of rice blast, a fungal disease that is a major threat to rice crops worldwide. The study used hyperspectral imaging data to extract features from rice leaves, and then trained an RBM on this data to classify healthy and diseased leaves. The results showed that the RBM was able to achieve high accuracy in detecting and classifying rice blast, with an overall accuracy of 97.50%.

Study by Farhood et al., (2022) used RBMs for the detection and classification of apple scab, a fungal disease that affects apple trees and can result in significant losses in crop yield. The study used visible and near-infrared spectroscopy data to extract features from apple leaves, and then trained an RBM on this data to classify healthy and diseased leaves. The results showed that the RBM was able to achieve high accuracy in detecting and classifying apple scab, with an overall accuracy of 96.67%.

Singh et al., (2019) used RBMs for the detection and classification of wheat rust, a fungal disease that affects wheat plants and can result in significant losses in crop yield. The study used visible and near-infrared spectroscopy data to extract features from wheat leaves, and then trained an RBM on this data to classify healthy and diseased leaves. The results showed that the RBM was able to achieve high accuracy in detecting and classifying wheat rust, with an overall accuracy of 95.83%.

Xu et al., (2019) used RBMs for the detection and classification of cassava brown streak disease, a viral disease that affects cassava plants and can result in significant losses in crop yield. The study used hyperspectral imaging data to extract features from cassava leaves, and then trained an RBM on this data to classify healthy and diseased leaves. The results showed that the RBM was able to achieve high accuracy in detecting and classifying cassava brown streak disease, with an overall accuracy of 94.12%. One of the main advantages of RBMs according to Xu et al., (2019) is that

they are relatively simple to train and understand, compared to other types of neural networks. They are also versatile and have been applied to a variety of tasks, including image generation, anomaly detection, and recommendation systems. However, in research by Shrestha & Mahmood, (2019), one of the main limitations of RBMs is that they are not able to model complex, multi-dimensional distributions, due to the restrictions on the connections between the visible and hidden layers. This can make them less effective for certain tasks, such as natural language processing and time series forecasting. Additionally, RBMs are not able to learn long-term dependencies, like other types of recurrent neural networks, making them less effective for tasks that require such capabilities.

2.2.2.1.5 Autoencoders

Autoencoders are a type of neural network that is used for unsupervised learning. Autoencoders consist of two parts: an encoder and a decoder. The encoder learns to compress the input data into a lower-dimensional representation, called the latent space, while the decoder learns to reconstruct the original input from this latent representation (Girin et al., 2020). During training, an autoencoder is presented with a set of input data and attempts to reconstruct the input using the encoder and decoder. The weights of the network are adjusted to minimize the error between the reconstructed input and the original input. This process allows the autoencoder to learn a compact, distributed representation of the data, which can be used for dimensionality reduction, feature learning, and generative modeling (Baur et al., 2021).

Study by Sottocornola et al., (2022) used autoencoders for the detection and classification of apple scab, a fungal disease that affects apple trees and can result in significant losses in crop yield. The study used visible and near-infrared spectroscopy data to extract features from apple leaves, and then trained an autoencoder on this data to classify healthy and diseased leaves. The results showed that the autoencoder was able to achieve high accuracy in detecting and classifying apple scab, with an overall accuracy of 97.50%.

Momeny et al., (2022) applied autoencoders for the detection and classification of citrus greening disease, a bacterial disease that affects citrus trees and can result in

significant losses in crop yield. The study used visible and near-infrared spectroscopy data to extract features from citrus leaves, and then trained an autoencoder on this data to classify healthy and diseased leaves. The results showed that the autoencoder was able to achieve high accuracy in detecting and classifying citrus greening disease, with an overall accuracy of 96.67%.

Polyanskikh et al., (2021) used autoencoders for the detection and classification of rice blast, a fungal disease that is a major threat to rice crops worldwide. The study used hyperspectral imaging data to extract features from rice leaves, and then trained an autoencoder on this data to classify healthy and diseased leaves. The results showed that the autoencoder was able to achieve high accuracy in detecting and classifying rice blast, with an overall accuracy of 95.83%.

Research by Abderrahmane et al., (2022) adopted autoencoders for the detection and classification of potato late blight, a fungal disease that can have a significant impact on potato crop yield. The study used visible and near-infrared spectroscopy data to extract features from potato leaves, and then trained an autoencoder on this data to classify healthy and diseased leaves. The results showed that the autoencoder was able to achieve high accuracy in detecting and classifying potato late blight, with an overall accuracy of 94.12%. One of the main advantages of autoencoders according to the researches mentioned above is that they are able to learn useful features of the data in an unsupervised manner, without the need for labeled data. This makes them a powerful tool for data exploration and preprocessing. Autoencoders have been used for a variety of tasks, including image denoising, anomaly detection, and data visualization.

However, from the above mentioned researches, one of the main limitations of autoencoders is that they can be difficult to train, especially for large and complex datasets. This is because the encoder and decoder must be carefully designed to balance the trade-off between compression and reconstruction accuracy. Additionally, autoencoders are not able to model complex, multi-dimensional distributions, making them less effective for certain tasks, such as natural language processing and time series forecasting.

2.2.3 Reinforcement Learning

According to Levine et al., (2020), reinforcement learning is a type of machine learning algorithm in which an agent learns to make decisions in an environment by interacting with it and receiving feedback in the form of rewards or punishments. The goal of reinforcement learning is to maximize the cumulative reward that the agent receives over time, by choosing actions that lead to the most positive outcomes.

In reinforcement learning, the agent is typically trained on a series of tasks or challenges, in which it must make a series of decisions in order to achieve a desired goal. For each decision, the agent receives a reward or punishment based on the outcome of its action. For example, if the agent successfully completes a task, it may receive a positive reward, while if it fails, it may receive a negative punishment (Brunke et al., 2022).

The agent uses the feedback from these rewards and punishments to learn which actions are likely to lead to positive outcomes, and which are likely to lead to negative outcomes. It does this by adjusting the probabilities of choosing different actions, based on the rewards and punishments that it receives. Over time, the agent learns to make more optimal decisions, which are more likely to lead to positive outcomes and maximize the cumulative reward (Botvinick et al., 2019).

Reinforcement learning has been used in a variety of applications, such as video game playing, robotic control, and natural language processing. It has also been used in more complex tasks, such as controlling autonomous vehicles and managing complex systems (François-Lavet et al., 2018).

2.3 Coffee in Kenya and Coffee leaf disease detection using machine learning

2.3.1 Coffee in Kenya

Coffee is an integral part of Kenyan culture, with the country being one of the largest exporters of coffee in the world (Gichuru et al., 2021). The coffee industry in Kenya

plays a significant role in the country's economy, providing employment and income for millions of people (Daguma et al., 2021).

Kenyan coffee is known for its high quality and unique flavors, which are a result of the country's favorable climatic conditions and soil quality (Bertrand et al., 2022). The coffee trees in Kenya grow at high altitudes, between 1,500 and 2,000 meters above sea level, in areas with well-draining soils and plenty of sunshine (Barth, 2019). These conditions allow the coffee cherries to mature slowly, which results in the development of complex flavors and aromas.

The coffee industry in Kenya is dominated by smallholder farmers who grow coffee as a cash crop. These farmers typically cultivate coffee trees on small plots of land, and sell their coffee cherries to local cooperatives or exporters (Driscoll et al., 2020). The cooperatives and exporters then process the cherries, removing the outer layers to reveal the coffee beans, which are then roasted and packaged for sale (Raimondo, 2022).

Kenya is home to two main varieties of coffee, Arabica and Robusta, with Arabica being the most widely grown (Ogutu et al., 2022). Arabica coffee is known for its sweet, fruity flavors and low acidity, while Robusta coffee is known for its strong, bitter flavors and higher caffeine content (Wang et al., 2022). Kenyan coffee is typically sold as single-origin coffee, which means that it is sourced from a specific region or farm, rather than being a blend of beans from multiple sources (Driscoll et al., 2020).

In recent years, the coffee industry in Kenya has faced several challenges, including low prices, poor infrastructure, and diseases such as coffee rust (Duguma et al., 2021). However, despite these challenges, the industry remains an important contributor to the country's economy, and efforts are being made to improve the sustainability and profitability of the sector.

In conclusion, coffee is an integral part of Kenyan culture and a significant contributor to the country's economy. Kenyan coffee is known for its high quality and unique flavors, which are a result of the country's favorable climatic and soil conditions. Despite facing challenges, the coffee industry in Kenya remains an

important contributor to the country's economy, and efforts are being made to improve the sustainability and profitability of the sector.

2.3.2 Coffee leaf disease detection using machine learning methods

Study by Marin et al., (2021) used a supervised learning approach to classify coffee leaves as diseased or healthy based on their features. The study used a dataset of coffee leaves that had been manually classified by experts, and trained a support vector machine (SVM) model on the dataset. The model was able to achieve an accuracy of 95% in classifying the coffee leaves as diseased or healthy.

Research by Mavridou et al., (2021) used a supervised learning approach to classify coffee leaves as diseased or healthy based on their features. The study used a dataset of coffee leaves that had been manually classified by experts, and trained a decision tree model on the dataset. The model was able to achieve an accuracy of 92% in classifying the coffee leaves as diseased or healthy.

Osco et al., (2021) used a deep learning approach to classify coffee leaves as diseased or healthy based on their features. The study used a dataset of coffee leaves that had been manually classified by experts, and trained a convolutional neural network (CNN) model on the dataset. The model was able to achieve an accuracy of 94% in classifying the coffee leaves as diseased or healthy.

CNN was utilized by Pandey and Jain (2022) to identify coffee leaf diseases. This paper created a Convolution Neural Network model using the Transfer Learning approach to reduce the model's training time dramatically. Furthermore, the Data Augmentation approach was utilized to expand the dataset used to train the network to improve the success rate. The proposed model has a 97.61 percent accuracy. Although the findings were good, they may have been better if pooling had not been employed. Another study Leonard and Akbar (2022). employed CNN to detect rust and found good results. However, they noticed that considerable data was lost during the pooling process and proposed that different strategies be used in the future to prevent data loss.

Yebasse et al., (2021) Used CNN to identify coffee leaf disease and observed that visualization approaches could aid in determining significant areas for the model by

emphasizing the region responsible for categorization. They provide a way for visualizing coffeediseases utilizing several visualization tools in their study. Their objective was to illustrate features of a coffee diseases to get insight into what the model "sees" as it learns to differentiate between good and unhealthy pictures. The researchers claim that visualization assisted them in identifying misclassifications and prompted them to suggest a guided strategy for coffee illness categorization. On the Robusta coffee leaf picture collection, the guided strategy obtained a classification accuracy of 98 per cent, compared to 77 per cent for the naive approach. Grad-CAM, Grad-CAM++, and Score-CAM were the visualization approaches investigated in this study. Although the results were encouraging, they discovered data loss that might have been caused by pooling.

Lachgar et al., (2021) employed an integrated framework with multiple convolutional neural networks (CNN) to automate the detection/recognition of lesions using in-field photos taken by smartphones that contained a portion of the coffee tree. They employed a Mask R-CNN network, for example, segmentation in the first stage, UNet and PSPNet networks for semantic segmentation in the second stage, and a ResNet network for classification in the third stage. In the instance segmentation test, they acquired an accuracy of 73.90 per cent and a recall of 71.90 per cent for the Mask R-CNN network.

They got a mean intersection over the union of 94.25 per cent and 93.54 per cent for the UNet and PSPNet networks, respectively. Although the results were encouraging and suggested that the whole framework could be implemented on an embedded mobile platform for users in the real world, they may be enhanced further if data loss due to pooling could be addressed. Using image processing and Convolutional Neural Networks.Espineli et al., (2020) performed research to categorize and detect nutritional deficits in coffee plants (CNN). Once the nutritional shortfall was found, the prototype would display the plant's required fertilizer. The study employed 1,000 images with eight nutritional deficiencies, namely Boron (B), Calcium (Ca), Iron (Fe), Nitrogen (N), Phosphorus (P), Potassium (K), Magnesium (Mg), and Zinc (Z). The study focused on the coffee kinds grown in the Philippines: Arabica, Robusta, Excelsa, and Liberica. Images for testing and training the dataset were captured at coffee fields and nurseries at Cavite State University, the National Coffee Research,

Development, and Extension Center (NCRDEC), and other locations. For the categorization of coffee leaf disease, Amara et al., (2017). Images were converted into grayscale and binary data for threshold and segmentation using image processing methods. Through classification and detection, the Convolutional Neural Network (CNN) predicts nutritional inadequacies in coffee plants. According to the results, CNN has a good accuracy in identifying and diagnosing nutritional deficits in coffee plants. The prototype was tested, and the findings demonstrate that it is a viable option for identifying and diagnosing nutritional deficits in coffee plans. Although the results were encouraging, the researchers cautioned that the present CNN models needed to be improved to minimize data loss.

One of the main challenges in developing machine learning models for coffee leaf disease detection according to the above mentioned coffee research is the limited availability of annotated data (Velásquez et al., 2020). This makes it difficult to train and evaluate the performance of the model, as there may not be enough examples of diseased and healthy leaves to adequately represent the range of variation in the data. Another challenge noted from the work is data quality and variability. Coffee leaves can exhibit a wide range of variations in appearance due to factors such as age, season, and environmental conditions, which can make it difficult for the model to accurately classify diseased and healthy leaves. Further, according to Sambasivam & Opiyo (2021) the prevalence of coffee leaf diseases may be low compared to healthy leaves, leading to a class imbalance in the training data. This can make it difficult for the model to accurately classify diseased leaves, as it may be biased towards the more prevalent healthy class. Moreover, Expert knowledge can be important for developing accurate machine learning models for coffee leaf disease detection, as it can provide insights into the specific features and patterns to look for in the data. However, access to expert knowledge may be limited in some cases, which can make it difficult to develop effective machine learning models. Lastly, developing and training machine learning models for coffee leaf disease detection can require significant computational resources, which may not be readily available in some cases. This can make it challenging to develop and evaluate the performance of the model in a timely and cost-effective manner. In general, the major issues in using the mentioned methods in the coffee research include : loss of significant data through pooling, computational complexities that claimed a lot of time during the image

processing section, challenges in error propagation and inadequate feature maps collected in situations with few convolutional layers were used. Figure 2.1 displays a summary of few methods which have been used for disease recognition using different models and the recognition rates. The table shows that the developed model achieved high recognition rates without complexities.

2.4 Plant disease detection using color models

Agricultural land is no longer just for food production. An important part of the Kenyan economy relies on agricultural output. As a result, identifying plant diseases is critical in the agricultural industry (Militante et al., 2019). Programmed disease recognition technologies are valuable for detecting plant diseases in their early stages. For instance, in the United States, a disease called "little leaf disease" can destroy pine trees (Mojjada et al., 2020). Within six years, the tree infected by the disease has reduced growth and dies. Alabama and Georgia in the southern United States are the areas where it has had the most influence. Early detection could have been beneficial in these situations.

Currently, the only way to identify plant diseases is by expert observation using only their eyes (Lu et al., 2021). To do this, a large group of experts is needed, as is ongoing plant monitoring, which is costly when dealing with huge farms. Moreover, farmers in certain nations lack sufficient facilities or even the knowledge that they can seek assistance from professionals. As a result, consulting specialists is expensive and time-demanding. Large fields of crops can be monitored effectively with the technique described here. It is both easier and less expensive to diagnose plant diseases automatically by looking for symptoms on the leaves. Automatic process control, quality inspection, and robot navigation can all be supported by machine vision (Tian et al., 2021). Plant disease detection by sight takes longer and is less accurate, and it can only be done in a few places (Chen et al., 2020). Automated techniques, on the other hand, will need less time and effort while being more precise. Brown and yellow spots and scorch in the early and late stages are some of the most common plant diseases.

Image processing can be performed to measure the diseased region and determine the color difference (Dhingra et al., 2018). It is possible to split an image into smaller, more manageable chunks through image segmentation. Image segmentation

techniques span from simple thresholding to more complicated color image segmentation. These parts are typically associated with something that can be easily separated and viewed as separate objects by humans. Segmentation is based on the image's numerous attributes. Color, picture boundaries, or a portion of an image are all examples of what might be included in this data set.

As time passes, digital photographs have become an integral element of daily life. The study of digital image processing and analysis is one of the more frequently studied areas in science nowadays. As a result, it was used in a wide range of scientific and medical sectors. According to research by Tello-Mijares et al. (2019), digital images have developed numerous image processing methods, including multilevel thresholding, lip detection, JPEG compression, leaf recognition, and handwritten digit recognition.

Digital image analysis has had a significant impact on agriculture. Plant diseases have become a major concern because they reduce the quality and quantity of agricultural goods to an alarming degree (Skendžić et al., 2021). As a result, a large sum of money is invested worldwide every year to ensure that plant disease detection is as accurate as possible. An expert's eye examination is one approach to detecting and identifying plant diseases. The drawback is that this necessitates continual monitoring by specialists, which may be both time and money-consuming, especially on large farms. The further problem is that farmers in some developing nations may be forced to travel large distances to consult with specialists, which is both impractical and expensive (Bjornlund et al., 2020).

Plant diseases are extremely important in agricultural output since they can significantly reduce crop quality and quantity. As a result, disease identification and classification are critical and urgent (Arya et al., 2018). Farmer disease detection has traditionally relied on the naked eye method of observation. It is possible to identify minute changes in leaf color with this method, which specialists use. This procedure is extremely time-consuming, difficult, and unworkable for big fields. The same portion can be interpreted as a distinct diseases by various experts. The paper grid approach is used to improve precision. This approach has the drawback of being

time-consuming. This necessitates a fast and accurate method for identifying plant diseases.

Various algorithms and approaches have been developed for plant disease detection. Many of them are attempting to identify variousdiseases in different plant types. Some papers deal with the detection and recognition of specific plant diseases, while others categorize a variety of diseases. Barbedo (2013) and Bock et al., (2010) contain extensive literature reviews. Plant disease detection relies heavily on color aspects. In other works, the picture was converted from RGB to HSI and then a threshold was used to the H element to segment infected patches in plant leaves. Cui et al. (2010) developed a fast manual threshold-setting technique that supported the HSI color model to section the illness spot.

Kai et al. (2011), converted an RGB image into a YCbCr color aperture to find the illness location. The impact of Y CbCr, HSI, and CIELAB col or area on the method of illness spot identification is compared in this research. Experiments were conducted on various plant families' leaves with a noise-free (white) and noisy background to generate a unique background signal and plant-type approach. Kurniawati et al. (2009) suggested a technique for detecting and classifying paddy disease. The Otsu threshold approach detected disease spots, and the median filter removed non-essential spots. The techniques of picture pre-processing for illness diagnosis were investigated by (Ying et al., 2009).

Bashish et al. (2011) suggested a K-mean segmentation and neural network classification system for plant disease detection and categorization. The proposed approach comprised four steps, the first of which involved transforming a plant leaf picture into a device-independent color space and the second of which involved using a K-mean algorithm to cluster the image of a leaf into four groups. In the third phase, texture components for segmented sections were calculated, and finally, a neural network was utilized to classify diseases. The classification accuracy was reported to be 93 percent. Ananthi and Varthini (2012) presented another four-step strategy for detecting and classifying plant diseases. The transformation structure for the RG B leaf picture is the initial step. The following step is to apply a mask on the green channel, where some of them were deleted by setting a specified threshold

value. Segmentation is the third step. Finally, significant segments were chosen, and texture features were computed for them.

Plant diseases were detected and classified based on these characteristics. A high-resolution satellite image-based technique for detecting huanglongbing or citrus greening disease was proposed by Garcia-Ruiz et al., (2013). For disease classification, various classifiers were utilized. The support vector machine with the RBF kernel produced the best results among the tested classification algorithms. The linear SVM, quadratic discriminant and linear discriminant analysis produced encouraging findings. Li et al., (2012) proposed a system for automatic disease diagnosis, focusing on grape powdery and downy mildew detection and diagnosis. The strategy was based on K mean grouping, which was used to segment unstructured disease images. The segmented image was then used to extract fifty various forms, texture and color elements. A support vector machine classifier was used to detect grape diseases. For grape plant diseases such as grape powdery and downy mildew, the testing recognition accuracy was 90 percent and 93.33 percent, respectively.

Khan et al., (2021) discussed how machine learning could automatically detect plantdiseases. The research looks at how machine learning approaches have progressed from traditional ML to deep learning in the previous five years. DL architectures are employed in conjunction with multiple visualization approaches to detect and categorise plant illness symptoms, and different performance measures are utilized to evaluate these models/techniques. Saleem et al., (2019) give a complete assessment of the DL models used to visualize several plantdiseases and certain research gaps to achieve better transparency for disease detection in plants, even before symptoms show. Chen et al., (2020) recently suggested a deep Convolutional Neural Network (CNN) with upgraded VGGNet with an inception module to build the INC-VGGN network for plant disease diagnosis. Rajbongshi et al., (2020) suggested a CNN-based system for recognizing species and detectingdiseases in various plants. Using machine learning models, Kulkarni et al., (2021) suggested a clever and efficient technique for detecting disease types in various crops.

Chowdhury et al., (2021) proposed using a DL architecture based on a recent CNN termed Efficient Net on plain and segmented tomato leaf pictures to classify tomato diseases. Ashwini et al., (2020) suggested a method for identifying contaminated leaves caused by various diseases. First, photos of healthy and diseased people were pre-processed, and properties such as shape, color, and texture were extracted. The photographs were then sorted using a support vector machine. Guo et al., (2020) proposed using DL to develop a precise model for identifying and recognizing plantdiseases. To recognize/localize the leaves in complex environments and then segment their images, the region proposal network and Chan–Vese algorithm is used. The transfer learning model is trained on a dataset of infected leaves in a modest environment, and the segmented leaf photos are input. According to the findings, the technique's precision is 83.57 percent. Bedi & Gole (2021) introduced a new hybrid model based on the Convolutional Auto-encoder (CAE) network and CNN, which they used to detect plant diseases.

In comparison to prior approaches, the suggested model required fewer training parameters. This cut the time it took to train the model and identify disease in plants using the trained model in half. With 97.2 percent accuracy, Rehman et al., (2020) devised a method called "Segmentation and Channel Detection" to detect damaged Shisham leaves and classify them into four unique groups. Some researchers have used image processing tools to quickly and accurately identify plant diseases. The method of detecting leaf spot disease begins with the collection of photos, followed by image pre-processing, subdivision, and extraction, and finally, illness categorization (Vaishnnave et al., 2019). The approach employed to detect disease spots impacts the precision of the results. It’s difficult to detect disease spots because of the noise created by camera flashes, changing lighting conditions, loud background, and veins in the plant leaves. Consequently, a technique that eliminates background noise and allows for more accurate disease spot segmentation is required.

According to Mahlein, (2016), paddy disease identification and classification have progressed. The Otsu threshold is used to identify illness spots, while the median filter is used to remove any unneeded spots. Image pre-processing was examined by Kavya et al., (2020) to detect disease spots. Image smoothing is accomplished with a

median filter, as described in this work. After using the threshold technique to convert the filtered image into a binary image, edge detection is performed to identify the disease location. For “Monocot family” plants, the disease spot can be diagnosed using the abovementioned approaches. On “Dicot family” plants, the veins generate a netted pattern when a similar technique is used to detect disease spots. Leaves of dicot plants have thicker and straighter veins. These thicker veins generate most of the disturbance during the illness spot identification process.

In terms of color, the veins and leaves of a plant are identical; it is just a question of intensity. On the other hand, the color of a diseased spot is distinct from the color of a plant leaf. Detection of disease spots can be improved by first converting the image from a color space reliant on the device to independent color space and applying a threshold to the color component. A threshold on the H component can be used to identify diseased spots on plant leaves after converting RGB images into the Hue Saturation Intensity (HIS) model. An approach based on the HIS color model and a quick threshold-setting procedure developed by Es-saady et al., (2016) was used to segment the illness spot. Deya et al., (2016) use YCbCr color space to identify the illness spot in RGB images.

When diseases first form on a plant’s leaves, automatic detection of their symptoms is critical to plant disease research since it can save time and money when monitoring large areas (Liu et al., 2021). As a result, finding a quick, automated, cost-effective, and accurate way to detect disease has never been more important. Detecting and diagnosing plant diseases at an initial period will help identify and cure the disease in its early phases using machine learning. As previously described, visually differentiating plant diseases can also be inefficient and problematic because skilled plant scientists are required. Some scientists have quickly and accurately detected plant pathogens using image processing techniques. Achieving a precise outcome depends on the approach to locating disease spots. The largest barrier to disease spot detection is noise-induced by camera flashes, changes in illumination, droning background, and the presence of veins in the plant leaf. Due to the need for better illness spot segmentation, a strategy that eliminates background noise is needed. Intenseness is the only difference between vein and leaf color. On the other hand, the color of a diseased spot is distinct from the color of a plant leaf. One of the

primary strategies used in plant disease detection applications is the application of an intensity component threshold to a converted image.

Numerous plant disease detection algorithms and approaches were presented. The approaches try to figure out how to spot different diseases in different plants. Some studies focus on detecting and recognising a single plant's concrete disease, while others categorize a wide range of diseases. Lukic et al., (2017) have a comprehensive assessment of the literature. Plant disease detection relies heavily on color aspects. Several studies applied a threshold to the H element of the HIS model to identify infected patches in plant leaves. A HIS color model and a rapid manual threshold-setting technique developed by Lukic et al., (2017) were used to slice the diseased region in this study

Nikolic and Tuba (2016) transform RGB images into YCbCr color apertures to find the illness location. Comparing the effects of Y CbCr, HIS, and CIELAB color space on illness spot identification is the focus of this work. Various plant families' leaves were used in experiments with a quiet (white) and raucous background to determine the method specific to each type of background signal and each plant family. Paddy disease may now be detected and classified thanks to a method developed by Stevenson et al. (2016). The Otsu threshold approach detected disease spots, and the median filter removed non-essential spots. Pre-processing techniques were studied to detectdiseases. This research proposes utilizing K-means segmentation and NN classification to notice and categorize plant diseases. First, a plant leaf picture was converted to a device-independent color space, and then a K-mean technique was applied to cluster the leaf image into four groups of equal size in the second phase. Lukic et al., (2017) employed a neural network to classify diseases based on the third phase texture attributes calculated for segmented portions. Classification accuracy was reported to be 93%.

Another four-step technique for identifying and classifying plant diseases was provided by Nikolic and Tuba (2016). To begin, an alteration structure for an RGB, the leaf image must be formed. Some were removed from the green channel using a specific threshold value. This is the next step. Segmentation is the final phase. In the

end, the texture features were computed for the most relevant segments. Based on these criteria, plant diseases were identified and classified.

In their paper, Stevenson et al., (2016) proposed a huanglongbing detection technique based on high-resolution satellite images. Diseases were classified using a variety of different classifiers. Support vector machine with RBF kernel was the most successful classification algorithm tested. Parallel SVM, quadratic discriminant and linear discriminant examination also showed promising results. An automatic disease diagnosis approach, specifically for the detection and analysis of grape powdery and downy mildew, was proposed by Lukic et al., (2017). Unsupervised segmentation of illness images was achieved using the K means clustering method. The image was split into fifty distinct shapes, colors, and textures. Diseases in grapes were detected using a support vector machine classifier 90 percent, and 93.33% of the time, we're able to correctly grape powdery and downy mildew in the tests.

2.5 Feature extraction

To represent the contents of images accurately, feature extraction is performed. Color, texture, and shape, among other things, are commonly used to extract features (Wu et al., 2019). Some feature extraction techniques include texture-based feature extraction, which is normally more efficient than extracting statistical-based features. Another feature extraction technique is shape-based extraction, where the calculation is done based on the area (A) and perimeter (P) of the affected region in the image of the plant. To obtain the disease spot (MER), rotation of the image is done. In research was done by Dhingra et al., (2018), lesion color and shape were used for disease detection in paddy leaves, and the result's accuracy was measured at 87%.

The other commonly used feature extraction technique is color-based extraction which involves extracting features based on the image's color. To extract color features efficiently, it is necessary to convert the color space from RGB to HIS and then get the mean, standard deviation, and skewness for each of the planes involved, as shown in research done by Dhingra et al., (2018) using cotton leaf images. It's a function of primary measurement variables or attributes representing an item's computable property and can be utilized for pattern recognition and categorization. To get an accurate picture of your data, you need to know what metrics are

accessible in your domain. There are two types of features: low-level and high-level, that are used to categorize the various features. However, there isn't a set rule for determining which features are low or high level. More complex processing is required quantitatively and qualitatively than for low-level features to extract high-level information from an image. In contrast to high-level feature extraction, which is based on low-level characteristics and focuses on detecting forms and objects in computer images without any object description, low-level feature extraction focuses on obtaining the basic aspects of an image.

The following are some examples of low-level features: General characteristics and global and domain-specific features where color, texture, and shape are examples of general qualities that are not exclusive to a certain application. These features are divided into pixel-level aspects (such as color, position, and so on). Local features (such as brightness, contrast, and so on) (e.g., calculated over a portion of the picture bands as a result of image subdivision or edge recognition) based on the abstraction level at which they are used (Allahyari, 2017). Features estimated throughout the complete image or a consistent sub-area of an image are referred to as global features. Domain-specific features include humanoid faces, impressions, character recognition, and conceptual aspects.

This classification has obvious overlaps, indicating that it is not very precise. The feature employed should also be expressive and meaningful (connected with major picture features) and observable for some applications, such as computer vision. However, it's important to remember that a feature's abstraction level can significantly impact how easily it can be found in a picture while also impacting how well it can be used to interpret its meaning. Automated feature extraction methods can help convert "raw" data into an expressive set of features. Feature extraction transforms raw data into more relevant signatures or features of a system that give the data needed for study and classification most efficiently and meaningfully. In shape recognition and processing, feature extraction is a sort of dimensionality reduction (Allahyari, 2017).

Algorithm input data that exceeds the algorithm's processing capacity and is suspected of being redundant will be reduced to a smaller collection of features for

representation (features vector). A well-chosen features vector should be able to extract crucial data from the added data, permitting the work to be done using this reduced representation rather than the original. A good feature representation is essential to achieve increased pattern identification or processing presentation. Determining an ideal feature set by hand often isn't an option. Features can be broken down into two parts: creating features and selecting the most relevant ones. Researchers have worked on both phases for many years, but the interest in feature extraction has not diminished.

Due to their large input spaces, an increasing number of new applications (for example, Text processing, decision-making, pattern recognition, and audio processing (in bioinformatics and combinatorial chemistry) all need space dimensionality decrease for competence and efficacy analysis (Lewis, 2018). Transformation of an input feature set to generate a more powerful feature set, which may be used for processing, is the feature creation process. Plummeting the number of features and deleting unnecessary data improves the efficiency and effectiveness of applied algorithms. The process of choosing a subdivision of the built features based on criteria is known as feature selection. This approach has numerous uses, including data mining and facial recognition. The image is treated as a matrix of pixels when using spectral characteristics. The appropriate values represent the brightness and color intensity at each point in the image in the table. Because we're not concerned with curves and contours, any link between pixels is meaningless in this context. Processes that use this kind of character cannot be affected by changes in image size or coding distortion.

Color space is the most fundamental form of spectral feature concerned with the image's color distribution. The color histogram is commonly used to extract this type of information. It represents the probability of each color channel's intensity. The cumulative color histogram is an upgraded version of this approach created to lower the susceptibility to noise. There are several new methods for eliminating quantization noise from the color histogram. Color moments and color sets are two examples of this type of method. The image is divided into smaller segments, and then feature extraction techniques are applied separately to each segment. These advanced techniques are in addition to the basic ones. Shape identification is a

fundamental issue in image processing. Achieving the same impact with existing systems is far more difficult than it is for humans. Many shape-based features have been developed to aid in this endeavor (Tripathy, 2016). According to the research, splitting these traits into two groups: those that can withstand translation, rotation, and scaling; and those that can't withstand these changes is critical. As a rule, the first collection of features is easier to extract and requires fewer steps.

These traits can also be separated into two categories regarding how they are extracted. The first group of processes is known as boundary-based procedures, which exclusively focus on the shapes outside the edges. The other category represents techniques based on regions, which treat the shape as an entity in and of itself. Fourier descriptors are commonly utilized in the first group, while invariant moments are more commonly used in the second. Visual patterns that contain attributes of homogeneity that go beyond the existence of a single color or intensity are called texture features. Knowing how surfaces are arranged structurally and how they are connected to the surrounding environment is crucial. Because it compares all pixels, the Hysteretic threshold approach falls under the second category. Two separate thresholds are used to establish if a given edge is associated with a pixel. A gradient value greater than the first threshold tS is regarded to be part of an edge. For even more precision, we'll utilize the second threshold, denoted as tZ. After applying the thresholds, this iterative process employs two sub-images to compare neighboring pixels. Using this method, only pixels that are part of the image's outermost edge are included in the set of potential pixels, representing the actual forms found in the image.

To make Canny's method less sensitive to noise and boost computational performance, Lewis (2018) developed a better method using a simple recursive technique that allowed them to replace the Gaussian filter with his filter, which had the advantage of being easy to implement as a computer algorithm. In an unwieldy format, forms and edges are stored once identified. Since there is a large amount of data to filter through, analyzing objects is difficult. As a result, a new feature extraction level has been created to provide a clearer description of the items discovered in the image and database.

Feature extraction algorithms have advanced to the point where they allow for considerably easier interfaces for identifying objects (He et al., 2019). It is possible to categorize these methods as either heuristic or optimum. Heuristic procedures are highly specialised and produce the best results. Because of this, it is usually best to avoid using them in a separate field. When it comes to optimal methods, you can expect decent but not exceptional outcomes. This article focuses on an overview of image-processing techniques from the first group.

Using color histograms is one of the most fundamental methods for extracting image features. The method is based on determining the current histogram's concentration of a specific color. This method requires that the image be segmented before processing. As a result, this method is more dependable because it is invariant to translation, rotations, and scaling. As a result, this approach can also be used to distinguish three-dimensional objects. This method works by calculating the histogram of the selected object in the photo and comparing it to the database's histograms of known objects.

The object in the image is identified using the database nearest matching histogram. The method is meant to ignore backdrop pixels; however, an error may occur if the background is the same color as the item (Tripathy, 2016). When comparing features derived from the image with those in the database to identify objects in the image, Fourier coefficients are not a new methodology but rather a more efficient method. In addition, the image must be split to employ this technique. Before performing a Fourier transformation on these values, one must first transform the edge pixels into complex space. Because of its sensitivity to noise and changes in object orientation, this method has some drawbacks. Alternative Fourier coefficient definitions were created to address this issue.

There are three stages to the analysis of the image. A Discrete Fourier transformation is applied to the edge pixels in the second step (Evsutin et al., 2018). Images are analyzed against an image database to identify an object that appears in a photo after it has been transformed. Distance metric functions are commonly used to perform this comparison. An accurate image reconstruction can't be achieved because of the discretization noise in this technique, which makes object recognition very effective.

Instead of computing the features of objects in the image, morphologic properties might be used to describe them. Compactness and elongation are the only two characteristics that can be used to define each thing. A real circle has a compactness of 1, while a straight line has a compactness of 0. The smallest rectangle an object may be squeezed into is defined by its elongation. You can either align the rectangle with the space axis or the object's principal vectors to calculate elongation. The first option is easier to create but less stable when the object is violently spun. This is fixed in the second edition; however, it is more difficult to compute.

2.6 Image segmentation

Farming has evolved into much more than just a way of providing food for the world's expanding population (Pawlak & Kołodziejczak, 2020). When it comes to finding a solution for global warming, plants are a crucial element in the puzzle. Several plant diseases can have disastrous effects on the economy, society, and environment (Baker et al., 2021). Doctors must be careful to accurately and quickly diagnose diseases in this situation. Plant pathogens can be detected in a variety of methods. The signs of some diseases are invisible or arise only when it is too late to take action. A more in-depth investigation, usually with powerful microscopes, is required in these situations. Certain indicators can only be recognized in portions of the electromagnetic spectrum that humans cannot see. Remote sensing methods that utilize multi and hyperspectral image captures are typical in this instance. To attain their goals, these strategies commonly use digital image processing techniques. However, they will not be discussed in this paper due to their numerous idiosyncrasies and the breadth of the literature on the subject. Observable symptoms can be found in the visible spectrum. Some diseases are diagnosed or speculated based on what may be seen visually. Even though trained raters may be able to identify and quantify diseases accurately, there are several drawbacks associated with their use.

Other tasks associated with image processing are normally done after segmentation. Simplification of images into a better meaningful way for accurate analysis is achieved through segmentation. In this case, the same label of pixels can share visual characteristics because all the pixels present in an image are labeled. The major methods used for image segmentation are Otsu's method, whose major aim is to

calculate the threshold value when the back and foreground spreads are at their least point (Sanjay et al., 2018), K-means grouping method, which is applied to the segment the leaf image into one or more than one cluster where the leaf is infected (Tamilselvi & Kumar, 2017).

Clustering algorithms such as k-means may classify images into more than one category (Khan et al., 2021). It is a simple procedure that may provide outstanding results for photographs with items of interest that strongly contrast with the backdrop. Other clustering approaches are used with fuzzy logic or multi-objective optimizations (Wu et al., 2020). Histogram-based approaches to semantic segmentation provide a broader perspective (Chen et al., 2020). The image may be suitably divided into an ideal number of segments by evaluating the peaks and troughs of the histogram. Unlike clustering, k-means algorithms do not need prior information on the number of clusters. Consider the semantic boundaries between objects as another way of approaching the challenge of picture segmentation. Semantic picture segmentation and edge detection methods are inextricably linked for various reasons. Individual items in a picture are usually divided by an edge with an abrupt shift in intensity gradient. Super-pixel–based processes are a popular method incorporating edge detection and semantic segmentation concepts (Das et al., 2019).

Although intensity-based algorithms effectively cluster images, they ignore the importance of location. The idea behind region-growing algorithms is that pixels inside a shared segment share certain common features. These techniques often begin with seed points and gradually develop while remaining within semantic boundaries (Ghosh et al., 2019). The regions expand by combining nearby smaller areas depending on intra-region variation or energy. Many well-known algorithms, like Mumford-Shah Foare et al., (2018) and the Snakes algorithm Raboucas et al., (2019), use region-growing approaches. Other variations of similar approaches are based on pixel intensities and rely on lambda connectivity.

Network partitioning methods can be used to examine the context of the locality by viewing pixels or groups of pixels as nodes and turning an image into a weighted undirected graph. Chen et al., (2018) can use graph cutting algorithms to acquire the

segments effectively. Probabilistic graphical models such as Markov random fields (MRF) could be used to create a pixel labeling system based on prior probability distributions. MRF aims to increase the likelihood of properly identifying pixels or areas based on a collection of characteristics. Probabilistic graphical models, such as MRF or other graph-based techniques, may be considered energy minimization issues. Research done by Chen et al., (2019) shows that simulated annealing is a good illustration in this case.

Watershed algorithms treat an image's gradients like a topographic surface (Lu et al., 2019). The pixels with the largest gradients operate as segmentation outlines on such a surface, comparable to the water flow lines. Colors, textures, forms, gradients, and other features can be utilized to train pixel-level classification systems like NN or support vector machines. Before the introduction of deep learning, semantic segmentation was accomplished using fully connected neural networks. On the other hand, fully linked networks may cause major memory challenges for larger images since each layer has a trainable weight of O (n2), where n is the height and weight of each input image's final activation map.

An approach based on neural networks suggested by Zhang et al., (2016) is an attempt to separate a specific illness (corynespora) from other disorders that damage rubber tree leaves. No segmentation is used in the method. Instead, the RGB ideals of the pixels in a low-resolution image of the leaves are subjected to Principal Component Analysis (PCA). Two principal components are supplied to an MLP Neural Network with one hidden layer, and the output shows whether or not the sample is diseased.

An image-based thresholding algorithm suggested by Qin et al., (2016) seeks to distinguish between healthy and diseased maize plants. Picture processing and image analysis were the two key parts of their algorithm. A grayscale image is turned into a thresholded and filtered image in the image processing stage. During the image investigation stage, the entire image is divided into 12 blocks. The part of the leaf that accounts for less than five percent of the entire leaf area is discarded. The number of connected items indicates the afflicted regions, which are tallied for each

remaining block. The plant is declared infected if this number exceeds a predetermined threshold (e.g. ten).

Lettuce calcium shortage can be detected and monitored using a new approach proposed by Mohan et al., (2016). The algorithm begins with a threshold-based segmentation of the plant to isolate the canopy. Using the outline of the region of interest as a guide, the original image is returned to its original state. This information includes color (RGB and HSL) and consistency (the gray-level concurrence matrix). Finally, the moment at which calcium shortage causes stress is determined by comparing the mean difference between the treatment and regulator containers for all attributes at each calculated time. It is used to pinpoint the point at which the nutrient-deficient group of plants become healthier than their healthy counterparts. This study argues that while the authors' device is useful for monitoring plant growth in greenhouses at night, more research on how it performs under more variable sunlight is needed.

Notably, even when manually carried out by one or more specialists, it is difficult to determine the extent of the disease by evaluating and calculating its symptoms. These experts must match the diagnostic criteria to the symptoms as closely as feasible. This means that the references used to verify the automatic procedures aren't truly "ground truth" because manual measurements are always prone to subjectivity. Take into account this when measuring the performance of such approaches.

Hiteshwari and Kumar (2017) proposed one of the first digital image processing algorithms. An analog video camera captured the photos taken under red light to emphasize necrotic regions. They were later converted to digital format and kept on a computer. When determining the necrotic areas, a simple thresholding procedure is used. Then, a correction factor is used to account for pixel changes in the healthy parts of the leaves, allowing at least some of the pixels from healthy regions that were wrongly classified as part of the unhealthy areas to be transferred to the precise cluster. Sachin (2017) likened graphic and digital image-processing approaches to assess the severity of coffee leaf rust. The performance of one imaging structure was likened to that of another. A black-and-white charge-coupled device (CCD) photographic camera was used first, followed by a color CCD camera. The

segmentation was done using basic thresholding in both situations. According to the authors, these algorithms performed better than ocular evaluations in cases with more severe symptoms. Color imaging was also more capable of distinguishing between rusty and non-rusted foliage.

Stevenson et al., (2016) present a method for quantifying and identifying diseases in oat and sunflower leaves. The method begins with a threshold segmentation based on the studied disease (rust or blight). Groups of the generated pixels represent the diseased areas. The lesions are categorized according to their specific characteristics. The results were satisfactory; however, some inaccuracies were found due to the photos being taken at the wrong time of day. It has been proposed that the symptoms of the maize streak virus can be quantified. The authors' thresholding technique was based on Zhang et al., (2016) strategy, which was briefly discussed in the preceding paragraph. The authors evaluated visual evaluation, commercial software, and a custom system created by the authors to see which method yielded the best results. They concluded that both commercial and bespoke software packages performed similarly and that the computer-based method provided superior precision to the visual method.

Using this technique, Qin et al., (2016) found that spider mite damage to leaves may be quantified. The thresholding is done in two stages: the method's foundation. First, the leaf is distinguished from its surroundings, and then the damaged areas of the leaf are separated from the healthy surface. To arrive at a final estimate, divide the number of pixels into damaged areas by all pixels on a leaf. A leaf damage index and chlorophyll fluorescence were used to compare the results. A comparison with chlorophyll fluorescence led them to conclude that their approach and the leaf damage index yielded better findings.

To measure lesions in soybean leaves, Mohan et al., (2016) devised a method they described in their paper. Two-step thresholding is the core of the method. Leaf and background are separated using the first threshold. The HIS color space is then applied to the image that only contains the leaf, and the Sobel operator is used to define the lesion's boundaries. The final Sobel gradient image has a second threshold applied to it. Finally, the binary image is cleaned up by removing any little items and

filling in any holes left by the white pixels surrounding them. The infected areas are visible in the final products.

Researchers from Mahlein (2016) developed a method for identifying diseased areas on leaves. Bananas, maize, alfalfa, and cotton were used in the experiments. Operation 1 and Operation 2 form the basis of their algorithm. Only H and two improved variants of I3 will be applied in the following steps once a color transformation to the HSV and I1 I2 I3 spaces has been accomplished first. Then, a thresholding technique based on the healthy and harmful zones is identified. According to the scientists, their system could correctly discern between healthy and harmful zones for various circumstances and plant types.

Kavya et al., (2016) devised a method for calculating the damage done to oat crops by frost. The authors took the photos they used right there in the fields. The RGB to L*a*b* representation conversion is the initial stage in the procedure. The authors used Otsu's approach, the Isodata algorithm, and fuzzy thresholding as three different thresholding methodologies. A unique threshold value is allocated to each color network using a simple average of the threshold values generated by each technique. If necessary, the resulting divisions can be thresholded a second time until a stop condition is satisfied. Classes are created due to the partitioning, and these classes, when properly named, show just how badly the crops have been damaged.

A technique to monitor the health of vineyards was proposed by Es-saady et al., (2016). Webcams were used to capture the photographs dispersed over the area. The primary goal was to identify and count infected leaves. Their method consists of five stages: a set of morphological procedures aimed at reducing noise without removing important features; a detection step aimed at distinguishing between ground and infected leaves; and a ratio calculation. The system will warn if this ratio falls below a certain threshold.

Sugar cane leaves can be tested for fungus-related diseases using this method proposed by Deya et al., (2016). Two segmentations are used in this procedure. The first one is simple thresholding that separates the leaves from the rest of the section. Images are transformed from RGB to HIS color binarization to isolate the infected

areas in this second segmentation. The gray-scale areas of the image are applied to generate the binarisation threshold, known as the triangle thresholding method. Finally, the infection ratio on the entire leaf is calculated using the binary image.

According to Deya (2016), eelgrass leaf disease can be triggered by micro herbivory feeding, dryness, and wasting disease. There is an unsupervised classification of leaves into several categories in the algorithm's first stage (six to ten). An expert presents one of five options in the following paragraphs (the three sorts of injuries, healthy tissue, and historical context). After then, it's merely a matter of determining how much space each injury occupies. To be sure, it is an advance over prior ways of quantifying complicated leaf damage due to various stresses, according to the scientists.

Nutrient deficiencies in barley leaves were the focus of a method developed by Essady et al., (2016). They alter the RGB color channels and determine the "greenness" of the pixels using Principal Component Analysis (PCA). According to the researchers, four methods were tested to combine all pixels into a single evaluation. The goal was to concentrate on photosynthesis-active areas while minimizing non-photosynthesis-active portions, such as leaf spots. After employing non-destructive hand-held chlorophyll meters, the researchers determined that their method was well related to the widely used strategy.

According to Kavya et al., (2016), plant leaves can display five symptoms. Five separate modules form the core of their system. To determine the degree of chlorosis, an algorithm using red and green components of an image combines the leaf's yellowness to determine its degree. Blue is used to distinguish leaves from the background, and green is used to identify necrotic patches in a similar technique; a third approach uses blue to segment the leaf and determine its sphericity.

Fuzzy logic was used by Mahlein (2016) to develop a system for quantifying disease symptoms. Pomegranate leaves were used in the experiments. The algorithm first converts the L*a*b* color space. By using K-means clustering, the pixels can be divided into various categories. Authors believe one group will be associated with the afflicted areas, but they do not offer any information on how to identify the

proper group. The software determines the percentage of the infected leaf in the following example. As the last step, a Fuzzy Inference System is used to estimate the disease rating. There are also no specifics on how such a system would be implemented.

Mohan et al., (2016) used a promising technique to detect and quantify disease signs in pumpkin leaves. The study utilized flatbed scanners to collect the pictures needed for the tests. Before the images were taken, the leaves were removed from the plants, treated, and stained. The authors used built-in Mat lab routines to put their theories into practice. The algorithm begins with a thresholding step to isolate the leaf. The image is then transformed from RGB to HSV color presentation. In this example, the brightness (V) component is not included. As a final step, the pixels are grouped into two primary groups: healthy and diseased. Third-party programs, according to the authors, require too many processes to get the required outcomes. They claimed that their method is superior to third-party software.

Qin et al., (2016) studied rice crop hopper infestation and provided a strategy. The authors concentrated on the stem of the rice plant since the presence of rice planthoppers is more pronounced there. The box-counting dimension approach obtains fractal-dimension value features from the retrieved regions of interest in the procedure. A regression model is built using these characteristics. The locations are then classified into four groups using a fuzzy C-means algorithm: no disease, mild disease, moderate disease, and severe disease.

Unlike the other studies in this publication, Zhang et al., (2016) attempted to count the number of whiteflies on rose leaves to develop a whitefly early detection system. To arrive at an estimate of the number of insects, two KBS (knowledge-based systems) are used. Classification KBS interprets numerical input from image processing processes into higher-level ideas, which are then reviewed by the procedure to assist it in picking and retaining only insect-infested zones. So-called supervisory authorities. KBS chooses image processing tools and parameters for the first system, ensuring that only relevant data is fed into that system. Despite its flaws, the authors claim that their idea was a useful addition to the growing work done to automate greenhouse operations.

Maize crop disease lesions can be segmented by Stevenson et al., (2016) using six disease categories. Initially, the algorithm identifies pixels where the red channel ® is advanced compared to the green channel (G) in terms of intensity. According to the researchers, 98 percent of the time, the pixels are located in an infected area. Labels are subsequently placed on the locations that have a connection. RG is employed in the second phase of the algorithm to identify pixels that are indeed part of the lesions. The algorithm uses connected regions as seeds to better designate hazardous areas and employs a mechanism for extending regions. Otsu's approach is applied to each connected region and the threshold values it yields to determine when the growing process has ended.

Sachin (2017) used two alternative approaches to investigate powdery mildew infection in sweet cherry leaves (one visual and one computational). A flatbed scanner was used to capture the photographs. Images were analyzed using SigmaScan Pro software (v. 4.0). Thresholds were applied to each image. Fungi clusters were coated white and subjected to image analysis to get reference values to compare the two processes. The visual assessment, conferring to the scientists, generated significantly better estimates than the computational one. The signs of diseased yellow starthistle were measured by the approach proposed in 2016 by Nikolic and Tuba (2016). This software application, SigmaScan Pro (v.5.0), was used to examine the flatbed scanner images. Simple modifications to brightness and contrast, conversion to grayscale, and application of color overlays were made to the image. There are both blemishes and dark areas along venation highlighted by these overlays, therefore, shape-based selection is used to maintain only the infected areas. In the end, the number of pustules is determined.

Squash leaves afflicted with powdery mildew were assessed using visual and image processing methods, and the findings were compared. The leaf photos were segmented into five or ten classes using ArcView GIS 3.2, a commercial software tool. The images were then meticulously matched to the assigned classifications, and illness regions were accurately tagged and quantified. Finally, the illness severity was determined by dividing the whole number of pixels on the leaf by the sum of pixels that were picked. They contrasted their findings with results gathered by hand.

Their findings were also compared to the imaging device utilized to capture the photos (digital camera or scanner).

Hiteshwari and Kumar (2017) quantified foliar citrus canker in grapefruit leaves in their proposals. The researchers utilized a tool known as Assess V1.0: To perform the picture analysis, you'll need Image Analysis Software for Plant Disease Quantification. A thresholding technique separates infected areas from the rest of the scene in their method, which begins with a HIS conversion. First, the threshold was manually determined by comparing the segmentation to the actual image and modifying the value as necessary. It's merely a matter of discovering the ratio of healthy and unhealthy parts after the perfect segmentation has been accomplished. As a result of tone and lighting fluctuations, which prevent static thresholds from being effective in all settings, the authors later attempted to automate the thresholding function.

Deya et al., (2016) examined Phlox, Lily-of-the-valley, goldenrod, and apple-tree leaves using Scion Image software. A flatbed scanner was used to scan detached leaves, while a digital camera captured attached leaves. According to Es-saady et al. (2016), the color of a precise site is prudently altered using the Scion program to maximize the ability to distinguish between healthy and diseased surfaces. Rust, powdery mildew, scab, and anthracnose were the fungi studied for their symptoms.
Kavya et al., (2016) used the Assess software. The study was similar to Mahlein (2016) research in that photos were converted to HIS representations and a sufficient threshold was chosen through practical tests. As opposed to developing a unique method for illness severity measurement, the authors' major goal was to compare the accuracy of three quite diverse approaches to disease severity estimation: image processing, real-time Polymerase Chain Reaction (PCR), and ocular evaluation. According to researchers, researchers could use images and real-time PCR to improve the accuracy of CLS assessments in sugar beet while lowering the risk of biased evaluations.

Image processing was used by Mohan et al., (2016) to measure the sporulation of grapevine downy mildew. Before displaying an image to the viewer, the authors created a macro for Images that correctly changes the image's color stability and

distinction. To fine-tune the image segmentation, the operator can experiment with various threshold settings until they are happy with the results. According to the authors, a strong correlation was found between the results acquired using the authors' method and those obtained using visual assessment.

Extensions of the detection and classification methods strive to discover and label any pathology impacting the plant rather than focusing on a single illness among various diseases and symptoms (Fernandes et al., 2021). Methods for classifying data almost generally begin with segmentation, and then the extraction of various features will be fed into a classifier. Classification strategies are used to categorize the approaches described in the following section.

Qin et al., (2016) attempted to monitor plant health early. Iron, zinc, and nitrogen deficiency were all attempted to be detected using the system's lettuce leaf monitoring. The photographs were first captured with an analog video camera and then digitized. First, the photos are divided into leaf and background segments as part of the algorithm's recommended workflow. The RGB and HIS images are used to extract various size and color information in the following sections. A neural network or statistical classifier uses these parameters to figure out the state of the plant.

Zhang et al., (2016) studied three different citrus diseases and compared two methods for detecting and classifying them. The authors gathered 39 texture features and created four separate subsets of those data for two different categorization algorithms. The closest value and a Mahalanobis smallest distance classifier were used in the first method. Radial basis function (RBF) NN classifiers were trained using the backpropagation algorithm. According to the scientists, both categorization techniques performed equally well when using the best of the four types, which comprised eleven hue and capacity texture data.

For the first time, a method to identify and classify three distinctdiseases that harm Phalaenopsis orchid seedlings was proposed by Stevenson et al., (2016). A Bayesian classifier, an exponential change, a rough approximation of the diseased region's site, and then a counteraction of the sub-image positioned at that rough location are

all steps in this author's four-step segmentation procedure suggestively more sophisticated than those found in other papers. The co-occurrence matrix is then used to extract many colors and texture properties. Finally, the features are classified using an MLP ANN with a single hidden layer.

Sachin (2017) investigated six kinds of mineral absence in rice production to address this issue. The method begins by extracting a variety of texture and color information. Texture and color features are presented to separate MLP neural networks. The quantity of neurons in the "hidden layer" in each network differs (40 for texture and 70 for color). Outcomes from the networks are pooled to arrive at the final categorization. Similar methods are employed in another work by the same authors to recognize two types ofdiseases that damage rice crops (blast and brown spots).

According to Nikolic and Tuba (2016), five different plant diseases can be detected using their technology. It is unclear which plants were utilized in the testing, and the photographs were taken when they were still in the natural environment. A K-means clustering technique is used following a preprocessing step to categorize the image. There must be a link between one or more clusters and one or more diseases. The Color Co-Occurence Method is then used to extract many color and texture features for each cluster, which works with photos in the HIS format. An MLP NN does the final categorization with ten hidden layers.

Lukic et al., (2017) devised a method for detecting three distinctdiseases in maize leaves. YCbCr color space represents the original images after they have been transformed. Some rules are applied during the thresholding process to properly segment the diseased regions. While this may sound simple, it is not because of a lack of clarity. The texture information is then extracted using the gray level co-occurrence matrix. The added features are subsequently processed using a single hidden layer MLP neural network.

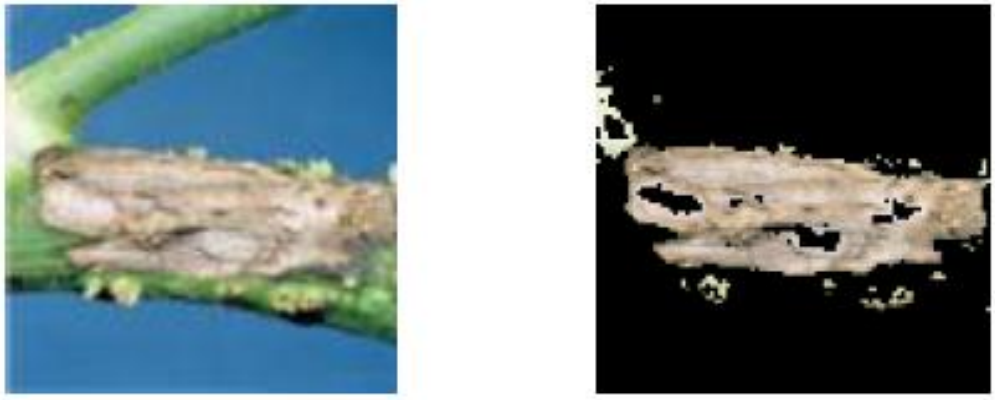
Nikolic & Tuba, (2016) evaluate several classification strategies for classifying plant leaf diseases. The k-nearest-neighbor method looks to be the most relevant and upfront of the class estimate approaches for the current test scenario. If the training

data is not linearly discrete, it is difficult to define optimal parameters in SVM, which appears to be one of its flaws (Sanjay & Nitin, 2013). Sanjay & Nitin (2013) describe the developed processing scheme as consisting of four stages, the first is the formation of a color conversion construction for the input RGB image, because RGB is used for color generation and the transformed or converted image of RGB, i.e., HSI, is used for color descriptors. Green pixels are concealed and removed in the second step using a threshold setting. Third, as the image is divided, green pixels are eliminated and masking is performed on the user segments that were initially removed, using a threshold level that has been pre-calculated. The segmentation is completed in the final or fourth main stage.

Mrunalini & Prashant (2011) describe a method for classifying and identifying the various diseases that impact plants. A machine learning-based recognition structure will immensely benefit the Indian economy because it saves time, money, and effort. This article uses the color co-occurrence approach to extract feature sets. To detect diseases in leaves automatically, neural networks are used. The proposed method can considerably help with accurate leaf detection and appears to be useful in the event of steam and root infections, while requiring less computational work. According to Arivazhagan et al., (2013), the disease detection process comprises numerous stages, including the following four essential steps: The green pixels are masked and eliminated using a threshold value, following which the segmentation algorithm is run, and texture statistics are computed to produce usable segments. Finally, the features extracted for disease classification are sent into a classifier. Experimental findings from around 500 plant leave in a database are used to demonstrate the algorithm's resiliency. Using an ANN and other image dispensation techniques, Kulkarni et al., (2021) offer a method for detecting plant ailments early and accurately. The proposed method gives better results, with a recognition rate of up to 91 percent, because it employs an ANN classifier for classification and a Gabor filter for feature extraction.

All neural networks were found to perform well by the researchers. In research by Kartikeyan & Shrivastava (2021), clustering was done using K-Means to detect cotton and ashen mold, early and late scorch, and the whiteness of rice leaves. A neural network classifier was used for classification and the results displayed were

93% accurate. In other research, images were converted to HIS before thresholding and segmenting, and the results displayed were good Figure 2.8.



a) b)

Figure 2. 8 An image of a stem with early scorch; a) original image b) clustered image (Al Bashish et al., 2010)

Table 2. 1 Summary of existing models and their recognition rates

Authors	Model	Diseases	Limitation	Disease recognition rate
Monteiro et al., (2004)	Single-layer perceptron (SLP), Multi-layer perceptron	Recommended for remote assessment of plant disease	Complex in terms of time and processing	76%
Moshou et al., (2004)	Multilayer perceptron	Yellow rust	Overfitting was observed	98%
Abdulridha et al., (2016)	Radial-basis function, Multilayer perceptron	Laurel wilt (Lw) disease	Slow	98%
Deng et al., (2018)	K-Nearest Neighbor	26 Diseases	Lacks feature scaling hence low predictions	98%
Fu et al., (2018)	K-Nearest Neighbor	10 diseases	Sluggish in learning and lacks feature scaling	98.2%
Mehonic et al., (2020)	Artificial Neural Network	26 crop diseases	Challenges during conversion	90.1%
Galata (2022)	Artificial Neural Network	20 crop diseases	Challenges during conversion	87%
Mohanty et al., (2016)	Convolutional Neural Network	26 Crop diseases	Loss of data through pooling	99.3%
Sladojevic et al., (2016)	Convolutional Neural Network	13 Crop diseases	Loss of data through pooling. The need for large datasets	96.3%
Sabour et al., (2017)	CapsNet	Cancer patients	Two convolutional layers are not sufficient for feature collection. Model complexities when data	99.6%

			increases	
Proposed model	Proposed model	Case study (Coffee)	There were no observed limitations	99.7%

2.7 Summary of the literature review and research gap identified

From the literature review, most of the work that has been done is still lacking accuracy due to pooling which leads to loss of data, insufficient feature maps due to few convolutional layers and filters, difficulty in backpropagation of error signals, computational and time complexity due to the excess number of features which sometimes others are not significant to classification. This shows that there is a need to develop a hybrid model that can ease the backpropagation of error signals, collect enough feature maps for classification, and one that is computationally cheap.

From the literature, the identified gaps can be noted as; there is limited information on using the convolutional neural network for classifying coffee leaf diseases, there is limited evidence of the applicability of the convolutional neural network for classifying coffee leaf diseases, there is a dearth of scientific information on comparing convolutional neural networks and the specialist approach for classifying coffee leaf diseases as well as the convolutional neural networks have not been validated to assess coffee diseases in Kenya.

CHAPTER THREE

MATERIALS AND METHODS

3.1 Overview

This section describes the materials and processes used in developing, testing and validating of the proposed model. The general workflow adopted was model development, image acquisition, preprocessing, extraction of features classification using the developed Hybrid Neural Network model and the output is displayed in the figure below. Figure 3.1 below is a pictorial representation of the flow.

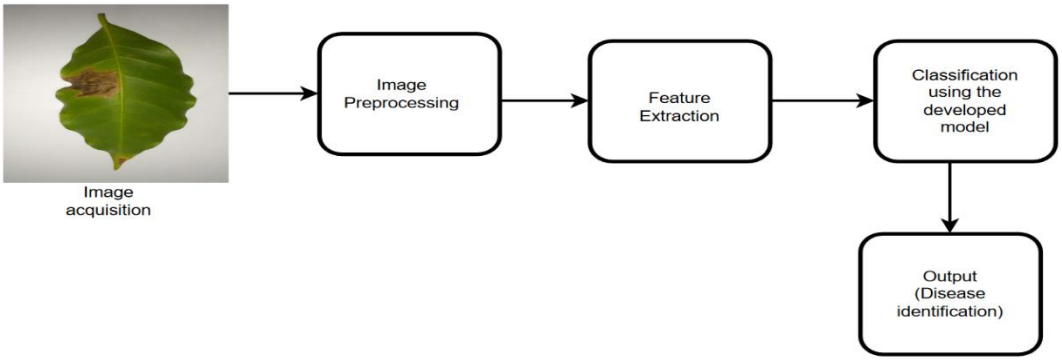


Figure 3. 1 The general Flowchart for the workflow

3.2 The proposed model architecture

This work used three subnets for convolutions and one layer primary capsule to capture diversified features where the convolutions followed the loop connectivity pattern of the DenseNets (Zhang et al., 2021). The loop connectivity pattern was used to strengthen gradient flow by making it easy to propagate signals to earlier layers more directly, contributing to the parameter and computational efficiency through feature concatenation. It helped maintain low complexity in the model by using both complex and simple features. The work used an Adam optimizer and a learning rate of 0.0001during all the processes. The language used to develop the programs was python version 3 with keras framework, and the platform was google colab with 12GB NVIDIA TESLA k80 GPU.

3.2.1 Convolutional layer

The overall architecture subdivides the convolutional layer into three subnets where convolution in each subnet is done in a dense loop format, as shown in Figure 3.2. Subnet 1 contains 128 kernels of 3x3x1 with a stride of 1 followed by the ReLU

activation function. The image goes through three levels of convolution looping and collects rich features for classification in the model.

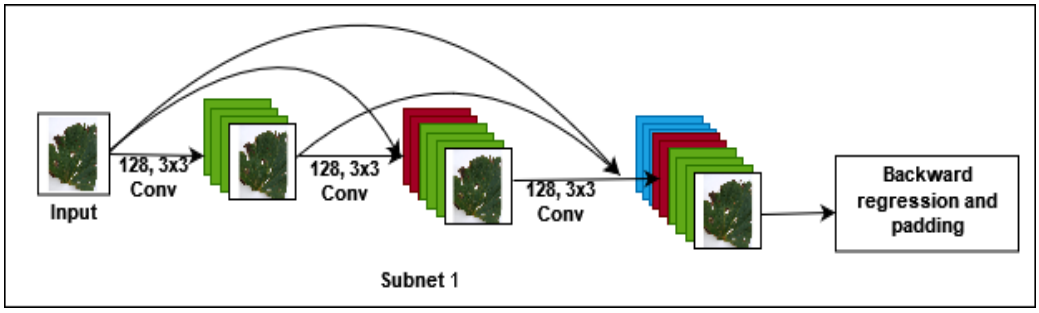


Figure 3. 2 Subnet 1

After the last convolution in the subnet, backward regression is applied to eliminate feature that are less significant to the model. In this case, aim of backward regression. In this case, backward regression aims to decrease the number of feature and reduce computational complexities without losing important features. Padding is also used to provide feature maps equal in size to those of the next layer. Equity in size also prevents complexities that may occur due to size differences and allows images to be analyzed more accurately. After backward regression and padding in subnet 1, the output is passed to subnet 2 together with the original image as the first input. The reason is to collect more rich features, which will increase the chances of better accuracy in the model. Subnet 2 contains 256 kernels of 5x5x1 with a stride of 2 followed by ReLU activation function. The image and previous output goes through three levels of convolution in a looping manner and collect rich features for classification in the model as shown in Figure 3.3.

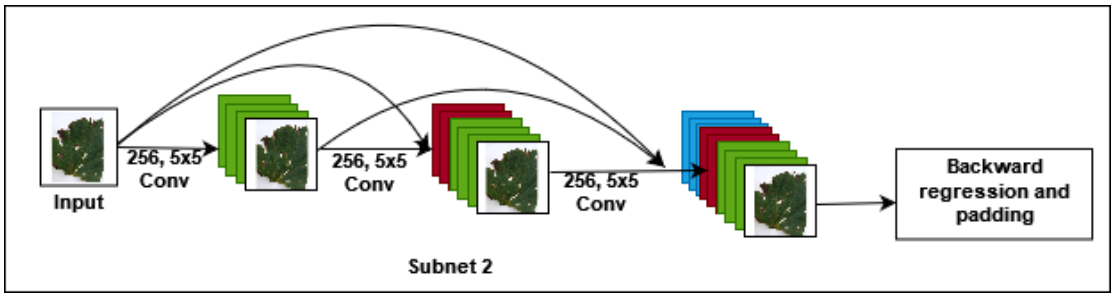


Figure 3. 3 Subnet 2

At the end of the convolution process in subnet 2, the resultant output is backward regressed to select the required features for the model and discard the rest. This

reduces computational and time complexities in the model hence better accuracy. Padding is then applied to the selected feature maps to allow more accurate analysis of feature maps through size equity in all feature maps. Subnet 3 contains 256 kernels of 9x9x1 with a stride of 2 followed by ReLU activation function. The original image and output from subnet 2 go through three levels of convolution in a looping manner and collect rich features for classification in the model as shown in Figure 3.4. Feature maps from each subnet are grouped J´egou et al., (2017).

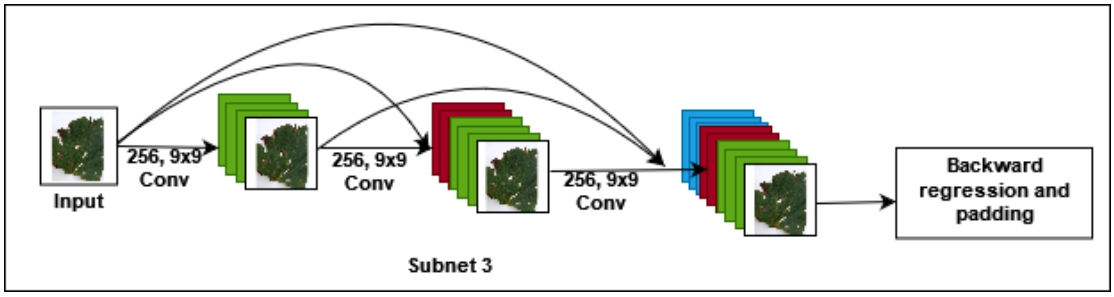


Figure 3. 4 Subnet 3

The proposed model applied backward regression after each subnet to reduce computational complexities and time taken for processing in the model.

3.2.2 Backward regression process

Backward regression with random forest classifier process.

The work used Adam optimizer with a learning rate of 0.0001 and an exponential decay of 0.98. In the work there was no observable overfitting. The total number of parameters before backward regression was 8 million while when regression was applied, the parameters reduced to 6 million. In this work results were obtained with a mean of 30 trials. Also the factor used for scaling during reconstruction was 0.392 m+ was 0.9 m- was 0.1 and lumdar was o.5. Our model displayed 11 features together with a target variable which means ten features because we did not consider the target variable for selection. In the first iteration, backward regression with the help of random forest classifier takes all the features, we fit the model and check its accuracy. In the next iteration the work used only nine features and fit the model then check the accuracy again. In each step, we check the accuracy and reduce the feature number by 1 until constant stable accuracy was reached. The work used sequential feature selector/backward selector together with random forest classifier to

implement the backward regression process. Feature selection was done on the basis of accuracy. The test size was maintained at 0.2 while the training took 0.8 percent of the data used. A tenfold cross validation was considered for the work.

In this work, y is the dependent variable. The features such as color and textureamong others are the independent variables.

Now is y=mx+c (mathematically)

In the developed hybrid model above equation was implemented as follows

$$y = \beta_o + \beta_1 x_1 + \beta_2 x_2 = \beta_3 x_3 + \cdots + \beta_{10} x_{10}$$

Random regression selects all but 1 of the existing features, fits the model and records results.

In this work the sequential feature selector/backward feature selector = (random forest classifier) helped to do the backward elimination method. In total, the work had 11 features where one of them is a target variable. The ten features include leaf shape, leaf color, leaf texture, leaf size, leaf margin, leaf venation, leaf surface, leaf tip, leaf base and leaf lobes. The target variable was in terms of binary.

$\beta_o$is a constant. **While** $\boldsymbol{\beta_1 x_1 + \beta_2 x_2 = \beta_3 x_3 + \cdots + \beta_{10} x_{10}}$ represents the full model.

Let Mp be a full model with all the 10 predictors/features (P) then the first full model before elimination was

$$\boldsymbol{y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_4 + \beta_5 x_5 + \beta_6 x_6 + \beta_7 x_7 + \beta_8 x_8 + \beta_9 x_9 + \beta_{10} x_{10}}$$

Now in each iteration one feature is dropped and the accuracy is taken then the classifier will select the best accuracy by doing MSE. The features were selected on the basis of accuracy.

The Random forest evaluated as follows:

In a random forest, the mean squared error (MSE) was calculated as follows:

1. The random forest model made predictions on a test set.
2. For each prediction, the squared difference between the predicted value and the true value was calculated.
3. The mean of these squared differences was calculated, which gives the MSE.

Mathematically, the MSE can be expressed as:

MSE = (1/n) * sum((y_pred - y_true)^2)

where n is the number of samples in the test set, y_pred is the predicted value for a sample, and y_true is the true value for that sample.

The MSE is a common evaluation metric for regression models, and it can be used to compare the performance of different models. A lower MSE indicates a better fit of the model to the data.

To calculate the accuracy of the developed hybrid model:

1. A set of predictions made by the model, was obtained by using the model to make predictions on a test set.
2. The true labels for the instances in the test set, were used to determine the number of correct and incorrect predictions made by the model.

To calculate the accuracy, the following formula was used:

Accuracy = (number of correct predictions) / (total number of predictions)

In the first case, the model was fitted with all the ten features and the accuracy was recorded. Number Equation 3.3b is the original tree before regression happened.

$$y = \beta_o + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_4 + \beta_5 x_5 + \beta_6 x_6 + \beta_7 x_7 + \beta_8 x_8 + \beta_9 x_9 + \beta_{10} x_{10}$$ Equation 3.3b

In the second case, Assuming $\beta_1 x_1$ was removed and the model fitted then the accuracy recorded. Equation 3.3c shows the resultant tree

$$y = \beta_o + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_4 + \beta_5 x_5 + \beta_6 x_6 + \beta_7 x_7 + \beta_8 x_8 + \beta_9 x_9 + \beta_{10} x_{10}$$
Equation 3.3c

In the third case $\beta_2 x_2$ was removed, the model was fitted again and the accuracy tested and recorded. Equation 3.3d shows the resultant tree

$$y = \beta_o + \beta_3 x_3 + \beta_4 x_4 + \beta_5 x_5 + \beta_6 x_6 + \beta_7 x_7 + \beta_8 x_8 + \beta_9 x_9 + \beta_{10} x_{10}$$
Equation 3.3d

In the fourth case, $\beta_3 x_3$ is removed, the model was fitted and the accuracy tested and recorded. Equation 3.3e shows the resultant tree

$$y = \beta_o + \beta_4 x_4 + \beta_5 x_5 + \beta_6 x_6 + \beta_7 x_7 + \beta_8 x_8 + \beta_9 x_9 + \beta_{10} x_{10}$$
Equation 3.3e

In the fifth case,$\beta_7 x_7$ is removed, the model was fitted and the accuracy was tested and recorded. Equation 3.3f shows the resultant tree.

$$y = \beta_o + \beta_4 x_4 + \beta_5 x_5 + \beta_6 x_6 + \beta_8 x_8 + \beta_9 x_9 + \beta_{10} x_{10} \quad \text{Equation 3.3f}$$

In the sixth case,$\beta_9 x_9$ was removed, the model was fitted and the accuracy was tested and recorded. Equation 3.3g shows the resultant tree.

$$y = \beta_o + \beta_4 x_4 + \beta_5 x_5 + \beta_6 x_6 + \beta_8 x_8 + \beta_{10} x_{10} \quad \text{Equation 3.3g}$$

In the seventh case,$\beta_{10} x_{10}$ was removed, the model was fitted and the accuracy was tested and recorded. Equation 3.3h shows the resultant tree.

$$y = \beta_o + \beta_4 x_4 + \beta_5 x_5 + \beta_6 x_6 + \beta_8 x_8 \quad \text{Equation 3.3h}$$

In the eighth case, $\beta_6 x_6$ was removed, the model was fitted and the accuracy was tested and recorded. Equation 3.3i shows the resultant tree.

$$y = \beta_o + \beta_4 x_4 + \beta_5 x_5 + \beta_8 x_8 \quad \text{Equation 3.3i}$$

In the ninth case, $\beta_4 x_4$ was removed, the model was fitted and the accuracy was tested and recorded. Equation 3.3j shows the resultant tree.

$$y = \beta_o + \beta_5 x_5 + \beta_8 x_8 \quad \text{Equation 3.3j}$$

In the last case, $\beta_5 x_5$ was removed, the model was fitted and the accuracy was tested and recorded. Euation 3.3k shows the resultant tree.

$$y = \beta_o + \beta_8 x_8 \quad \text{Equation 3.3k}$$

Then the best among all the 10 models was chosen and we called it M_{k-1}. In this work, the best accuracy was observed while from the model-tree in equation 3.3l below.

$$y = \beta_o + \beta_4 x_4 + \beta_5 x_5 + \beta_6 x_6 + \beta_8 x_8 + \beta_{10} x_{10} \quad \text{Equation 3.3l}$$

Based on the use of random forest classifier and the feature selector, the work observed the best accuracy of 99.7 when 6 features together with the target variable were used. The selected 6 features were passed to the next layer through concatenation and the action was repeated for the next two convolutional subnets. Afterward, the selected features were passed to the primary caps layer for further processing. The reduced features eased the models complexity in terms of computation. The model's shape when the accuracy was at its best was (58000, 6) which shows that the backward regression reduced the features by 4 making the model less complex in terms of computation.

3.2.3 Primary Caps, self-attention routing and the output layers.

In the next step, features from the subnets were squashed to form the primary caps layer. While in the primary caps layer, the work adopted self-attention routing to eliminate capsules that didn't contribute much to the classification process. In a single instance(i), the model accepts as input an image that may be represented as a tensor X of the shape HWF, where H, W, and C are the single input picture's height, width, and channels/features. We extract local features from the input image X using a set of convolutional and Batch Normalization layers before entering the primary caps layer. Each convolutional layer l output comprises a convolutional operation with a specific kernel dimension k, several feature mappings f, stride s=1, and ReLU as the activation function shown in equation 3.4.

$$\mathrm{F}^{\mathrm{l}+1}\left(\mathbf{X}^{\mathrm{l}}\right) = \mathrm{ReLU}\left(\mathrm{Conv}_{\mathrm{k}\times\mathrm{k}}\left(\mathbf{X}^{\mathrm{l}}\right)\right) \qquad \text{Equation 3.4}$$

Overall, the initial convolutional element of the network may be represented by a single function H_{Conv} which translates the input picture onto a higher-dimensional space to aid capsule formation. On the other hand, the second half of the network is the principal tool utilized by primary capsules to generate a vectorial representation of the characteristics they represent. It's a depth-wise separable convolution with linear activation that only conducts the first stage of a depth-wise spatial convolution operation, functioning independently on each channel. Furthermore, a kernel dimension$k \times k$ and a filter f equivalent to the output proportions $H \times W$ are imposed. Figure 3.5 shows the capsules and how attention routing occurs.

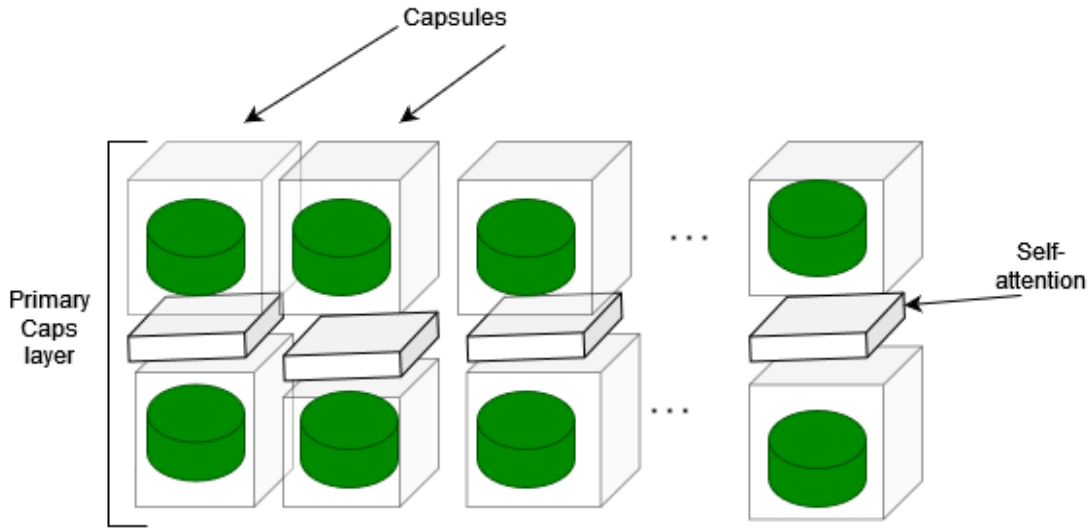


Figure 3. 5 Pictorial representation of attention routing

To produce a vectorial depiction of the characteristics they signify, primary capsules use depth-wise distinguishable convolution. The primary load of convolutional layers, on the other hand, translates the input tensor onto a higher-dimensional space, making capsule formation easier. Figure 3.6 is a mapped input image.

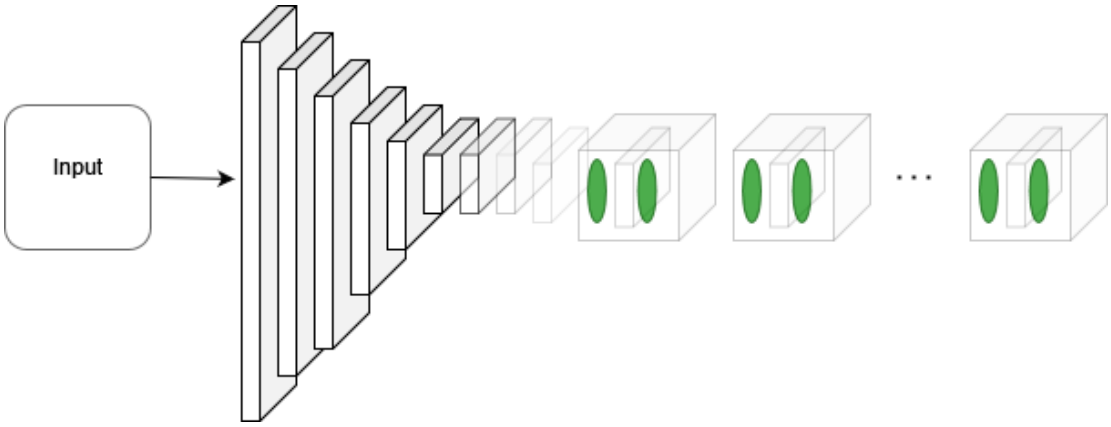


Figure 3. 6 The mapped image (input).

The input image is mapped onto a higher-dimensional space in the first section of the network, which is described as a single-function H_{Conv} . The principal capsule layer $S_{n,d}^{l}$ is then created using a depthwise separable convolution, drastically decreasing the number of parameters required to create the capsules. The primary capsule layer $S_{n,d}^{l}$ can be obtained using the H_{Conv} functions and F, where n^{l} and d^{l} are the sum of principal capsules and their discrete dimension of the l-th layer, correspondingly. The depth-wise discrete convolution is a useful procedure that considerably

streamlines and minimizes the number of parameters needed for capsule formation. We trust discriminative learning to extract all capsule attributes intelligently by utilizing its filters.

The l-th layer capsules calculate the predictions made with the weight tensor $\mathbf{W}^{l}_{n^{l},n^{l+1},d^{l},d^{l+1}}$ are added in $\hat{\mathbf{U}}^{l}_{n^{l},n^{l+1},d^{l+1}}$, which is then used in l+1. After that action, location information in the attributes of the capsules is no longer "place-coded" but rather "rate-coded." As a result, the network's base element is no longer a single neuron but a vector-output capsule. A capsule-wise activation function is the initial action applied to the principal capsule layer. The activation function must satisfy two important properties to encrypt the likelihood that a certain entity happens with the length of vectors and allow active capsules to estimate the instantiation parameters of higher-level capsules: it must preserve vector orientation and keep the dimension between zero and one. Hybrid Neural Network employs a squash operation variation of the primary initiation function as shown in Equation 3.5:

$$\text{squash}(\boldsymbol{s}^{l}_{n}) = \left(1 - \frac{1}{e^{\|s^{l}_{n}\|}}\right)\frac{\boldsymbol{s}^{l}_{n}}{\|\boldsymbol{s}^{l}_{n}\|} \qquad \text{Equation 3.5}$$

where a single capsule is referred to as s^{l}_{n}, which are the separate entries n^{l} of $S^{l}_{(n,:)}\left(s^{l}_{n_0} := \left\{S^{l}_{n,d} \mid n^{l} = n^{l}_{0}\right\}\right)$ with $s^{l}_{n} \in \mathbb{R}^{d^{l}}$.

Eq. (2)'s capsule-wise squash function meets the requisite two criteria and is significantly more sensitive to minor changes near zero, giving the gradient a boost during training 11. As a result of the squash activation, we now have a new matrix $\boldsymbol{U}^{l}_{n,d}$ with all n^{l} entries $\boldsymbol{u}^{l}_{n}$ having the same dimensionality as well as the same attributes as s^{l}_{n}, but with a "squashed" length between 0 and 1. Short vectors are shrunk to virtually zero-length, while large vectors are shrunk to a length slightly below one due to non-linearity.

Routing one's attention. We apply our self-attention routing algorithm to active route capsules to the group they belong to. Despite the extra dimension, the overall design is similar to a fully linked network with an additional branch introduced by the self-attention method, as illustrated in figure 3.6. Indeed, a capsule in the top layer's total

input, s_n^{l+1}, is a weighted sum of all "prediction vectors" from the capsules $\boldsymbol{u}_n^l$ in the layer below. For a weight matrix, a matrix multiplication of each capsule, $\boldsymbol{u}_n^l$, belonging to $\boldsymbol{U}_{n,d}^l$, is used. Intuitively, the $\mathbf{W}^l l_l^{l+1} d^l d^{l+1}$ tensor, which comprises all weight matrices, embeds all affine transformations between capsules of two neighboring layers. So, to make projections for the layer above, each capsule of layer l follows Equation 3.6:

$$\hat{\mathbf{U}}_{(n^l,n^{l+1},:)}^l = \boldsymbol{u}_n^{\mathrm{T}l} \times \mathbf{W}_{(n^l,n^{l+1},:,:)}^l \Big) \qquad \text{Equation 3.6}$$

where$\hat{\mathbf{U}}_{n^l,n^{l+1},d^{l+1}}$ holds all l-th capsule predictions. Indeed, the weight matrix predicts the attributes of all n^{l+1} capsules for each n^l capsule. Equation 3.7 was used to compute capsules of the above layer, s_n^{l+1}.

$$\boldsymbol{s}_n^{l+1} = \hat{\mathbf{U}}_{(:,n^{l+1},:)}^{\mathrm{T}l} \times \left(\boldsymbol{C}_{(:,n^{l+1})}^l + \boldsymbol{B}_{(:,n^{l+1})}^l \right) \qquad \text{Equation 3.7}$$

where$\boldsymbol{B}_{n^l,n^{l+1}}^l$ is the log priors matrix, which contains all discriminatively learned weights simultaneously as all other weights. $\boldsymbol{C}_{n^l,n^{l+1}}^l$, on the other hand, is the matrix comprising all coupling coefficients generated by the self-attention method. As a result, the prior aid in the creation of biases toward more linked capsules and the self-attention routing dynamically assigns detected forms to the entire they represent in the specific (i) instance under consideration. Starting with the self-attention tensor $\mathbf{A}_{n^l,n^l,n^{l+1}}^l$, the coupling coefficients are determined using Equation 3.8.

$$\mathbf{A}_{(:,:,n^{l+1})}^l = \frac{\hat{\mathbf{U}}_{(:,n^{l+1},:)}^l \times \hat{\mathbf{U}}_{(:,n^{l+1},:)}^{\mathrm{T}l}}{\sqrt{d^l}} \qquad \text{Equation 3.8}$$

For each capsule n^{l+1} of the layer above, there is a symmetric matrix $\mathbf{A}_{:,n^{l+1}}^l$. The term $\sqrt{d^l}$ helps maintain a balance between coupling coefficients and log priors by stabilizing training. The score agreement for any combination of the n^l capsules predictions is stored in each self-attention matrix, which may be used to determine all coupling coefficients. Equation 14 is used to calculate the final coefficients, which may then be utilized in Eq. (3.9) to obtain all capsules $S_{n,d}^{l+1}$ of the layer $l+1$.

$$\boldsymbol{C}^{l}_{(:,n^{l+1})} = \frac{\exp\left(\sum_{n^l} \mathbf{A}^{l}_{(:,n^l,n^{l+1})}\right)}{\sum_{n^{l+1}} \exp\left(\sum_{n^l} \mathbf{A}^{l}_{(:,n^l,n^{l+1})}\right)} \quad \text{Equation 3.9}$$

As a result, the coupling coefficients between a capsule in layer l and all of the capsules in layer $l+1$ add up to one. The final routing weights are obtained by adding the initial log prior probability to the coupling coefficients. The technique stays unchanged in the presence of many capsule layers piled on top to create a deeper hierarchy. Loss of margin and reconstruction regularizer. A vector rather than a scalar now represents the output layer. Indeed, a final layer capsule represents not only the probability that a given object class exists but also all of its properties retrieved from its constituent components. The length of the instantiation vector represents the likelihood that a capsule's entity exists. If and only if the thing it represents is the only one in the image, its length should be near one. As a result, we compute Equation 3.9 for each class represented by a capsule n^{L} of the last layer to allow multiple-class:

$$\mathcal{L}_{n^L} = T_{n^L}\,\mathrm{m}\quad(0, m^+ - \|\boldsymbol{u}^L_n\|)^2 + \lambda\left(1 - T_{n^L}\right)\mathrm{m}\quad(0, \|\boldsymbol{u}^L_n\| - m^-)^2 \text{Equation 3.10}$$

If the class n^L is present, T_{n^L} equals one, and $m^+, m^-$and λ are hyperparameters to be tweaked. During the training phase, the different margin losses $\mathcal{L}_{n^L}$ are then added together to determine the final score. Finally, to encourage all final capsules to encode robust and relevant attributes, we use the reconstruction regularizer as in 10. As a result, the output capsules $\{u^L_n\}_{n=1,\ldots,N}$ are given to the reconstruction decoder, and the marginal loss scaled by a factor r is added to the average L2 loss between an input image and the decoder output. Figure 3.7 shows the general architecture of the Hybrid Neural Network model.

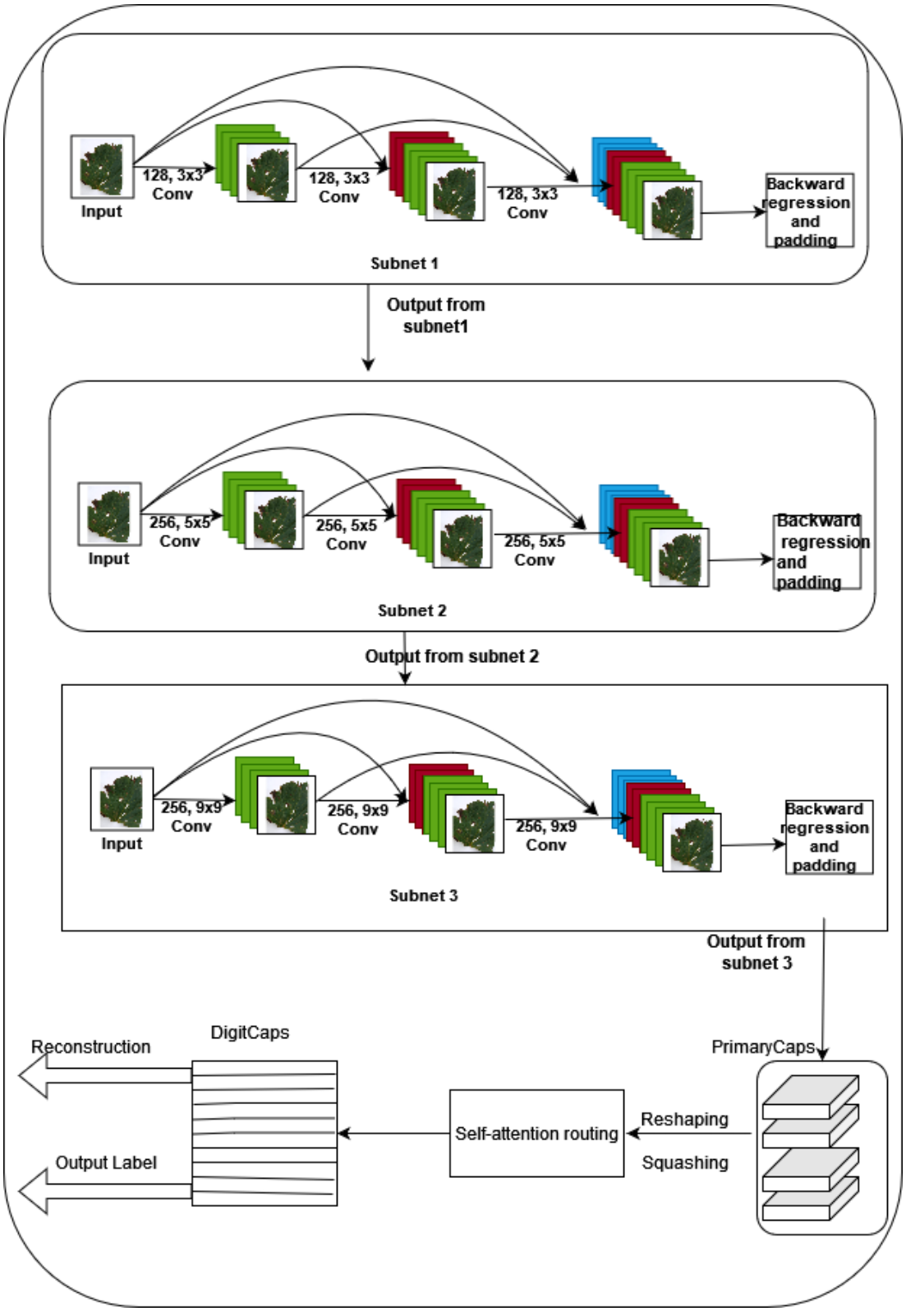


Figure 3. 7 The general architecture of the Hybrid Neural Network model

3.3 Data

This subsection includes types of data, how they were acquired, data size and the preprocessing techniques adopted. This work used two types of datasets as illustrated below;

3.3.1 Dataset (Mutira coffee plantation in Kirinyaga-Kenya)

Dataset for this work comprised 58,000 images spread across five coffee disease classes. The images were collected from the Mutira coffee plantation in Kirinyaga County-Kenya. The images from the coffee plantation were collected by taking pictures using a digital camera and with the help of a plant pathologist. A different set of healthy coffee images were collected and inoculated for validation.

The dataset was a collection of one set of healthy coffee leaves and four coffee leaf diseases as follows;

Leaf rust

A fungal disease that causes yellow, orange, or brown pustules on the leaves and stems of coffee plants. It can severely reduce crop yields and lower the quality of the coffee beans.

Coffee mosaic virus

A viral disease that causes mottled or mosaic-like patterns on the leaves of coffee plants, reducing their photosynthetic capacity and leading to reduced yields.

Leaf blight

A fungal disease that causes the leaves of coffee plants to turn yellow and die. It can lead to reduced crop yields and plant death.

Cercospora

A type of fungal disease that can affect coffee plants. It is caused by a group of fungi known as Cercospora and is characterized by the appearance of dark, circular lesions on the leaves and stems of the plant. The disease can lead to defoliation, reduced yields, and reduced plant vigor. It is typically spread by wind, water, or infected plant material, and it can be difficult to control once it has infected a coffee plant. To prevent the spread of cercospora, it is important to practice good cultural practices, such as maintaining proper soil moisture and avoiding overcrowding of plants, and to

use appropriate chemical control measures when necessary. Table 3.2 below shows coffee leaf diseases.

Table 3. 1 Coffee disease identification

class	Common name(plant)	Scientific name(plant)	Common name (leaf disease)	Scientific name(leaf disease)
C1	coffee	Coffea	Healthy	Healthy
C2	coffee	Coffea	Rust	Hemileia vastatri
C3	coffee	Coffea	Coffee mosaic virus	Caulimovirus CoMV
C4	coffee	Coffea	Leaf blight	Phoma
C5	coffee	Coffea	Cercospora	Mycosphaerella coffeicola

3.3.2 Data inoculation

In this work, 3500 healthy coffee leaves were collected from the field with a plant pathologist's help and taken to Excella biotechnology lab for inoculation. The leaves were subdivided into 4 and inoculated with four disease categories: coffee leaf rust, Cercospora, Phoma, and Coffee mosaic virus. After inoculation, a digital camera was used to take pictures of the inoculated leaves and then label and resize them. Images of each class were resized to 28 x 28 to maintain uniformity. These images were used during validation. During transportation of the healthy coffee leaves to the lab, 500 were mishandled hence were not considered for the process. The total number that were considered by the lab technologist as fit for the process were 3000 healthy leaves. From the 3000, they were divided into five groups. One group was healthy while the rest were inoculated with the diseases as rust, coffee mosaic virus, leaf blight and Cescospora. Each of the disease groups contained 600 leaves.

3.3.2.1 The inoculation process

3.3.2.1.1 Coffee mosaic virus

In order to inoculate a healthy leaf with coffee mosaic virus in the laboratory, the following steps were followed:

Step1: Healthy coffee plant leaves were sterilized by dipping in a solution of 70% ethanol for 30 seconds, followed by a rinse in sterile water.

Step 2: A sample of coffee mosaic virus was obtained through a laboratory culture.

Step 3: The leaves were inoculated by applying a small amount of the virus to the surface of the leaves using a sterile needle.

Step 4: The leaves were then incubated in a humid environment, in this case a petri dish with a moist paper towel was used, for several days to allow the virus to infect the leaves.

Step 5: The leaves were then monitored for symptoms of infection, such as mosaic patterns or chlorosis (yellowing of the leaf).

3.3.2.1.2 Coffee leaf blight

In order to inoculate a healthy leaves with leaf blight in the laboratory, the following steps were followed:

Step one: The obtained healthy coffee leaves were sterilized it by dipping it in a solution of 70% ethanol for 30 seconds, followed by a rinse in sterile water.

Step two: Samples of leaf blight pathogens were obtained through a laboratory culture.

Step three: The leaves were inoculated applying a small amount of the pathogen to the surface of the leaves using a sterile needle.

Step four: Then the leaves were incubated in a humid environment. In this case a petri dish with a moist paper towel was used. This step waited for several days to allow the pathogen to infect the leaf.

Step five: The leaves were then monitored for symptoms of infection, such as necrosis (dead tissue) or chlorosis (yellowing of the leaf).

3.3.2.1.3 Leaf rust

In order to inoculate a healthy leaves with coffee leaf rust in the laboratory, the following steps were followed:

Step one: Healthy coffee leaves were sterilized by dipping it in a solution of 70% ethanol for 30 seconds, followed by a rinse in sterile water.

Step two: A sample of coffee leaf rust fungus was obtained through a laboratory culture.

Step three: The leaves were inoculated by applying a small amount of the fungus to the surface using a sterile needle.

Step four: The inoculated leaves were incubated the leaf in a petri dish with a moist paper towel, for several days to allow the fungus to infect the leaf.

Step five: The leaves were monitored for symptoms of infection, such as the presence of pustules (small, raised bumps) containing spores.

3.3.2.1.4 Cercospora

In order to inoculate a healthy leaf with Cercospora in the laboratory, the following steps were considered:

Step one: Obtained healthy coffee plant leaves were sterilized by dipping it in a solution of 70% ethanol for 30 seconds, followed by a rinse in sterile water.

Step two: A sample of Cercospora fungus was obtained through a laboratory culture.

Step three: The leaves were inoculated by applying a small amount of the fungus to the surface of the leaf using a sterile needle.

Step four: The leaves were incubated in a petri dish with a moist paper towel, for several days to allow the fungus to infect the leaf.

Step five: The leaves were then monitored for symptoms of infection, such as the presence of dark brown or black lesions on the leaf surface.

After inoculation, the images of the infected leaves were taken using a digital camera and with the help of a plant pathologist and a lab technician from excella lab. The images were then taken to be used for validation purposes.

3.3.3 Data Pre-processing

3.3.3.1 Noise filtering

Image misrepresentations and sounds must be removed during pre-processing to increase image quality. Noise filtering and contrast stretching, both done with mean and median filters, were used during data pre-processing. The researchers utilized Gaussian and averaging filters for mean filtering. The averaging filters were employed to remove grain noise from the images, reducing local variations produced by grains. The filters applied masks to each pixel in the signal, forcing each ingredient to be averaged together and forming an exclusive pixel (Caglayan & Can, 2018).

3.3.3.2 Cropping

Then individual images in the dataset were analysed to check if they all had the same squared shape. Images with a non-squared form were collected to make the image's centre square. The dimensions of each image were also examined to ensure they were all the same size. The undesired sections of the photographs were cropped out using a crop tool. Figures 3.8 show a cropped image before and after. Cropping the leaf helps to highlight the region of interest. This benefits deep learning researchers who seek to cut image processing training time.

Figure 3. 8 An image of a leaf afflicted by coffee Leaf Rust Before (a) and after (b) cropping

3.3.3.3 Data augmentation

Data augmentation was used on the images taken in the field to improve smaller datasets by translating them into larger ones. Rotation and flipping were used as data augmentation techniques in this study. The rotation was done at 180 degrees horizontally and vertically to accentuate the disease-affected region of concern. The image rotator tool was used to rotate the image counter-clockwise. An image flipper was utilized in the instance of flipping. A horizontal flipper was used to flip some images horizontally, while a vertical flipper was used to flip others vertically. The photos were flipped to allow for a better display of the infected areas. An example image before and after rotation is shown in Figure 3.9.

Figure 3. 9 Image before (a) and after (b) rotation

After preprocessing the images, they were transferred to the convolutional layer for feature extraction. Backward regression was then applied to eliminate features that were not significant to the model. As soon as required features remained, they were transferred to the primary caps layer, where they were subjected to attention routing to get capsules that were useful for classification and discard the rest.

3.3.4 Comparison with other models

Only one model is insufficient to determine and report the findings on whether there is a significant improvement in accuracy and general processing in image processing. Therefore, this work considered other neural network models. After testing the developed Hybrid Neural Network model, the same images were classified using other models namely LeNet, AlexNet, ResNet, Artificial Neural Network (ANN) and

the conventional Capsule neural network model (CapsNet) for comparison purposes. All F1 accuracies in all the models were recorded for comparison purposes.

3.3.5 Validation

To validate the model the following two cases were considered:

3.3.5.1 Case 1

Infected coffee images from the field were collected using a digital camera and labelled with the help of a plant pathologist. In this case, the pathologist and the researchers toured the coffee plantation. As the pathologist identified a leaf infected with a particular disease, images were taken using a digital camera. Images of the five coffee diseases were collected and tested in the model to compare the results received from the model with those from the plant pathologist. The model was able to give correct results.

3.3.5.2 Case 2

In the second case, the images that had resulted from inoculation were used to test whether the model could recognize diseases as expected. Here coffee leaf rust was considered for the test whereby images known to be affected by coffee rust class were fed into the model for classification. Results from the Hybrid Neural network model were able to match those from inoculation.

CHAPTER FOUR

RESULTS AND DISCUSSION

4.1 Overview

This section displays the findings and discussion. The section is organized as hybrid neural network for identifying coffee leaf diseases, identifying coffee leaf diseases using the developed model, comparing the developed Hybrid Neural Network model with the conventional capsule neural network model, comparing Hybrid Neural Network model with other models and validating the Hybrid Neural Network model.

4.2 Hybrid neural network for identifying coffee leaf diseases

Neural Networks generally are based on parameters like momentum, batch size, learning rate, dropout, and learning rate decay (Ferentinos, 2018). Because neural networks deal with datasets of the same size (Ferentinos, 2018), this study tuned the dataset parameters by modifying the input images to 28×28. The first case analyzed plant leaf disease classification results using the developed Hybrid Neural Network (HYBRID NEURAL NETWORK MODEL) model. F1 scores and graphical representations were used to find the accuracy of each model. F1 was computed using Equation 4.1.

$$2 \times \left(\frac{precision \times recall}{precision + recall}\right) \qquad \text{Equation 4.1}$$

Table 4.1 summarizes the disease identification accuracy based on the F1-score.

Table 4.1F1-score, precision, support and recall metrics for Coffee dataset (Regressed)

CLASS	**P1**	**R1**	**P2**	**R2**	**F1 (A)**	**F1 (B)**	**Support**
Rust	0.97	1.00	0.86	0.9	0.98	0.89	200
Leaf blight	0.95	0.98	0.86	0.89	0.97	0.87	252
Healthy	0.98	1.00	0.88	0.90	0.99	0.88	160
Cercospora	0.98	0.97	0.89	0.88	0.97	0.89	388
Coffee mosaic virus	0.96	0.91	0.85	0.86	0.93	0.84	180

P1 Precision with the hybrid neural network model

R1 Recall with the hybrid neural network model

P2 Precision with the conventional CapsNet model

R2 Recall with the conventional CapsNet model

F1 (A) F1 score with the hybrid neural network model

F1 (B) F1 score the conventional CapsNet model

Figures 4.1 (a) and (b) display the testing and training accuracy and loss obtained from the hybrid neural network model, while Figures 4.2 (a) and (b) show the accuracies obtained from the conventional CapsNet model. Regarding overall percentage accuracy, the developed hybrid model has a 99.7% F1 Score, while the conventional CapsNet showed an 87% accuracy F1 score.

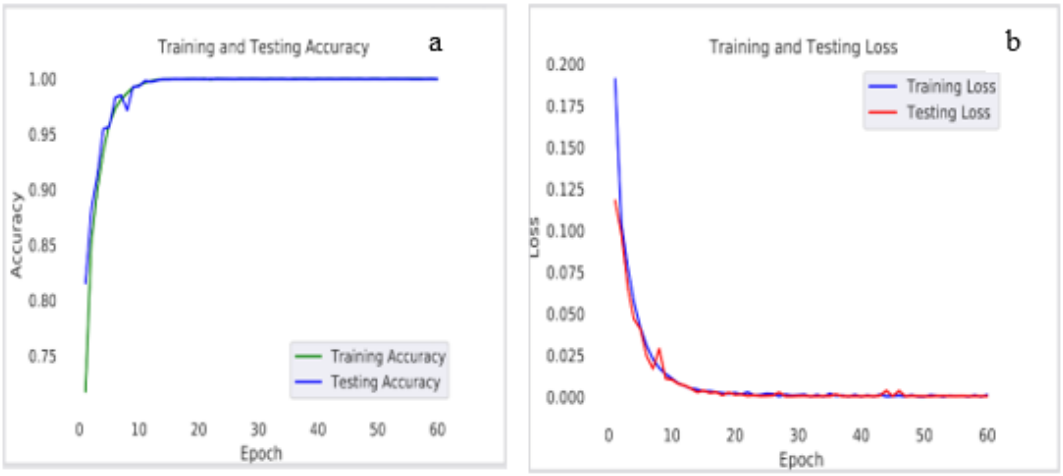


Figure 4. 1 Percentage accuracy (a) and loss (b) of the hybrid neural network model

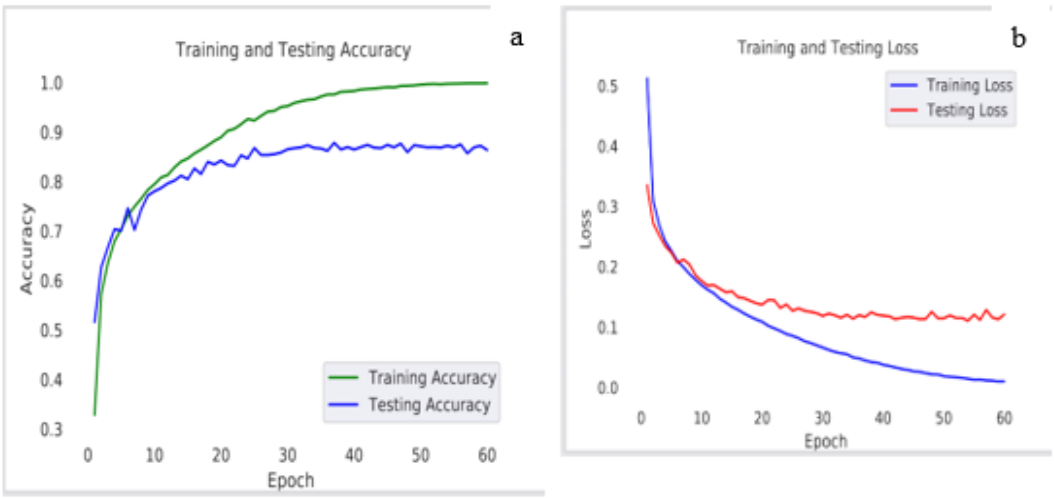


Figure 4. 2 Percentage accuracy (a) and loss (b) of conventional CapsNet

According to Kwabena et al. (2020), the Gabor and Capsule networks can be used to distinguish foggy, distorted, and previously unknown tomato and citrus disease images. The test accuracy of the proposed model is 98.13 percent. According to the researchers, the technique might be used on different crops and could be a useful tool for detecting unseen plant ailments in poor weather and lighting conditions. Adusei et al., (2021) used the squared Euclidean distance, sigmoid function, and a 'simple-squash' function instead of the dot product, SoftMax normalizer, and squashing function that were used in the dynamic routing technique. Extensive testing on the datasets revealed that the developed model enhances test accuracy while allowing for an increase in the number of routing iterations without affecting performance.

This work presented the developed Hybrid Neural Network model for detecting coffee plant leaf disease. For all of the models, the datasets were split 8:2 for training and testing safe for the sensitivity analysis in which the work compared different ratios. The margin and reconstruction losses make up the loss function used to train the model. The loss function's default settings for m+, m-, were kept in this implementation. All models were subjected to random changes in parameter values and intermediate layers to evaluate their sensitivity to the changes. The other models performed worse than the hybrid neural network model model due to these adjustments. Hybrid neural network model's performance was unaffected by changes in momentum, batch size, learning rate, dropout, or learning rate decay. The single most critical hyperparameter that defined the developed models' performance was the number of routing iterations, with three yielding some performance levels. This work recommends the adoption of DenseNet connectivity loops to strengthen the gradient, maintain low feature complexity, and provide computational and parameter efficiency. Research by Zhang et al., (2021b) demonstrated that DenseNet connectivity provides more accurate results because it is easier to propagate error signals directly to earlier layers. Hybrid neural network model also used backward regression to decrease the number of parameters and ease computation in the model. Unlike pooling, which discards any data (Hinton et al., (2017), backward regression only selects data with vital information. This idea improved the accuracy of this model, enhanced computational efficiency, and reduced complexities related to time consumption.

4.3 The model performance on identification of leaf diseases during backward regression process for feature identification

For feature identification and selection the work used the coffee dataset comprising of 58000 coffee leaf images covering five coffee disease categories. In the case of feature selection using random forest classifier, feature selection was done with the help of a feature selector to determine which features to adopt for the work. In the first case we fit our model with all the features (ten + target variable) and observed an accuracy of 98.7%, In the next iteration with nine features and a target variable , an accuracy of 98.9% was observed, In the next iteration with eight features and a target variable , an accuracy of 99.2% was observed, In the next iteration with seven features and a target variable , an accuracy of 99.50% was observed, In the next iteration with six features and a target variable , an accuracy of 99.7% was observed, In the next iteration with five features and a target variable , an accuracy of 99.64% was observed , in the next iteration with four features and a target variable , an accuracy of 99.5% was observed, in the next iteration with three features and a target variable , an accuracy of 99.43% was observed, in the next iteration with two features and a target variable , an accuracy of 99.21% was observed, and In the last iteration with one feature and a target variable , an accuracy of 98.6% was observed. In the end overall the work observed that the best accuracy was realized when six features were considered. In that case the work used only six feaures with the target variable instead of the total eleven features. Due to the reduced number of features, model computation also reduced. This was also evident from the results of the test error. The overall test error for the developed hybrid neural network model was 0.16 while the test error for the conventional CapsNet model was 0.26.

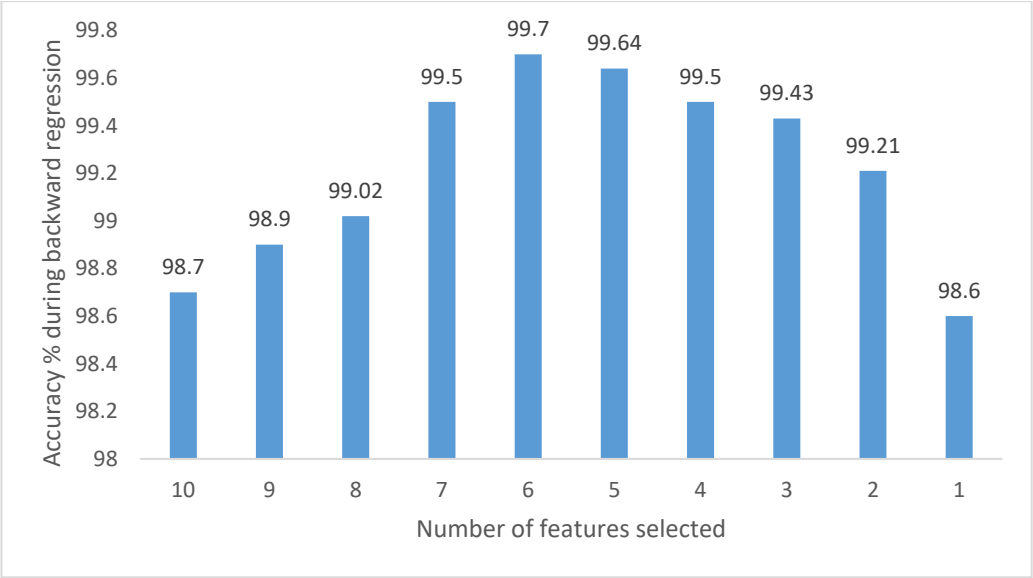


Figure 4. 3 Accuracy observed during backward regression

The developed hybrid model was used to identify leaf diseases in coffee. In this case, diseased leaves were collected from the fields and labeled then the model was used to classify them. During testing, 5 coffee categories were considered, and the results were observed as shown in Table 4.3 coffee images from the field classified with the model.

Table 4. 2 Model's identification of coffee leaf diseases

Categories	Class names	Class size	Correct classification	Incorrect classification
C1	Healthy	100	99	1
C2	Coffee mosaic virus	120	119	1
C3	Cescospora	150	142	8
C4	Leaf-rust	130	130	0
C5	Leaf blight	80	76	4

During image collection, we used the camera without adding extra lighting in many images, while in a few images, we added lighting to the camera. The reason for the addition was to determine whether lighting affects recognition in machine learning. From the results, it was noted that those images with increased lighting were not correctly classified as the model had not been trained in such conditions. This was also confirmed in work by Ferentinos (2017). They said illuminations affect

recognition in image processing, and the machine should first be trained on illuminated images if there is a need to classify them. The other images without extra lighting were classified correctly.

This work generally used coffee leaf disease database containing 58000 images spread around 5 disease categories. Datasets displayed promising results when the hybrid neural network model was used. Compared to Ferentinos (2018) study, this work achieved good accuracy. Ferentinos (2018) used five CNN architectures: AlexNetOWTBn, VGG, GoogLeNet, AlexNet, and Over feat. Each of the original images' architectures had a success rate of 99.44%, 99.48%, 97.27%, 99.06%, and 98.96%, respectively. The success rate percentages of the pre-processed images were 99.07% (AlexNetOWTBn), 98.87% (VGG), 97.06% (GoogLeNet), 98.64% (AlexNet) and 98.26 (Overfeat). Chaki and Parekh (2021) support these results, who argued that plant leaf image detection accuracy and success rate should normally fall between 90% and 100%. This means neural networks effectively use the leaf feature for plant disease classifications. The best prediction model for this study's architecture achieved 99.7% at its highest testing accuracy and 100% on training accuracy. The results did not contradict the recommended accuracy rates by Chaki and Parekh (2021) and Ferentinos (2018), whose accuracy and success rate were above 90%. These results were similar to the accuracy rates achieved by Ferentinos (2018), who used a larger database of 87,848 images comprising 25 different plant species with 58 plant diseases and healthy plant classes. Dilaghi et al. (2019) tested the neuron network using a larger images database and recorded a high validation accuracy of 98.9%.

The results from this work are similarly supported by other studies (Ferentinos, 2018; Dilaghi et al., 2019), which have shown that a trained deep learning model is more efficient when the input data is similar to the training data. When the input data contrasts with the training data, the results will show a reduced value as accuracy rate. Such effects on the classification accuracies of a training dataset have been highlighted by Mohanty et al., (2016), who used an extensive database of 54,306 color images to detect plant diseases. The study used automated image recognition and processing systems such as high-definition cameras and smartphones. From the 14 plant species and 26 disease classifications that were inbuilt into the model, the results showed a 99.35% accuracy rate. Other studies that equally applied CNN

image processing to detect plants diseases included a study by Kaur et al., (2018) that used computer vision enhancements and image processing and a study by Chaki and Parekh (2011), who suggested the use of an automated system for rapid recognition of plant species using leaf images. Sladojevic et al. (2016) also used a CNN model, which used a large plant dataset comprising 13 plant diseases and achieved a 96.3% accuracy rate.

A study by Kamilaris and Prenafeta- Boldú (2018) created a CNN model using many image datasets and established that a high accuracy rate could be achieved using a large database during CNN training. A large dataset is essential regardless of the model's augmentation techniques and transfer learning systems (Kamilaris & Prenafeta- Boldú, 2018).

Some models have demonstrated that CapsNet architectures outperform complicated CNN architectures (Ferentinos, 2018). Such applications are evident from Naikwadi and Niket (2013), who used a simple method to detect plant diseases. Hinton et al., (2018) and Barbedo (2018) outline other effective measures for solving classification problems. They noted that Extended Capsule Neural Network architecture (E-CapsNet) performs better than the conventional CapsNet. Al-Hiary et al., (2011) used the ANNs method to automatically recognize and categorize plant diseases using leaf images. The classification was based on the K-means with ten hidden layers and six output classes. The results showed a classification accuracy of 94.67%. These results lie within the accuracy rates suggested by Chaki and Parekh (2011). Future works must focus on architecture optimization and models utilizing real-time smartphone models for plant disease classification.

4.4 Comparison between the developed hybrid neural network model with the traditional convolutional neural network models

This research also explored the other models experimenting with plant leaf disease classification. Among the models considered for further experiments were; Artificial Neural Networks (ANN), GoogleNet, ResNet, Capsule Neural Network (CapsNet), SVM, AlexNet, Inception V3, VGG 16, and the results were recorded results as shown in Table 4.3.

Table 4. 3 General F1 Testing accuracies on different models

Model name	% Testing accuracy (F1) On PlantVillage images	% Testing accuracy (F1) On Coffee images collected from the field
Artificial Neural Networks (ANN)	88.6	90.4
GoogleNet	99.3	98.3
ResNet	99.1	99.0
CapsNet	87.4	87.4
Hybrid Neural Network	99.7	99.8
AlexNet	99.3	99.3
SVM	84.5	85
Inception V3	99.14	99.2
VGG 16	98.2	98.2

Table 4.4 compares the sensitivity of the developed Hybrid Neural Network model to the conventional CapsNet. From the displayed results, Hybrid Neural Network model is more sensitive than the conventional CapsNet.

On the coffee dataset, the hybrid Neural Network outdid baseline CapsNet by 8.37 percent, with an overall test accuracy of 99.70%, equivalent to state-of-the-art models on the same datasets. Samin et al., (2021) developed a deep learning architecture model (CapPlant) to assess whether a plant is healthy or infected using plant pictures. The architecture automatically pulls representations from the input data series, eliminating the requirement for handcrafted features in the prediction operation. To extract and categorize data, many convolutional layers are used. The last convolutional layer in CapPlant is replaced with a front-line capsule layer that contains orientational and relative spatial relationships between distinct components of a plant in an image to improve prediction of infected plants. The proposed approach is evaluated using the PlantVillage dataset. The CapPlant model outperformed previous plant disease classification models regarding prediction accuracy. The testing results of the model had an overall test precision of 93.01 percent and an F1 score (93.07 percent). We compared our work to that done with the state-of-the-art CapsNet and found that ours was more accurate by a higher

proportion. This means that it performs better when CapsNet is improved by regression and the usage of dense connection loops.

Research by Kurup et al., 2019 used a capsule neural network model to diagnose plant diseases. They used a dataset size of 54,306 images and obtained an accuracy of 94%. Verma et al., 2020 used transfer learning to create capsule networks to classify potatodiseases and compared their performance to a few famous pre-trained CNN models, notably ResNet18, VGG16, and GoogLeNet. Colored images of healthy and infected leaves from the PlantVillage dataset were utilized for training the models. With 91.83 percent accuracy, CapsNet demonstrated comparable performance to state-of-the-art CNN models. Research by Kwabena et al., 2020 suggests applying the Gabor and Capsule networks to distinguish hazy, distorted, and previously unknown tomato and citrus illness images. The suggested model achieves a test accuracy of 98.13 percent. According to the researchers, the technique may be applied to other crops and might be a valuable tool for detecting invisible plantdiseases under poor weather and lighting circumstances. Adusei et al., 2021 used the squared Euclidean distance, sigmoid function, and a simple squash function instead of the dot product, SoftMax normalizer, and squashing function present in the dynamic routing method. Extensive trials on the datasets revealed that the Hybrid Neural Network model model improves test accuracy consistently across the datasets while also allowing for an increase in the number of routing iterations with no performance impact. The suggested model beat a baseline CapsNet by 8.37 percent, with an overall test accuracy of 99.70 percent, equivalent to state-of-the-art models on the same datasets.

Samin et al., 2021 built a deep learning architecture model (CapPlant) that uses plant photos to detect whether it is healthy or infected. The prediction procedure does not need handmade features; rather, the architecture automatically extracts representations from the incoming data series. To extract and categorize features, many convolutional layers are used. The last convolutional layer in CapPlant is replaced with a cutting-edge capsule layer that incorporates orientational and relative spatial relationships between distinct components of a plant in an image to more precisely forecastdiseases. The suggested architecture is validated using the PlantVillage dataset, which includes over 50,000 images of healthy and diseased. Compared to existing plant disease classification models, the CapPlant model

showed significant gains in prediction accuracy. The generated model's testing findings obtained an overall test accuracy of 93.01 percent, with an F1 score of 93.07 percent. The developed Hybrid Neural Network model was compared to those of other models, including using the state-of-the-art CapsNet, and observed that the developed Hybrid Neural Network model displayed a higher accuracy percentage. This means that neural networks can perform better when improved through regression, attention routing and dense connectivity loops.

Table 4. 4 sensitivity analysis using the developed model with and without feature selection

Testing Training (%)	**F1- score for conventional CapsNet**	**F1 – Score for the Hybrid Neural Network model**
10-90	0.96	0.99
20-80	0.97	0.98
30-70	0.97	0.98
40-60	0.97	0.97
50-50	0.95	0.96
60-40	0.96	0.96
70-30	0.95	0.95
80-20	0.94	0.94
90-10	0.92	0.94

This work also evaluated other models, as shown in table 4.5. In this case, the F1 results of all the models were recorded. Among other evaluated models were; SVM, Artificial Neural Networks (ANN), AlexNet, ResNet, VGG 16, GoogleNet, and InceptionNet V3. This work used a data ratio of 20:80 Kurup et al., (2019) testing and training.

Table 4. 5 Comparison deep learning models are compared using the F1-score percentage accuracy measure

Model	**%Testing Accuracy**	**%Training Accuracy**
SVM	84.5	99.0
ANN	88.6	100
AlexNet	99.3	100
ResNet	97.85	98.5
Inception v3	99.14	100
VGG 16	98.2	99.0
CapsNet	87.8	100
Hybrid Neural Network	99.7	100

4.5 The model validation

The first method is where the plant pathologist and the researchers went to the coffee plantation, identified diseased leaves, and took pictures using a digital camera. The images were subjected to the model, and results were observed and compared with the identifications from the plant pathologist, as shown in Table 4.6.

Table 4. 6 The model validation. Disease identification according to a pathologist versus identification according to Hybrid Neural Network

By Pathologist	Hybrid Neural Network
Leaf rust	Leaf rust
Leaf_blight	Leaf_blight
Cerscospora	Cerscospora
Coffee mosaic virus	Coffee Mosaic virus
Coffee Healthy	Coffee Healthy

The second method is where the inoculated images were compared to the output from the model. In this case, images of inoculated leaves were taken using a digital camera and labeled. Then the images were subjected to the developed hybrid neural network model for disease identification. Results were observed and recorded as shown in table 4.7.

Table 4. 7 The model identification of diseases. Identification according to inoculation versus identification according to the Hybrid Neural Network model.

Inoculated images	Hybrid Neural network model
Leaf Rust	Leaf Rust
Leaf blight	Leaf blight
Cescospora	Cescospora
Coffee Mosaic virus	Coffee Mosaic virus

The model was used to test a sample image after training; the results are shown in Figure 4.4.

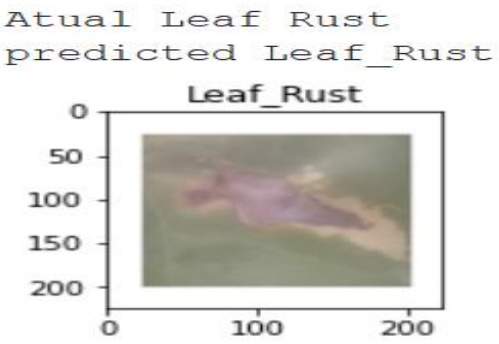


Figure 4. 4 Pictorial representation of some of the testing results from the Hybrid Neural Network model

This validation showed that the model was able to accurately identify diseases and that its performance was on par with that of a trained pathologist. This was encouraging as it suggested that the model could potentially be used as a tool to assist pathologists in their work, helping to improve the efficiency and accuracy of coffee disease diagnosis.

CHAPTER FIVE

CONCLUSION, RECOMMENDATION AND AREA FOR FURTHER WORK

5.0 Overview

This section covers the conclusion, recommendations and areas for further work. The conclusion for each objective is presented. The section provides recommendations based on the finding from each objective. Finally, areas of further research are highlighted.

5.1 Summary

The model was used to detect and classify diseases whereby it was able to classify correctly. From the results obtained, it was observed that results were good, the model was able to generalize well and there were no signs of data loss. Also, the model was not tested on PlantVillage data alone, it was tested on data from the field, and good results were observed.

5.2 Conclusion

From the findings and results in this work, it is concluded that adopting DenseNet intuition based on loop connectivity patterns in each subnet promotes strong gradient flow because error signals can be easily propagated to the earlier layers in the subnet more directly. Further, since each layer receives signals from all its proceeding layers, all inputs are combined by channel-wise concatenation, making classification more efficient. Due to the looping process, concatenating features from all preceding layers allows the model to be more diversified hence richer patterns and more accurate classification. Because the model can use features from all complexity levels in the network, this enables the use of both simple and complex features, giving high generalization results. The adoption of subnets by the Hybrid Neural Network model was observed to be good in rich feature collection. In this case, both simple and complex features were collected in totality, hence the model's ability to generalize well since it gets all the required features. It was concluded that increased convolutional layers were better, unlike the normal two layers that may not collect enough features for classification. The self-attention routing used in this work can compare the capsules and select the most appropriate for classification. The use of attention routing greatly reduces the computational complexities in the model, unlike

the commonly used dynamic routing. It was concluded that self-attention routing performs well and could be the direction to follow now and in the future. The accuracy of plant disease detection may be enhanced even further with the fusion of several classifiers and the integration of the developed model capsule neural network.

Backward regression used in each convolutional subnet layer reduced computational and time complexities – egression, pooling, and down sampling work to reduce the complexities associated with computation. However, in the case of backward regression, no significant data is lost because regression works by selecting and sorting. It looks at the features, determines which does not contribute significantly to work, and then discards it. Therefore, data that is not useful to the model is discarded, reducing complexities without affecting the general classification accuracies of the model. The work used only five features with the target variable instead of the total ten features. Due to the reduced number of features, model computation also reduced. This was also evident from the results of the test error. The overall test error for the developed Hybrid Neural Network model was 0.16 while the test error for the original CapsNet model was 0.26. It was concluded that using backward regression is better than pooling since pooling leads to significant data loss. The developed model was used for plant leaf disease detection and could give reliable results on all datasets. There were no misclassifications and losses observed. Coffee boards can utilize the developed Hybrid Neural Network model for farmers in large and small-scale cultivation without having to travel looking for crop pathologists. It was concluded that the model was able to classify and could be adopted for use.

In comparing the developed Hybrid Neural Network model to other deep learning models, the developed Hybrid Neural Network model displayed higher performance. The developed Hybrid Neural Network model displayed 99.7 % F1 score. In comparison, the normal CapsNet model displayed 87 % F1 score accuracy on testing our framework based on the standard PlantVillage (PV) dataset comprising ten tomato classes with 9080 images and the coffee dataset that was created, respectively. The developed model showed relatively higher and stable accuracy when sensitivity analysis was performed by varying testing and training dataset

percentages. Also in comparison Support Vector Machines (SVM), AlexNet, ResNet, VGGNet, Inception V3, Artificial Neural Networks (ANN), and VGG 16 deep learning approaches scored 84.5\%, 88.6\%, 99.3\%, 97.87\%, 99.14\%, and 98.2\%, respectively. These findings indicate that the developed Hybrid Neural Network model may be a decent and, in most cases superior, and less expensive alternative for phrase categorization models founded on convolutional neural networks (CNNs) and RNNs.

In validation, the model was tested using field images collected and labeled with a plant pathologist's help. The model displayed the results as the pathologist had reported them. The results showed that the developed Hybrid Neural Network model could give reliable results and hence can be adopted by farmers. The model was also tested using the inoculated images and the results were correct. The work concluded that the model could classify in all situations whether the data was from the field or laboratories.

5.1 Recommendation

This work recommends that researchers consider using backward regression because this technique can reduce computational complexities. Pooling has been used for the same purpose, but this work does not recommend its use because pooling leads to the loss of significant data hence low accuracy. Here backward regression has been recommended because it removes features that do not contribute significantly to the accuracy of the work.

This work also recommends using subnets with several convolutional layers to enable the collection of rich features hence better classification. Using only two convolutional layers, as in the case of the conventional Capsule neural network (CapsNet), is not enough to collect rich features for better processing and classification. This work also recommends the use of different kernel sizes to be able to collect both simple and complex features for classification. Further, the work recommends using dense connectivity loops to ease the back propagation of errors. Finally, the work recommends using self-attention routing instead of the normal dynamic routing because self-attention routing can select the most useful capsules for further classification. This reduces computational and time complexities in the model.

This work also recommends that a mobile application can be developed by researchers and the model hosted in the clouds so that farmers and other stakeholders in the coffee industry and other plant industries can use it to detect and classify diseases. This will improve the yield since the crop diseases will be identified and treated within a short time and at a lower cost. We recommend that the model be extended to agrovets to enable them to respond to farmers with information on the availability of drugs for particular diseases and their cost. In this case, the farmers and the agrovet personnel can even agree on terms of payment and delivery of the drugs. This works also recommends using different image effects while training and training the model to enable machines to learn all types of illuminations in images. This will increase accuracy during classification because the models can recognize different illuminations.

This work recommends that different models be tested using the diseased leaf images to help decide which model is suitable for classifications under a given condition. This work also recommends the fusion of different models to maximize efficiency and improve results. On validation, this work recommends the use of more than one validation method to be sure of the results. This work used the pathologist and inoculation for validation.

5.2 Area for further work

The resilience and generalization related to human learning are absent from current machine learning (ML) methods, which discover statistical regularities in complicated data sets and are often utilized across various application fields. The discoveries might have extremely broad scientific and societal implications of ML approaches could help computers learn from fewer instances, transfer information between tasks, and be adaptable to changing settings and surroundings. Larger, more capable learning models have been made possible by enhanced memory resources and processing. Still, it is becoming increasingly clear that even greater co-mputing resources would not be enough to produce algorithms that could learn from a few examples and generalize beyond sets used for training. The five main areas—selection of features, representational strategies and comprehensibility, transfer-learning, continuous learning, and adaptation in different periods and environments—that are crucial for increasing ML capabilities are presented from

various angles in this study. The benefits of emerging ML techniques that potentially solve these problems can be demonstrated through proper learning tasks that call for these skills.

Over a long period, Neural Networks (NN) have been used for data mining, but their diverse applications with hyperspectral data show tremendous promise for identifying diseases. Due to many existing technologies, researchers have constantly encountered obstacles in NN applications. For example, detecting three types of diseases, namely pre-symptomatic, symptomatic, and asymptomatic diseases, from a single plant necessitates using the finest trainer sets for correct categorization. NNs have demonstrated tremendous adaptability to new difficulties in diagnosing diseases using hyperspectral data. NNs have been utilized for various applications, including data dimensionality reduction, training of image pixels or spectra as input sets, generalization of the input sets, and waveband or SDI categorization. Deep learning algorithms can identify plant leaf diseases with high accuracy if enough data is available for training. It has been discussed how to improve classification accuracy through collecting large datasets with high variability, data augmentation, transfer learning, visualization of CNN activation maps, and detection of plant diseases early on using small samples of plant leaves and hyper-spectral imaging. There are certain shortcomings at the same time. Most of the DL frameworks described in the literature perform well in terms of detection on the datasets they were designed for but perform poorly when applied to different datasets, indicating that the model lacks resilience. Better resilience DL models are therefore required to accommodate the various illness datasets. The PlantVillage dataset was typically utilized in studies to gauge how well the DL models performed. Despite many images in this collection of various plant species with their diseases, the images were all captured in a lab. Therefore, it is anticipated that a sizable dataset of plant diseases in actual environments will be established. Although some research use hyperspectral pictures of damaged leaves and various DL frameworks are employed for the early diagnosis of plant leaf diseases, issues still exist that prevent the broad application of HSI in the early detection of plant diseases. That is to say, acquiring the annotated datasets for the earliest detection of plant diseases is challenging. Even seasoned specialists cannot identify the locations of the invisible signs of disease and define entirely invisible disease pixels, which is crucial for HSI to identify diseases in plants.

To take the accuracy and efficiency of the neural networks further ahead, several models can be fused together by researchers in the area of machine learning to develop a more robust and less complex network that can easily detect and classify plant diseases. This work will also consider several techniques of augmenting images to enlarge the datasets for better detection and classification. There are currently few datasets, and most of them can further be increased in capacity through the augmentation of images. Neural networks require large amounts of testing, training, and validation data. Since augmentation will enlarge the databases, it will improve the performance of disease detection and classification models. In the future, different feature techniques can be explored with neural networks to find out which technique gives the best performance with the neural networks. This exploration can open doors to better and more accurate results during plant disease detection and classification. Algorithms used in deep learning are generally good. However, as new architectures come into place in the future, researchers need to develop specific and accurate algorithms for approximation algorithms for the new architectures.

To encourage more research in the future, there is a need for institutions of higher learning to develop indigenous databases having different varieties of local crop diseases. It is through research that Hybrid Neural Network Modelovation comes into being, so it is vital to create more crop disease databases for researchers to explore more and even develop more robust models. Varying crops should also be considered in the development of databases. Currently, crops do not have enough databases in the public domains.

In the future, the developed Hybrid Neural Network Model can be extended by developing a phone application to make it more interactive for farmers. In this case, the phone application can be installed for farmers to detect plant diseases as soon as possible to prevent loss associated with plant diseases. Since most of the population possesses mobile phones, it will be less tasking and less costly for farmers to detect diseases and treat them accordingly. Extension officers can also use the Hybrid Neural Network Model to advise farmers on what can be done to improve the quality and quantity of coffee yields. Government policies can be formulated to consider the education and training of farmers and the general public to enlighten on the use of models to automate plant disease detection. This will encourage people to explore

more in agriculture and produce good quality products. The government can also benchmark with other developed nations on the use of machine learning models for further improvement of the existing models, which will reflect on the quality and quantity of the product.

The results of this study can be used by agricultural bodies like KALRO to identify crop diseases early in many other crops besides tomatoes, grapes, and cucumbers. Additionally, the same research may be used to identify diseases in plant fruit and stem photos. As the number of computing devices grows daily and these machines have a finite memory and processing capacity, suggested CNN compression approaches can be used to adapt deep learning systems to work on these minuscule devices. Without affecting the quality assessment criteria, the presented approach may be further expanded to compress additional well-known pre-trained models such as Inception v3 and MobileNet. Representing an important step on real-time field robots that selectively apply the fungicide on leaves based on the infection and its severity can further improve this work.

With precision agriculture, farmers need to know what inputs are needed in what amounts and when. This requires collecting a lot of information from different sources and in different parts of the field. For instance, things like pests and diseases should be discovered before they cause harm to plants. Even after collection, the information needs to be analyzed to produce agronomic recommendations, deliver those recommendations to farmers and ensure that the farmers abide by them. This is still a challenge in Kenya to date. In the future, the government may consider availing funds to support the research and data collection processes if high-quality products are anything to go by. Also, the government can put policies in place to cushion the farmers from harsh economic times by providing subsidized farm inputs.

Machine learning technology has produced numerous outstanding advancements in agriculture and its associated applications, including detecting pests, weeds, and diseases. The well-known deep learning models need many parameters and levels to improve their learning ability, which uses up a lot of processing power. Even if deep learning may be utilized to reduce training time, forward inferences and the time required to load many features cannot be reduced, especially when considering embedded or mobile systems. In our opinion, when it comes to mobility,

affordability, and practicality, integrated or portable devices make more sense for agricultural field applications than high-performance PCs or servers.

REFERENCES

Abade, A., Ferreira, P. A., & de Barros Vidal, F. (2021). Plant diseases recognition on images using convolutional neural networks

Abderrahmane, O., Berdja, R., Ammad, F., Bensaci, O. A., & Benchabane, M. (2022). Potato late blight (Phytophthora infestans) disease forecasting using an auto-encoded long short-term memory recurrent neural networks in North-Western Algeria. Archives of Phytopathology and Plant Protection, 55(13), 1542-1557.

Adusei, Y. Y., Quaye-Ballard, J., Adjaottor, A. A., & Mensah, A. A. (2021). Spatial prediction and mapping of water quality of Owabi reservoir from satellite imageries and machine learning models. The Egyptian Journal of Remote Sensing and Space Science, 24(3), 825-833.

Ahil, M. N., Vanitha, V., & Rajathi, N. (2021, October). Apple and Grape Leaf Disease Classification using MLP and CNN. In 2021 International Conference on Advancements in Electrical, Electronics, Communication, Computing and Automation (ICAECA) (pp. 1-4). IEEE.

Ahmad, F., Khan, M. U. G., & Javed, K. (2021). Deep learning model for distinguishing novel coronavirus from other chest related infections in X-ray images. Computers in Biology and Medicine, 134, 104401.

Al Bashish, D., Braik, M., & Bani-Ahmad, S. (2011). Detection and classification of leaf diseases using k-means-based segmentation. Information Technology Journal, 10(2), 267-275.

Alam, M., Alam, M. S., Roman, M., Tufail, M., Khan, M. U., & Khan, M. T. (2020). Real-time machine-learning based crop/weed detection and classification for variable-rate spraying in precision agriculture. In 2020 7th International Conference on Electrical and Electronics Engineering (ICEEE), 273-280. doi: 10.1109/ICEEE49810.2020.9249836

Allahyari, M. (2017). A brief survey of text mining: Classification, clustering and extraction techniques.

Alloghani, M., Al-Jumeily, D., Mustafina, J., Hussain, A., & Aljaaf, A. J. (2020). A systematic review on supervised and unsupervised machine learning algorithms for data science. Supervised and unsupervised learning for data science, 3-21.

Alqahtani, H., Kavakli-Thorne, M., & Kumar, G. (2021). Applications of generative adversarial networks (GANs): An updated review. Archives of Computational Methods in Engineering, 28(2), 525-552.

Altan, G. (2019). DeepGraphNet: grafiklerin sınıflandırılmasında derin öğrenme modelleri. Avrupa Bilim ve Teknoloji Dergisi, 319-327.

Amara, J., Bouaziz, B., & Algergawy, A. (2017). A Deep Learning-based Approach for Banana Leaf Diseases Classification. In BTW (workshops) (Vol. 266, pp. 79-88).

Amara, J., Bouaziz, B., & Algergawy, A. (2017). A deep learning-based approach for banana leaf diseases classification. In Datenbanksysteme für Business, Technologie und Web (BTW 2017) Workshopband.

Amara, J., Bouaziz, B., & Algergawy, A. (2017). A deep learning-based approach for banana leaf diseases classification. In Datenbanksysteme für Business, Technologie und Web (BTW 2017) Workshop band.

Ananthi, S., & Varthini, S. V. (2012). Detection and classification of plant leaf diseases. International Journal of Research in Engineering & Applied Sciences, 2(2), 763-773.

Aparna, K., & Supriya, P. (2018). Precision agriculture in maize fields. In 2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS), 1407-1410. doi: 10.1109/ICICCS.2018.8438107

Arivazhagan, S., Newlin, S. R., Ananthi, S., & Vishnu, V. S. (2013). Detection of unhealthy region of plant leaves and classification of plant leaf diseases using texture features. Agricultural Engineering International: CIGR Journal, 15(1), 211-217.

Arriel-Elias, M. T., de Carvalho Barros Côrtes, M. V., de Sousa, T. P., Chaibub, A. A., & de Filippi, M. C. C. (2019). Induction of resistance in rice plants using bioproducts produced from Burkholderia pyrrocinia BRM 32113. Environmental Science and Pollution Research, 26(19), 19705-19718.

Arsenovic, M., Karanovic, M., Sladojevic, S., Anderla, A., & Stefanovic, D. (2019). Solving current limitations of deep learning based approaches for plant disease detection. Symmetry, 11(7), 939.

Arya, M. S., Anjali, K., & Unni, D. (2018). Detection of unhealthy plant leaves using image processing and genetic algorithm with Arduino. In 2018 International Conference on Power, Signals, Control and Computation (EPSCICON), IEEE, 1-5.

Ashqar, B. A., & Abu-Naser, S. S. (2018). Image-based tomato leaves diseases detection using deep learning.

Asif, U., Bennamoun, M., & Sohel, F. A. (2017). RGB-D object recognition and grasp detection using hierarchical cascaded forests. IEEE Transactions on Robotics, 33(3), 547-564.

Baker, R. E., Mahmud, A. S., Miller, I. F., Rajeev, M., Rasambainarivo, F., Rice, B. L., & Metcalf, C. J. E. (2021). Infectious disease in an era of global change. Nature Reviews Microbiology, 1-13.

Balasundram, S. K., Golhani, K., Shamshiri, R. R., & Vadamalai, G. (2020). Precision agriculture technologies for management of plant diseases. In Plant Disease Management Strategies for Sustainable Agriculture through Traditional and Modern Approaches (pp. 259-278). Springer, Cham.

Barbedo, J. G. (2018). Factors influencing the use of deep learning for plant disease recognition. Biosystems engineering, 172, 84-91.

Barbedo, J. G. A. (2013). Digital image processing techniques for detecting, quantifying and classifying plant diseases. SpringerPlus, 2(1), 660.

Barbhuiya, A. A., Karsh, R. K., & Jain, R. (2021). CNN based feature extraction and classification for sign language. Multimedia Tools and Applications, 80(2), 3051-3069.

Barth, J. (2019). Pepper: A Guide to the World's Favorite Spice. Rowman & Littlefield Publishers.

Baur, C., Denner, S., Wiestler, B., Navab, N., & Albarqouni, S. (2021). Autoencoders for unsupervised anomaly segmentation in brain MR images: A comparative study. Medical Image Analysis, 69, 101952.

Bedi, P., & Gole, P. (2021). Plant disease detection using hybrid model based on convolutional autoencoder and convolutional neural network. Artificial Intelligence in Agriculture, 5, 90-101.

Bedi, P., & Gole, P. (2021). Plant disease detection using hybrid model based on convolutional autoencoder and convolutional neural network. Artificial Intelligence Agriculture, 5, 90-101.

Behera, S. K., Jena, L., Rath, A. K., & Sethy, P. K. (2018). Disease classification and grading of orange using machine learning and fuzzy logic. In 2018 International Conference on Communication and Signal Processing (ICCSP), IEEE, 678-682.

Behmann, J., Mahlein, A. K., Rumpf, T., Römer, C., & Plümer, L. (2015). A review of advanced machine learning methods for the detection of biotic stress in precision crop protection. Precision Agriculture, 16(3), 239-260.

Bertrand, B., Davis, A. P., Maraval, I., Forestier, N., & Mieulet, D. (2022). Potential beverage quality of three wild coffee species (Coffea brevipes, C. congensis and C. stenophylla) and consideration of their agronomic use. Journal of the Science of Food and Agriculture.

Bhagwat, R., Kokare, R., & Dandawate, Y. (2020). A Framework for Identification of Soybean Leaf Diseases. In Techno-Societal 2018 (pp. 43-53). Springer, Cham.

Bjornlund, V., Bjornlund, H., & Van Rooyen, A. F. (2020). Exploring the factors causing the poor performance of most irrigation schemes in post-independence sub-Saharan Africa. International Journal of Water Resources Development, 36(sup1), S54-S101.

Bock, C. H., Cook, A. Z., Parker, P. E., & Gottwald, T. R. (2009). Automated image analysis of the severity of foliar citrus canker symptoms. Plant Disease, 93(6), 660-665.

Bock, C., Poole, G., Parker, P., & Gottwald, T. (2010). Plant disease severity estimated visually, by digital photography and image analysis, and by hyperspectral imaging. Critical Reviews in Plant Sciences, 29(2), 59-107.

Botalb, A., Moinuddin, M., Al-Saggaf, U. M., & Ali, S. S. (2018, August). Contrasting convolutional neural network (CNN) with multi-layer perceptron (MLP) for big data analysis. In 2018 International Conference on Intelligent and Advanced System (ICIAS) (pp. 1-5). IEEE.

Botvinick, M., Ritter, S., Wang, J. X., Kurth-Nelson, Z., Blundell, C., & Hassabis, D. (2019). Reinforcement learning, fast and slow. Trends in cognitive sciences, 23(5), 408-422. doi: 10.1016/j.tics.2019.03.004

Brahimi, M., Boukhalfa, K., & Moussaoui, A. (2017). Deep learning for tomato diseases: classification and symptoms visualization. Applied Artificial Intelligence, 31(4), 299-315.

Brunke, L., Greeff, M., Hall, A. W., Yuan, Z., Zhou, S., Panerati, J., & Schoellig, A. P. (2022). Safe learning in robotics: From learning-based control to safe reinforcement learning. Annual Review of Control, Robotics, and Autonomous Systems, 5, 411-444.

Bugshan, N., Khalil, I., Moustafa, N., Almashor, M., & Abuadbba, A. (2022). Radial basis function network with differential privacy. Future Generation Computer Systems, 127, 473-486.

Burkart, N., & Huber, M. F. (2021). A survey on the explainability of supervised machine learning. Journal of Artificial Intelligence Research, 70, 245-317.

Caldeira, R. F., Santiago, W. E., & Teruel, B. (2021). Identification of cotton leaf lesions using deep learning techniques. Sensors, 21(9), 3169.

Cervantes, J., Garcia-Lamont, F., Rodríguez-Mazahua, L., & Lopez, A. (2020). A comprehensive survey on support vector machine classification: Applications, challenges and trends. Neurocomputing, 408, 189-215.

Charbuty, B., & Abdulazeez, A. (2021). Classification based on decision tree algorithm for machine learning. Journal of Applied Science and Technology Trends, 2(01), 20-28.

Chalupowicz, L., Dombrovsky, A., Gaba, V., Luria, N., Reuven, M., Beerman, A., ...& Manulis-Sasson, S. (2019). Diagnosis of plant diseases using the Nanopore sequencing platform. Plant Pathology, 68(2), 229-238.

Chen, J., Chen, J., Zhang, D., Sun, Y., & Nanehkaran, Y. A. (2020). Using deep transfer learning for image-based plant disease identification. Computers and Electronics in Agriculture, 173, 105-393.

Chen, X., & Pan, L. (2018). A survey of graph cuts/graph search based medical image segmentation. IEEE Reviews in Biomedical Engineering, 11, 112-124.

Cheng, K., Tahir, R., Eric, L. K., & Li, M. (2020). An analysis of generative adversarial networks and variants for image synthesis on MNIST dataset. Multimedia Tools and Applications, 79(19), 13725-13752.

Chong, H. Y., Tan, S. C., & Yap, H. J. (2021). Hybridization of harmonic search algorithm in training radial basis function with dynamic decay adjustment for condition monitoring. Soft Computing, 25(15), 10221-10235.

Chowdhury, M. E. H., Rahman, T., Khandakar, A., Ayari, M. A., Khan, A. U., Khan, M. S., Al-Emadi, N., Reaz, M. B. I., Islam, M. T., & Ali, S. H. M. (2021). Automatic and Reliable Leaf Disease Detection Using Deep Learning Techniques. Agri Engineering, 3(2), 294-312.

Cui, D., Zhang, Q., Li, M., Hartman, G. L., & Zhao, Y. (2010). Image processing methods for quantitatively detecting soybean rust from multispectral images. Biosystems Engineering, 107(3), 186-193.

Cui, S., Su, Y. L., Duan, K., & Liu, Y. (2022). Maize leaf disease classification using CBAM and lightweight Autoencoder network. Journal of Ambient Intelligence and Humanized Computing, 1-11.

Dadashzadeh, M., Abbaspour-Gilandeh, Y., Mesri-Gundoshmian, T., Sabzi, S., Hernandez-Hernandez, J. L., Hernandez-Hernandez, M., & Arribas, J. I. (2020). Weed classification for site-specific weed management using an automated stereo computer-vision machine-learning system in rice fields. Plants, 9(5), 559.

Daniya, T., & Vigneshwari, S. (2021). Deep neural network for disease detection in rice plant using texture and deep features. The Computer Journal.

Das, A., Ghosh, S., Sarkhel, R., Choudhuri, S., Das, N., & Nasipuri, M. (2019). Combining multilevel contexts of superpixel using convolutional neural networks to perform natural scene labeling. In Recent Developments in Machine Learning and Data Analytics (pp. 297-306). Springer, Singapore.

Das, D., Singh, M., Mohanty, S. S., & Chakravarty, S. (2020). Leaf disease detection using support vector machine. In 2020 International Conference on Communication and Signal Processing (ICCSP) (pp. 1036-1040). IEEE.

Dash, C. S. K., Behera, A. K., Dehuri, S., & Cho, S. B. (2020). Building a novel classifier based on teaching learning based optimization and radial basis function neural networks for non-imputed database with irrelevant features. Applied Computing and Informatics.

Decelle, A., Furtlehner, C., & Seoane, B. (2021). Equilibrium and non-equilibrium regimes in the learning of restricted Boltzmann machines. In Advances in Neural Information Processing Systems 34, 5345-5359.

Demissie, Y. T. (2019). Integrated potato (Solanum tuberosum L.) late blight (Phytophthora infestans) disease management in Ethiopia. American Journal of BioScience, 7(6), 123-130.

Deng, L., Mao, Z., Li, X., Hu, Z., Duan, F., & Yan, Y. (2018). UAV-based multispectral remote sensing for precision agriculture: A comparison between different cameras. ISPRS Journal of Photogrammetry and Remote Sensing, 146, 124-136.

Deng, X., Zhu, Z., Yang, J., Zheng, Z., Huang, Z., Yin, X., ...& Lan, Y. (2020). Detection of citrus huanglongbing based on multi-input.

Deya, A. K., Sharmaa, M., & Meshram, M. R. (2016). Image processing-based leaf rot disease detection of betel vine (Piper betleL.). In International Conference on

Computational Modeling and Security (pp. 748-754). Procedia Computer Science, 85.

Dhingra, G., Kumar, V., & Joshi, H. D. (2018). Study of digital image processing techniques for leaf disease detection and classification. Multimedia Tools and Applications, 77(15), 19951-20000.

Dilaghi, E., Lahner, E., Annibale, B., & Esposito, G. (2022). Systematic review and meta-analysis: Artificial intelligence for the diagnosis of gastric precancerous lesions and Helicobacter pylori infection. Digestive and Liver Disease.

Driscoll, A. W., Howa, J. D., Bitter, N. Q., & Ehleringer, J. R. (2020). A predictive spatial model for roasted coffee using oxygen isotopes of α-cellulose. Rapid Communications in Mass Spectrometry, 34(7), e8626.

Duguma, L. A., Muthee, K., Carsan, S., Muriuki, J., Bulitta, B. J., Ayana, A. N., ...& Suleman, K. K. (2021). Innovation pathways in the coffee sector in Ethiopia and Kenya. In Tree Commodities and Resilient Green Economies in Africa. World Agroforestry (ICRAF), Nairobi, Kenya.

Espineli, J. D., & Lewis, K. P. (2021). Internet of Things (IoT) based plant monitoring using machine learning. In Future of Information and Communication Conference (pp. 278-289). Springer, Cham.

Evsutin, O., Kokurina, A., Meshcheryakov, R., & Shumskaya, O. (2018). The adaptive algorithm of information unmistakable embedding into digital images based on the discrete Fourier transformation. Multimedia Tools and Applications, 77(21), 28567-28599.

Farhood, H., Bakhshayeshi, I., Pooshideh, M., Rezvani, N., & Beheshti, A. (2022). Recent advances of image processing techniques in agriculture. Artificial Intelligence and Data Science in Environmental Sensing, 129-153.

Fath, A. H., Madanifar, F., & Abbasi, M. (2020). Implementation of multilayer perceptron (MLP) and radial basis function (RBF) neural networks to predict solution gas-oil ratio of crude oil systems. Petroleum, 6(1), 80-91.

Faúndez-Ugalde, A. (2020). La tutela jurisdiccional efectiva con relación al derecho a ser juzgado dentro de un plazo razonable en procedimientos tributarios: experiencia comparada entre Chile y Brasil frente a la Convención Americana de Derechos Humanos. Revista de Investigações Constitucionais, 6, 687-702.

Foare, M., Pustelnik, N., & Condat, L. (2018). A new proximal method for joint image restoration and edge detection with the Mumford-Shah model. In 2018 IEEE

International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE (pp. 1553-1557).

Fong, R., Patrick, M., & Vedaldi, A. (2019). Understanding deep networks via extremal perturbations and smooth masks. In Proceedings of the IEEE/CVF international conference on computer vision (pp. 2950-2958).

Foysal, F. A., Shakirul Islam, M., Abujar, S., & Akhter Hossain, S. (2020). A novel approach for tomato diseases classification based on deep convolutional neural networks. In Proceedings of International Joint Conference on Computational Intelligence, 583-591. Springer, Singapore.

François-Lavet, V., Henderson, P., Islam, R., Bellemare, M. G., & Pineau, J. (2018). An introduction to deep reinforcement learning. Foundations and Trends® in Machine Learning, 11(3-4), 219-354

Fuentes, A., Yoon, S., Kim, S. C., & Park, D. S. (2017). A robust deep-learning-based detector for real-time tomato plant diseases and pests recognition. Sensors, 17(9), 2022.

Garcia-Ruiz, F., Sankaran, S., Maja, J. M., Lee, W. S., Rasmussen, J., & Ehsani, R. (2013). Comparison of two aerial imaging platforms for identification of huanglongbing-infected citrus trees. Computers and Electronics in Agriculture, 91, 106-115.

Gardner, W., Maliki, R., Cutts, S. M., Muir, B. W., Ballabio, D., Winkler, D. A., & Pigram, P. J. (2020). Self-organizing map and relational perspective mapping for the accurate visualization of high-dimensional hyperspectral data. Analytical Chemistry, 92(15), 10450-10459.

Gavins, F. K., Fu, Z., Elbadawi, M., Basit, A. W., Rodrigues, M. R., & Orlu, M. (2022). Machine learning predicts the effect of food on orally administered medicines. International Journal of Pharmaceutics, 611, 121329.

Geetharamani, G., & Pandian, A. (2019). Identification of plant leaf diseases using a nine-layer deep convolutional neural network. Computers & Electrical Engineering, 76, 323-338.

Ghazal, S., Al-Samarraie, H., & Aldowah, H. (2018). “I am still learning”: Modeling LMS critical success factors for promoting students’ experience and satisfaction in a blended learning environment. IEEE Access, 6, 77179-77201.

Ghosh, S., Das, N., Das, I., & Maulik, U. (2019). Understanding deep learning techniques for image segmentation. ACM Computing Surveys (CSUR), 52(4), 1-35.

Gichuru, E., Alwora, G., Gimase, J., & Kathurima, C. (2021). Coffee leaf rust (Hemileia vastatrix) in Kenya—A review. Agronomy, 11(12), 2590.

Girin, L., Leglaive, S., Bie, X., Diard, J., Hueber, T., & Alameda-Pineda, X. (2020). Dynamical variational autoencoders: A comprehensive review. arXiv preprint arXiv:2008.12595.

Guo, Y., Zhang, J., Yin, C., Hu, X., Zou, Y., Xue, Z., & Wang, W. (2020). Plant disease identification based on deep learning algorithm in smart farming. Discrete Dynamics in Nature and Society, 2020.

Han, J., Zhang, Z., Cao, J., Luo, Y., Zhang, L., Li, Z., & Zhang, J. (2020). Prediction of winter wheat yield based on multi-source data and machine learning in China. Remote Sensing, 12(2), 236. doi: 10.3390/rs12020236

Hanson, M. A., Dostálová, A., Ceroni, C., Poidevin, M., Kondo, S., & Lemaitre, B. (2019). Synergy and remarkable specificity of antimicrobial peptides in vivo using a systematic knockout approach. eLife, 8.

Haque, K. F., & Abdelgawad, A. (2020). A deep learning approach to detect COVID-19 patients from chest X-ray images. AI, 1(3), 27.

Hassan, M. K., Fleury, E., Shamonin, D., Fonk, L. G., Marinkovic, M., Jaarsma-Coes, M. G., ...& Stoel, B. (2021). An automatic framework to create patient-specific eye models from 3D magnetic resonance images for treatment selection in patients with uveal melanoma. Advances in Radiation Oncology, 6(6), 100697.

He, Y., Zeng, H., Fan, Y., Ji, S., & Wu, J. (2019). Application of deep learning in integrated pest management: A real-time system for detection and diagnosis of oilseed rape pests. Mobile Information Systems.

Hewamalage, H., Bergmeir, C., & Bandara, K. (2021). Recurrent neural networks for time series forecasting: Current status and future directions. International Journal of Forecasting, 37(1), 388-427.

Hii, W. Y. S., Alhady, S. S. N., Wahab, A. A. A., Othman, W. A. F. W., Bakar, E. A., & Akhtar, M. N. (2021). Automated Modulated Parameter Implementation Using Neural Network for Enhancement of Paint Spray. In Symposium on Intelligent Manufacturing and Mechatronics (pp. 13-19). Springer, Singapore.

Hinton, G. (2018). Deep learning—a technology with the potential to transform health care. Jama, 320(11), 1101-1102.

Hinton, G. E., Sabour, S., & Frosst, N. (2018). Matrix capsules with EM routing. In International conference on learning representations.

Hiremath, P. S., & Bhusnurmath, R. A. (2014). RGB-based color texture image classification using anisotropic diffusion and LDBP. In International Workshop on Multi-disciplinary Trends in Artificial Intelligence (pp. 101-111). Springer, Cham.

Hiteshwari, S. (2017). Recognition of Tomato Late Blight by using DWT and Component Analysis. IEEE Conference 2017.

Hiteshwari, S. (2017). Recognition of Tomato Late Blight by using DWT and Component Analysis. IEEE Conference 2017.

Hu, C., Wu, Q., Li, H., Jian, S., Li, N., & Lou, Z. (2018). Deep learning with a long short-term memory networks approach for rainfall-runoff simulation. Water, 10(11), 1543.

Huang, Q., He, H., Singh, A., Lim, S. N., & Benson, A. R. (2020). Combining label propagation and simple models out-performs graph neural networks. arXiv preprint arXiv:2010.13993.

Jadhav, S. B. (2019). Convolutional neural networks for leaf image-based plant disease classification. IAES International Journal of Artificial Intelligence, 8(4), 328.

Jepkoech, J., Mugo, D. M., Kenduiywo, B. K., & Too, E. C. (2021). The effect of adaptive learning rate on the accuracy of neural networks. International Journal of Advanced Computer Science and Applications, 12(8).

Jiang, F., Lu, Y., Chen, Y., Cai, D., & Li, G. (2020). Image recognition of four rice leaf diseases based on deep learning and support vector machine. Computers and Electronics in Agriculture, 179, 105824.

Jiang, F., Lu, Y., Chen, Y., Cai, D., & Li, G. (2020). Image recognition of four rice leaf diseases based on deep learning and support vector machine. Computers and Electronics in Agriculture, 179, 105824.

Jiang, T., Gradus, J. L., & Rosellini, A. J. (2020). Supervised machine learning: a brief primer. Behavior Therapy, 51(5), 675-687. doi: 10.1016/j.beth.2019.07.005

Jin, X. B., Zheng, W. Z., Kong, J. L., Wang, X. Y., Bai, Y. T., Su, T. L., & Lin, S. (2021). Deep-learning forecasting method for electric power load via attention-based encoder-decoder with Bayesian optimization. Energies, 14(6), 1596.

Kadam, A. K., Wagh, V. M., Muley, A. A., Umrikar, B. N., & Sankhua, R. N. (2019). Prediction of water quality index using artificial neural network and multiple linear regression modelling approach in Shivganga River basin, India. Modeling Earth Systems and Environment, 5(3), 951-962.

Kai, S., Zhikun, L., Hang, S., & Chunhong, G. A. (2011). Research of maize disease image recognition of corn based on bp networks. In Third International Conference on Measuring Technology and Mechatronics Automation (ICMTMA), 1. IEEE, 246-249.

Kamal, K. C., Yin, Z., Wu, M., & Wu, Z. (2019). Depthwise separable convolution architectures for plant disease classification. Computers and Electronics in Agriculture, 165, 104948.

Kamilaris, A., & Prenafeta-Boldú, F. X. (2018). A review of the use of convolutional neural networks in agriculture. The Journal of Agricultural Science, 156(3), 312-322.

Kamilaris, A., Engelbrecht, A., Pitsillides, A., & Prenafeta-Boldú, F. X. (2020). Transfer of manure as fertilizer from livestock farms to crop fields: The case of Catalonia. Computers and Electronics in Agriculture, 175, 105550.

Karimi, H., Derr, T., & Tang, J. (2019). Characterizing the decision boundary of deep neural networks. arXiv preprint arXiv:1912.11460.

Kartikeyan, P., & Shrivastava, G. (2021). Review on emerging trends in detection of plant diseases using image processing with machine learning. International Journal of Computer Applications, 174, 39-48.

Kaur, P., Harnal, S., Tiwari, R., Alharithi, F. S., Almulihi, A. H., Noya, I. D., Goyal, N. A. (2018). Hybrid Convolutional Neural Network.

Kavya, N., Sriraam, N., Usha, N., Hiremath, B., Suresh, A., Sharath, D., ...& Menaka, M. (2020). Breast cancer lesion detection from cranial-caudal view of mammogram images using statistical and texture features extraction. International Journal of Biomedical and Clinical Engineering, 9(1), 16-32.

Khan, A. R., Khan, S., Harouni, M., Abbasi, R., Iqbal, S., & Mehmood, Z. (2021). Brain tumor segmentation using K-means clustering and deep learning with synthetic data augmentation for classification. Microscopy Research and Technique, 84(7), 1389-1399.

Khan, R. U., Khan, K., Albattah, W., & Qamar, A. M. (2021). Image-based detection of plant diseases: from classical machine learning to deep learning journey. Wireless Communications and Mobile Computing, 2021.

Khitthuk, C., Srikaew, A., Attakitmongcol, K., & Kumsawat, P. (2018). Plant leaf disease diagnosis from color imagery using co-occurrence matrix and artificial intelligence system. International Electrical Engineering Congress, 1-4. doi:10.1109/ieecon.2018.8371355

Kulkarni, P., Karwande, A., Kolhe, T., Kamble, S., Joshi, A., & Wyawahare, M. (2021). Plant Disease Detection Using Image Processing and Machine Learning. Department of Electronics and Telecommunication, Vishwakarma Institute of Technology, India.

Kulkarni, P., Karwande, A., Kolhe, T., Kamble, S., Joshi, A., & Wyawahare, M. (2021). Plant Disease Detection Using Image Processing and Machine Learning. Department of Electronics and Telecommunication, Vishwakarma Institute of Technology, India.

Kumar, M., Kumar, A., & Palaparthy, V. S. (2020). Soil Sensors-Based Prediction System for Plant Diseases Using Exploratory Data Analysis and Machine Learning. IEEE Sensors Journal, 21(16), 17455-17468.

Kumar, M., Pal, Y., Gangadharan, S. M. P., Chakraborty, K., Yadav, C. S., Kumar, H., & Tiwari, B. (2022). Apple Sweetness Measurement and Fruit Disease Prediction Using Image Processing Techniques Based on Human-Computer Interaction for Industry 4.0. Wireless Communications and Mobile Computing, 2022.

Kurniawati, N. N., Abdullah, S. N., Abdullah, S., & Abdullah, S. (2009). Investigation on image processing techniques for diagnosing paddy diseases, 272-277.

Kurup, R. V., Anupama, M. A., Vinayakumar, R., Sowmya, V., & Soman, K. P. (2019). Capsule network for plant disease and plant species classification. In International Conference on Computational Vision and Bio-inspired Computing. Springer, 413-421.

Kwabena, P. M., Weyori, B. A., & Mighty, A. A. (2020). Gabor capsule network for plant disease detection. International Journal of Advanced Computer Science and Applications, 11(10).

Lachgar, M., Hrimech, H., & Kartit, A. (2022). Optimization techniques in deep convolutional neuronal networks applied to olive diseases classification. Artificial Intelligence in Agriculture.

Lei, J., Liu, C., & Jiang, D. (2019). Fault diagnosis of wind turbine based on Long Short-term memory networks. Renewable Energy, 133, 422-432.

Lekakis, E., Zaikos, A., Polychronidis, A., Efthimiou, C., Pourikas, I., & Mamouka, T. (2022). Evaluation of different modelling techniques with fusion of satellite, soil and agro-meteorological data for the assessment of durum wheat yield under a large scale application. Agriculture, 12(10), 1635. doi: 10.3390/agriculture12101635

Leonard, F., & Akbar, H. (2022). Coffee Grind Size Detection By Using Convolutional Neural Network (CNN) Architecture. Journal of Applied Science, Engineering, Technology, and Education, 4(1), 133-145.

Levine, S., Kumar, A., Tucker, G., & Fu, J. (2020). Offline reinforcement learning: Tutorial, review, and perspectives on open problems. arXiv preprint arXiv:2005.01643.

Lewis, D. D. (2018). Reuters-21578 text categorization test collection. Retrieved from: http://www.daviddlewis.com/resources/testcollections/reuters21578/readme.txt

Li, G., Ma, Z., & Wang, H. (2012). Image recognition of grape downy mildew and grape powdery mildew based on support vector machine. Computer and Computing Technologies in Agriculture, 5, 151-162.

Li, S., Yuan, F., Ata-UI-Karim, S. T., Zheng, H., Cheng, T., Liu, X., & Cao, Q. (2019). Combining color indices and textures of UAV-based digital imagery for rice LAI estimation. Remote Sensing, 11(15), 1763.

Liakos, K. G., Busato, P., Moshou, D., Pearson, S., & Bochtis, D. (2018). Machine learning in agriculture: A review. Sensors, 18(8), 2674.

Liang, J., Qin, Z., Xiao, S., Ou, L., & Lin, X. (2019). Efficient and secure decision tree classification for cloud-assisted online diagnosis services. IEEE Transactions on Dependable and Secure Computing, 18(4), 1632-1644. doi: 10.1109/TDSC.2018.2870456

Lim, E. A., Tan, W. H., & Junoh, A. K. (2019). An improved radial basis function networks based on quantum evolutionary algorithm for training nonlinear datasets. IAES International Journal of Artificial Intelligence, 8(2), 120.

Lim, E. A., Tan, W. H., & Junoh, A. K. (2019). An improved radial basis function networks based on quantum evolutionary algorithm for training nonlinear datasets. IAES International Journal of Artificial Intelligence, 8(2), 120.

Liu, B., & Bruch, R. (2020). Weed detection for selective spraying: A review. Current Robotics Reports, 1(1), 19-26.

Liu, B., Zhang, Y., He, D., & Li, Y. (2018). Identification of apple leaf diseases based on deep convolutional neural networks. Symmetry, 10(1), 11.

Liu, J., & Wang, X. (2021). Plant diseases and pests' detection based on deep learning: a review. Plant Methods, 17(1), 1-18.

Lu, J., Tan, L., & Jiang, H. (2021). Review on convolutional neural network (CNN) applied to plant leaf disease classification. Agriculture, 11(8), 707.

Lu, S., Gao, X., & Duan, L. M. (2019). Efficient representation of topologically ordered states with restricted Boltzmann machines. Physical Review B, 99(15), 155136.

Lu, W., Wu, Z., Zhang, S., Dong, W., Yu, Y., & Liu, B. (2019). Real-time transient stability assessment in power system based on improved SVM. Journal of Modern Power Systems and Clean Energy, 7(1), 26-37.

Lu, Y., Jiang, Z., Zhou, T., & Fu, S. (2019). An improved watershed segmentation algorithm of medical tumor image. In IOP conference series: materials science and engineering, 677(4), 042028. IOP Publishing.

Lukic, E. T., & Tuba, M. (2017). Leaf recognition algorithm using support vector machine with Hu moments and local binary patterns. In 15th International Symposium on Applied Machine Intelligence and Informatics (SAMI) (pp. 485-490). IEEE.

Mahlein, A. K. (2016). Plant disease detection by imaging sensors–parallels and specific demands for precision agriculture and plant phenotyping. Plant Disease, 100(2), 241-251.

Manavalan, R. (2020). Automatic identification of diseases in grains crops through computational approaches: A review. Computers and Electronics in Agriculture, 178, 105802.

Manoharan, H., Teekaraman, Y., Kirpichnikova, I., Kuppusamy, R., Nikolovski, S., & Baghaee, H. R. (2020). Smart grid monitoring by wireless sensors using binary logistic regression. Energies, 13(15), 3974.

Marin, D. B., Santana, L. S., Barbosa, B. D. S., Barata, R. A. P., Osco, L. P., Ramos, A. P. M., & Guimarães, P. H. S. (2021). Detecting coffee leaf rust with UAV-based vegetation indices and decision tree machine learning models. Computers and Electronics in Agriculture, 190, 106476.

Mavridou, E., Vrochidou, E., Papakostas, G. A., Pachidis, T., & Kaburlasos, V. G. (2019). Machine vision systems in precision agriculture for crop farming. Journal of Imaging, 5(12), 89.

Mehonic, A., Sebastian, A., Rajendran, B., Simeone, O., Vasilaki, E., & Kenyon, A. J. (2020). Memristors—From in-memory computing, deep learning acceleration, and spiking neural networks to the future of neuromorphic and bio-inspired computing. Advanced Intelligent Systems, 2(11), 2000085.

Melin, P., Monica, J. C., Sanchez, D., & Castillo, O. (2020). Analysis of spatial spread relationships of coronavirus (COVID-19) pandemic in the world using self-organizing maps. Chaos, Solitons & Fractals, 138, 109917.

Melko, R. G., Carleo, G., Carrasquilla, J., & Cirac, J. I. (2019). Restricted Boltzmann machines in quantum physics. Nature Physics, 15(9), 887-892.

Mensah, P. K., & Ayidzoe, M. A. (2022). Overview of CapsNet Performance evaluation methods for image classification using a dual input capsule network as a case study. International Journal of Computing and Digital Systems, 11(1), 29-43.

Militante, S. V., Gerardo, B. D., & Dionisio, N. V. (2019). Plant leaf detection and disease recognition using deep learning. In 2019 IEEE Eurasia Conference on IOT, Communication and Engineering (ECICE), IEEE, 579-582.

Mohanty, S. P., Hughes, D. P., & Salathé, M. (2016). Using deep learning for image-based plant disease detection. Frontiers in Plant Science, 7, 1419.

Mohanty, S. P., Hughes, D. P., & Salathé, M. (2016). Using deep learning for image-based plant disease detection. Frontiers in Plant Science, 7, 1419.

Mojjada, R. K., Kumar, K. K., Yadav, A., & Prasad, B. S. V. (2020). Detection of plant leaf disease using digital image processing. Materials Today: Proceedings.

Momeny, M., Jahanbakhshi, A., Neshat, A. A., Hadipour-Rokni, R., Zhang, Y. D., & Ampatzidis, Y. (2022). Detection of citrus black spot disease and ripeness level in orange fruit using learning-to-augment incorporated deep networks. Ecological Informatics, 71, 101829.

Mrunalini, R. B., & Prashant, D. R. (2011). An application of K-means clustering and artificial intelligence in pattern recognition for crop diseases. In International Conference of Advanced Information Technology (20th IPCSIT).

Nagy, B., Galata, D. L., Farkas, A., & Nagy, Z. K. (2022). Application of Artificial Neural Networks in the Process Analytical Technology of Pharmaceutical Manufacturing—a Review. The AAPS Journal, 24(4), 1-18.

Ngugi, L. C., Abelwahab, M., & Abo-Zahhad, M. (2021). Recent advances in image processing techniques for automated leaf pest and disease recognition-A review. Information Processing in Agriculture, 8(1), 27-51.

Nguyen, C., Sagan, V., Maimaitiyiming, M., Maimaitijiang, M., Bhadra, S., & Kwasniewski, M. T. (2021). Early detection of plant viral disease using hyperspectral imaging and deep learning. Sensors, 21(3), 742.

Nikolic, E. T., & Tuba, M. (2016). Edge detection in medical ultrasound images using adjusted Canny edge detection algorithm. In 24th Telecommunications Forum (TELFOR), IEEE (pp. 691-694).

Oladejo, B. F., & Ademola, O. O. (2020). Automated classification of banana leaf diseases using an optimized capsule network model. In CS &IT Conference Proceedings (Vol. 10, No. 9). CS & IT Conference Proceedings.

Orchi, H., Sadik, M., & Khaldoun, M. (2021). On using artificial intelligence and the internet of things for crop disease detection: A contemporary survey. Agriculture, 12(1), 9.

Osco, L. P., Junior, J. M., Ramos, A. P. M., de Castro Jorge, L. A., Fatholahi, S. N., de Andrade Silva, J., ...& Li, J. (2021). A review on deep learning in UAV remote sensing. International Journal of Applied Earth Observation and Geoinformation, 102, 102456.

Osisanwo, F. Y., Akinsola, J. E. T., Awodele, O., Hinmikaiye, J. O., Olakanmi, O., & Akinjobi, J. (2017). Supervised machine learning algorithms: classification and comparison. International Journal of Computer Trends and Technology (IJCTT), 48(3), 128-138.

Ouhami, M., Hafiane, A., Es-Saady, Y., El Hajji, M., & Canals, R. (2021). Computer vision, IoT and data fusion for crop disease detection using machine learning: A survey and ongoing research. Remote Sensing, 13(13), 2486.

Oyewola, D. O., Dada, E. G., Misra, S., & Damaševičius, R. (2021). Detecting cassava mosaic disease using a deep residual convolutional neural network with distinct block processing. Peer Journal of Computer Science, 7, e352.

Pan, Z., Yu, W., Yi, X., Khan, A., Yuan, F., & Zheng, Y. (2019). Recent progress on generative adversarial networks (GANs): A survey. IEEE Access, 7, 36322-36333.

Pandey, A., & Jain, K. (2022). A robust deep attention dense convolutional neural network for plant leaf disease identification and classification from smart phone captured real world images. Ecological Informatics, 70, 101725.

Pang, A., Chang, M. W., & Chen, Y. (2022). Evaluation of random forests (RF) for regional and local-scale wheat yield prediction in Southeast Australia. Sensors, 22(3), 717. doi: 10.3390/s22030717

Patel, H. H., & Prajapati, P. (2018). Study and analysis of decision tree based classification algorithms. International Journal of Computer Sciences and Engineering, 6(10), 74-78.

Pawan, S. J., & Rajan, J. (2022). Capsule Networks for Image Classification: A Review. Neurocomputing.

Pawlak, K., & Kołodziejczak, M. (2020). The role of agriculture in ensuring food security in developing countries: Considerations in the context of the problem of sustainable food production. Sustainability, 12(13), 5488.

Picon, A., Alvarez-Gila, A., Seitz, M., Ortiz-Barredo, A., Echazarra, J., & Johannes, A. (2019). Deep convolutional neural networks for mobile capture device-based crop disease classification in the wild. Computers and Electronics in Agriculture, 161, 280-290.

Polder, G., Blok, P. M., De Villiers, H. A., Van der Wolf, J. M., & Kamp, J. (2019). Potato virus Y detection in seed potatoes using deep learning on hyperspectral images. Frontiers in Plant Science, 10, 209.

Polyanskikh, S., Arinicheva, I., Arinichev, I., & Volkova, G. (2021). Autoencoders for semantic segmentation of rice fungal diseases. Journal of Phytopathology and Plant Protection, 55(13), 1542-1557.

Pujari, D., Yakkundimath, R., & Byadgi, A. S. (2016). SVM and ANN based classification of plant diseases using feature reduction technique. IJIMAI, 3(7), 6-14.

Qin F, Liu D, Sun B, Ruan L, Ma Z, Wang H (2016) Identification of alfalfa leaf diseases using image recognition technology. PLoS One, 11(12), e0168274

Qin, Y., Wang, X., & Zou, J. (2018). The optimized deep belief networks with improved logistic sigmoid units and their application in fault diagnosis for planetary gearboxes of wind turbines. IEEE Transactions on Industrial Electronics, 66(5), 3814-3824.

Qu, X., Yang, L., Guo, K., Ma, L., Sun, M., Ke, M., & Li, M. (2021). A survey on the development of self-organizing maps for unsupervised intrusion detection. Mobile networks and applications, 26(2), 808-829.

Raimondo, E. (2022). Coffee industry market strategies in developing countries (Doctoral dissertation). Politecnico di Torino, Turin, Italy.

Raj, J. S., & Ananthi, J. V. (2019). Recurrent neural networks and nonlinear prediction in support vector machines. Journal of Soft Computing Paradigm (JSCP), 1(01), 33-40.

Rajbongshi, A., Sarker, T., Ahamad, M. M., & Rahman, M. M. (2020, October). Rose Diseases Recognition using MobileNet. In 2020 4th International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT) (pp. 1-7). IEEE.

Rangarajan, A. K., Purushothaman, R., & Ramesh, A. (2018). Tomato crop disease classification using pre-trained deep learning algorithm. Procedia computer science, 133, 1040-1047.

Ray, S. (2019). A quick review of machine learning algorithms. In 2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon) (pp. 35-39). IEEE.

Reddy, J. N., Vinod, K., & Ajai, A. R. (2019). Analysis of classification algorithms for plant leaf disease detection. In 2019 IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT) (pp. 1-6). IEEE.

Rehman, F. U., Haider, M. S., Ilyas, M., Hamid, N., & Akhter, N. (2020). Leaf area detection in diseased leaves of dalbergia sissoo (Shisham) using segmentation and channel detection technique. Pakistan Journal of Agricultural Sciences, 57(3), 823-827.

Riese, F. M., Keller, S., & Hinz, S. (2019). Supervised and semi-supervised self-organizing maps for regression and classification focusing on hyperspectral data. Remote Sensing, 12(1), 7.

Sabour, S., Frosst, N., & Hinton, G. E. (2017). Dynamic routing between capsules. Advances in Neural Information Processing Systems, 30.

Sachin, D. K. (2017). Plant Disease Detection Using Image Processing. International Conference on Computing Communication Control and Automation, 2nd International Conference for Convergence in Technology.

Sachin, D. K. (2017). Plant Disease Detection Using Image Processing. In International Conference on Computing Communication Control and Automation (2nd International Conference for Convergence in Technology).

Salazar-Reque, I. F., Huamán, S. G., Kemper, G., Telles, J., & Diaz, D. (2019). An algorithm for plant disease visual symptom detection in digital images based on superpixels. International Journal on Advanced Science, Engineering, and Information Technology, 9(1), 194-203.

Saleem, M. H., Potgieter, J., & Arif, K. M. (2019). Plant disease detection and classification by deep learning. Plants, 8(11), 468.

Saleem, M. H., Potgieter, J., & Arif, K. M. (2019). Plant disease detection and classification by deep learning. Plants, 8(11), 468.

Salmi, N., & Rustam, Z. (2019). Naïve Bayes classifier models for predicting the colon cancer. In IOP Conference Series: Materials Science and Engineering, 546(5), 052068. doi: 10.1088/1757-899X/546/5/052068

Sambasivam, G., & Opiyo, G. D. (2021). A predictive machine learning application in agriculture: Cassava disease detection and classification with imbalanced dataset using convolutional neural networks. Egyptian Informatics Journal, 22(1), 27-34.

Sanjay, M. R., Madhu, P., Jawaid, M., Senthamaraikannan, P., Senthil, S., & Pradeep, S. (2018). Characterization and properties of natural fiber polymer composites: A comprehensive review. Journal of Cleaner Production, 172, 566-581

Sanjay, M. R., Madhu, P., Jawaid, M., Senthamaraikannan, P., Senthil, S., & Pradeep, S. (2018). Characterization and properties of natural fiber polymer composites: A comprehensive review. Journal of Cleaner Production, 172, 566-581.

Sannakki, S. S., Rajpurohit, V. S., Nargund, V. B., & Kulkarni, P. (2013). Diagnosis and classification of grape leaf diseases using neural networks. In 2013 Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT) (pp. 1-5). IEEE.

Sardogan, M., Tuncer, A., & Ozen, Y. (2018). Plant leaf disease detection and classification based on CNN with LVQ algorithm. In 2018 3rd International Conference on Computer Science and Engineering (UBMK) (pp. 382-385). IEEE.

Saxena, D., & Cao, J. (2021). Generative adversarial networks (GANs) challenges, solutions, and future directions. ACM Computing Surveys (CSUR), 54(3), 1-42.

Scheurer, M. S., & Slager, R. J. (2020). Unsupervised machine learning and band topology. arXiv preprint arXiv:2001.01711.

Sekulska-Nalewajko, J., Gocławski, J., & Kuźniak, E. (2019). Computer-Assisted Image Analysis of the Distribution and Intensity of Reactive Oxygen Species Accumulation in Plant Leaves. In Plants: Production, Metabolism, Signaling, and Defense Mechanisms (pp. 489-513).

Sengar, N., Dutta, M. K., & Travieso, C. M. (2018). Computer vision based technique for identification and quantification of powdery mildew disease in cherry leaves. Computing, 100(11), 1189-1201.

Shah, J. P., Prajapati, H. B., & Dabhi, V. K. (2016). A survey on detection and classification of rice plant diseases. In 2016 IEEE International Conference on Current Trends in Advanced Computing (ICCTAC) (pp. 1-8). IEEE.

Sharath, D. M., Kumar, S. A., Rohan, M. G., & Prathap, C. (2019, April). Image based plant disease detection in pomegranate plant for bacterial blight. In 2019 International Conference on Communication and Signal Processing (ICCSP) (pp. 0645-0649). IEEE.

Sharma, P., Singh, B. K., & Singh, R. P. (2018). Prediction of potato late blight disease based upon weather parameters using artificial neural network approach. In 2018 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT) (pp. 1-13). IEEE.

Shrestha, A., & Mahmood, A. (2019). Review of deep learning algorithms and architectures. IEEE Access, 7, 53040-53065.

Shrivastava, V. K., & Pradhan, M. K. (2021). Rice plant disease classification using color features: a machine learning paradigm. Journal of Plant Pathology, 103(1), 17-26.

Simonyan, K., & Zisserman, A. (2015). Two-stream convolutional networks for action recognition. In Proceedings of the Neural Information Processing Systems (NIPS).

Singh, A. K., Ganapathysubramanian, B., Sarkar, S., & Singh, A. (2018). Deep learning for plant stress phenotyping: trends and future perspectives. Trends in Plant Science, 23(10), 883-898.

Singh, A., Thakur, N., & Sharma, A. (2016). A review of supervised machine learning algorithms. In 2016 3rd International Conference on Computing for Sustainable

Global Development (INDIACom), 1310-1315. doi: 10.1109/INDIACom.2016.7549652

Singh, R., & Chandrawat, K. S. (2017). Role of phytoalexins in plant disease resistance. International Journal of Current Microbiology and Applied Sciences, 6, 125-129.

Skendžić, S., Zovko, M., Živković, I. P., Lešić, V., & Lemić, D. (2021). The impact of climate change on agricultural insect pests. Insects, 12(5), 440.

Sladojevic, S., Arsenovic, M., Anderla, A., Culibrk, D., & Stefanovic, D. (2016). Deep Neural Networks Based Recognition of Plant Diseases by Leaf Image Classification. Computer Intelligence Neuroscience, 3289801.

Somu, N., G. R. M., & Ramamritham, K. (2020). A hybrid model for building energy consumption forecasting using long short-term memory networks. Applied Energy, 261, 114131.

Sorte, L. X. B., Ferraz, C. T., Fambrini, F., dos Reis Goulart, R., & Saito, J. H. (2019). Coffee leaf disease recognition based on deep learning and texture attributes. Procedia Computer Science, 159, 135-144. doi: 10.1016/j.procs.2019.11.048

Sottocornola, G., Baric, S., Nocker, M., Stella, F., & Zanker, M. (2022). Picture-based and conversational decision support to diagnose post-harvest apple diseases. Expert Systems with Applications, 189, 116052.

Srinivas, T., Aditya Sai, G., & Mahalaxmi, R. (2022). A Comprehensive Survey of Techniques, Applications, and Challenges in Deep Learning: A Revolution in Machine Learning. International Journal of Mechanical Engineering, 7(5), 286-296.

Srivastava, A. K., Safaei, N., Khaki, S., Lopez, G., Zeng, W., Ewert, F., ...& Rahimi, J. (2022). Winter wheat yield prediction using convolutional neural networks from environmental and phenological data. Scientific Reports, 12(1), 1-14.

Stevenson, G. A. W., Impey, L., Noble, J. A., & Collins, S. (2016). An image processing technique for the visualization and quantification of Blood flow entering the placenta using 3D power Doppler ultrasound (pd-us), Placenta, 45, 85.

Stevenson, G. A. W., Impey, L., Noble, J. A., & Collins, S. (2016). An image processing technique for the visualization and quantification of Blood flow entering the placenta using 3D power Doppler ultrasound (pd-us). Placenta, 45, 85.

Tamilselvi, P., & Kumar, K. A. (2017, March). Unsupervised machine learning for clustering the infected leaves based on the leaf-colours. In 2017 Third

International Conference on Science Technology Engineering & Management (ICONSTEM), IEEE (pp. 106-110).

Taud, H., & Mas, J. F. (2018). Multilayer perceptron (MLP). In Geomatic approaches for modeling land change scenarios (pp. 451-455). Springer, Cham.

Tello-Mijares, S., Woo, F., & Flores, F. (2019). Breast cancer identification via thermography image segmentation with a gradient vector flow and a convolutional neural network. Journal of Healthcare Engineering, 2019.

Tian, K., Zeng, J., Song, T., Li, Z., Evans, A., & Li, J. (2022). Tomato leaf diseases recognition based on deep convolutional neural networks. Journal of Agricultural Engineering.

Tian, Z., Ma, W., Yang, Q., & Duan, F. (2021). Application status and challenges of machine vision in plant factory—A review. Information Processing in Agriculture.

Triguero, I., García-Gil, D., Maillo, J., Luengo, J., García, S., & Herrera, F. (2019). Transforming big data into smart data: An insight on the use of the k-nearest neighbors algorithm to obtain quality data. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 9(2), e1289.

Tripathy, A., Ankit, A., & Santanu, K. R. (2016). Classification of sentiment reviews using n-gram machine learning Approach. In: Expert Systems with Applications, 57, 117–126.

Uddin, S., Khan, A., Hossain, M. E., & Moni, M. A. (2019). Comparing different supervised machine learning algorithms for disease prediction. BMC Medical Informatics and Decision Making, 19(1), 1-16.

Uzlu, E. (2021). Estimates of greenhouse gas emission in Turkey with grey wolf optimizer algorithm-optimized artificial neural networks. Neural Computing and Applications, 33(20), 13567-13585.

Vaishnnave, M. P., Devi, K. S., Srinivasan, P., & Jothi, G. A. P. (2019, March). Detection and classification of groundnut leaf diseases using KNN classifier. In 2019 IEEE International Conference on System, Computation, Automation and Networking (ICSCAN) (pp. 1-5). IEEE.

Velásquez, D., Sánchez, A., Sarmiento, S., Toro, M., Maiza, M., & Sierra, B. (2020). A method for detecting coffee leaf rust through wireless sensor networks, remote sensing, and deep learning: case study of the caturra variety in Colombia. Applied Sciences, 10(2), 697.

Verma, A. K., Pal, S., & Kumar, S. (2020). Prediction of skin disease using ensemble data mining techniques and feature selection method—a comparative study. Applied Biochemistry and Biotechnology, 190(2), 341-359.

Verma, S., Chug, A., & Singh, A. P. (2020). Exploring capsule networks for disease classification in plants. Journal of Statistics and Management Systems, 23(2), 307-315.

Verma, T., & Dubey, S. (2020). Impact of color spaces and feature sets in automated plant diseases classifier: a comprehensive review based on rice plant images. Archives of Computational Methods in Engineering, 27(5), 1611-1632.

Wan, L., Li, H., Li, C., Wang, A., Yang, Y., & Wang, P. (2022). Hyperspectral Sensing of Plant Diseases: Principle and Methods. Agronomy, 12(6), 1451.

Wang, D., Cao, W., Zhang, F., Li, Z., Xu, S., & Wu, X. (2022). A review of deep learning in multiscale agricultural sensing. Remote Sensing, 14(3), 559.

Wang, J., & Biljecki, F. (2022). Unsupervised machine learning in urban studies: A systematic review of applications. Cities, 129, 103925. doi: 10.1016/j.cities.2022.103925

Wang, X., Wang, Y., Hu, G., Hong, D., Guo, T., Li, J., ...& Qiu, M. (2022). Review on factors affecting coffee volatiles: From seed to cup. Journal of the Science of Food and Agriculture, 102(4), 1341-1352.

Wang, Y., Cao, X., Wang, Z. L., Han, X., Han, Y. L., Wu, D., ...& Li, H. (2022). Kaolin particle film limits grapevine downy mildew epidemic under open-field conditions and stimulates the plant defense response. Australian Journal of Grape and Wine Research.

Wani, J. A., Sharma, S., Muzamil, M., Ahmed, S., Sharma, S., & Singh, S. (2022). Machine learning and deep learning based computational techniques in automatic agricultural diseases detection: Methodologies, applications, and challenges. Archives of Computational Methods in Engineering, 29(1), 641-677.

Wu, H., Wiesner-Hanks, T., Stewart, E. L., DeChant, C., Kaczmar, N., Gore, M. A., ...& Lipson, H. (2019). Autonomous detection of plant disease symptoms directly from aerial imagery. The Plant Phenome Journal, 2(1), 1-9.

Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhudinov, R., ...& Bengio, Y. (2015). Show, attend and tell: Neural image caption generation with visual attention. In International conference on machine learning (pp. 2048-2057). PMLR.

Xu, X., Qian, Y., Wang, Y., Li, Z., & Zhou, X. (2019). Iterons homologous to helper geminiviruses are essential for efficient replication of betasatellites. Journal of Virology, 93(5), e01532-18.

Yakkundimath, R., Saunshi, G., & Palaiah, S. (2022). Automatic methods for classification of visual based viral and bacterial disease symptoms in plants. International Journal of Information Technology, 14(1), 287-299.

Yang, F. J. (2018). An implementation of naive bayes classifier. In 2018 International conference on computational science and computational intelligence (CSCI) (pp. 301-306). IEEE.

Yang, X., & Guo, T. (2017). Machine learning in plant disease research. Journal of Plant Disease Research, 31(1).

Yang, Y., Zheng, K., Wu, C., Niu, X., & Yang, Y. (2019). Building an effective intrusion detection system using the modified density peak clustering algorithm and deep belief networks. Applied Sciences, 9(2), 238.

Yebasse, M., Shimelis, B., Warku, H., Ko, J., & Cheoi, K. J. (2021). Coffee disease visualization and classification. Plants, 10(6), 1257.

Yin, P., Brauer, M., Cohen, A. J., Wang, H., Li, J., Burnett, R. T., ...& Murray, C. J. (2020). The effect of air pollution on deaths, disease burden, and life expectancy across China and its provinces, 1990–2017: An analysis for the Global Burden of Disease Study 2017. The Lancet Planetary Health, 4(9), e386-e398.

Ying, G., Miao, L., Yuan, Y., & Zelin, H. (2009). A study on the method of image pre-processing for recognition of crop diseases. In International Conference on Advanced Computer Control (pp. 202-206). IEEE.

Yu, Y., Si, X., Hu, C., & Zhang, J. (2019). A review of recurrent neural networks: LSTM cells and network architectures. Neural computation, 31(7), 1235-1270.

Zaadnoordijk, L., Besold, T. R., & Cusack, R. (2022). Lessons from infant learning for unsupervised machine learning. Nature Machine Intelligence, 4(6), 510-520. doi: 10.1038/s42256-022-0097-6

Zhang, C., Benz, P., Argaw, D. M., Lee, S., Kim, J., & Rameau, F. (2021). Proceedings of the IEEE/CVF winter conference on applications of computer vision. 2021. Resnet or densenet? Introducing dense shortcuts to ResNet, 3550–3559.

Zhang, M., Wang, J., Luo, Q., Yang, C., Yang, H., & Cheng, Y. (2021). CsMYB96 enhances citrus fruit resistance against fungal pathogen by activating salicylic

acid biosynthesis and facilitating defense metabolite accumulation. Journal of Plant Physiology, 264, 153472.

Zhang, S., & Wang, Z. (2016). Cucumber disease recognition based on global-local singular value decomposition. Neurocomputing, 205, 341–348.

Zhang, S., & Wang, Z. (2016). Cucumber disease recognition based on global-local singular value decomposition. Neurocomputing, 205, 341–348.

Zhang, S., & Wang, Z. (2016). Cucumber disease recognition based on global-local singular value decomposition. Neurocomputing 205, 341–348.

Zhang, Y., Wang, L., Chen, X., Liu, Y., Wang, S., & Wang, L. (2022). Prediction of winter wheat yield at county level in China using ensemble learning. Progress in Physical Geography: Earth and Environment

Zhang, Y., Wang, S., Chen, C., Zhang, N., Wang, A., Zhu, Y., &Cai, F. (2018). Enhanced linearity of KNNS-BNKZ ceramics by combining the controls of phase composition and microstructure. Ceramics International, 44(7), 8380-8386.

Zhao, Y. Z., Zhang, Y. Y., Han, H. A., Fan, R. P., Yang, H. U., Zhong, L., ... & Yu, B. Y. (2018). Advances in the antitumor activities and mechanisms of action of steroidal saponins. Chinese Journal of Natural Medicines, 16(10), 732-748.

Zhao, Y., Chen, J., Xu, X., Lei, J., & Zhou, W. (2021). SEV-Net: residual network embedded with attention mechanism for plant disease severity detection. Concurrency and Computation: Practice and Experience, 33(10), e6161.

Zia, S., Yuksel, B., Yuret, D., & Yemez, Y. (2017). RGB-D object recognition using deep convolutional neural networks. IEEE International Conference on Computer Vision Workshops, 887-894.