A SUPPORT VECTOR MACHINE AND ARTIFICIAL NEURAL NETWORK MODEL FOR ENHANCED IMAGE CLASSIFICATION OF MAIZE LEAF DISEASES

Vincent Mbandu Ochango

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of Master of Science in Information Technology of Murang'a University of Technology

October, 2022

DECLARATION

I hereby declare that this thesis is my original work and to the best of my knowledge has not been presented for a degree award in this or any other university.

______________________ __________________

Vincent Mbandu Ochango Date

SC401/5430/2020

APPROVAL

The undersigned certify that they have read and hereby recommend for acceptance of Murang'a University of Technology a thesis entitled "**A Support Vector Machine and Artificial Neural Network Model for Enhanced Image Classification of Maize Leaf Diseases.**"

______________________ __________________

Dr. Geoffrey Mariga Wambugu, Date

Department of Information Technonology,

Murang'a University of Technology,

______________________ __________________

Dr. John Gichuki Ndia, Date

Department of Information Technonology,

Murang'a University of Technology,

DEDICATION

Dedicated to:

My mother Sarah Agiso Ochango,

You have always stood with me the entire journey of my course Masters of Science in Information Technology. The support you provided to me, financially and spiritually helped me in accomplishing my work. Whenever I tried to give up you always encouraged me to continue despite any challenges. May the almighty God bless you.

My siblings Caroline Atemi Ochango and David Atsieli,

You have really encouraged me to complete this thesis on time and you were always concerned with my progress and all the advice you gave to me, I can only say thank you, and may the almighty God bless all of you abundantly.

My uncle Benson Amukoa Abwao,

The financial support you provided to me and every time we could meet you were always concerned with my progress. You always told me that I work hard until I finish my course since it will be of benefit to me and our entire family. May God bless you.

And

In memory of my beloved father Clyde Apeli Ochango,

I always miss your encouraging words.

ACKNOWLEDGMENT

I give thanks to almighty God for giving me good health and strength to be able to write and complete this thesis on time.

I highly appreciate the support of my thesis supervisors Dr. Geoffrey Mariga Wambugu and Dr. John Gichuki Ndia who guided me through all the stages of this research. The lengthy discussion and time we had together really helped me in terms of exploring good ideas and coming up with a well-written thesis. All the comments raised by both my supervisors were addressed and this made me improve on my work hence resulting in a well-documented thesis. The supervisors also gave me humble time to work on any corrections that they gave me and were available whenever I needed clarification from them.

The encouragement from my friend James Mutinda, I don't take it lightly furthermore you created time to guide me on how to come up with contribution and every time I consulted you whether my contribution is making sense you were there to listen and give comments.

Am also grateful to my classmates Jackson Kamiri and Kennedy Githaiga whom we started this Masters of Science in Information Technology, I know the journey was not easy but your encouragement and motivation it has enabled me to finish the course, and Am so thankful to both of you and may God bless you abundantly.

I also appreciate the aid of various online resources which enabled me to carry out this thesis and finish it on time.

MAY GOD BLESS YOU ALL

ABSTRACT

Image classification accuracy is the total number of images predicted correctly out of the total images in the test dataset in the field of computer vision. Classifying the images accurately is still a challenge due to single image classification models being biased and having high variance. The research created a combination of two models (Artificial Neural Network + Support Vector Machine), the maize leaf disease image features that were extracted were passed to the developed model which classified the diseases with high accuracy compared to the single models. Dimensionality reduction was also considered to reduce the computational complexity and this was achieved by using the Histogram of Oriented Gradient feature descriptor which extracted only relevant features and through away information that was not necessary. The relevant features were considered as the key point since an image was differentiated from each other using the key points. The developed model input was the features extracted which were in a form of a vector space known as an array of numbers and each number represented a particular feature. The developed image classification model consists of two modules; the feature extraction module and the image classification module. The feature extraction module was integrated to work together with the classification module and the features extracted by the feature extraction module were normalized to make them scale-invariant and less susceptible to light which is one of the factors that usually affects image classification accuracy. The classification module was also adjusted by combining two classifiers; Artificial Neural Network and Support Vector Machine and the main reason were for the Support Vector Machine to replace the softmax layer used for classification in the Artificial Neural Network since the Support Vector Machine have the hyperplane component which is a line that accurately separates data belonging to different classes and this made SVM to classify maize leaf disease images accurately. The Support Vector Machine also has the capability of minimizing the generalization error on unseen data which resulted in better prediction results. The common rust, leaf spot, and northern leaf blight and healthy images were used during the feature extraction process, training, and validation of the model. The feature extraction methods were compared on how they perform with image classification models to find out which feature descriptor performs best. The experimental results indicated that the Histogram of Oriented Gradients performs well with the image classifiers compared to KAZE and Oriented FAST and Rotated BRIEF and the Histogram of Oriented Gradients method reduces computational complexity during the image feature generation process. The model which was a combination of three methods, Histogram of Oriented Gradient, Artificial Neural Network, and Support Vector Machine emerged the best in terms of image classification. The experimental outcome based on performance metrics indicated that the developed model had a 0.95 accuracy score. The experimental result shows that the Histogram of the Oriented Gradient together with Artificial Neural Network and Support Vector Machine classifier is the best combination model for maize leaf disease identification since it produced the highest accuracy score compared to the other image classification models. The researcher finally recommends the model to be used today and in the future when it comes to classifying maize leaf disease images.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACRONYMS AND ABREVIATIONS

ANN	Artificial Neural Network
BRIEF	Binary Robust Independent Elementary Features
CF	Cost Function
CNN	Convolutional Neural Networks
CV	Computer Vision
DT	Decision Tree
FN	False Negative
FP	False Positive
FV	Feature Vector
GD	Gradient Direction
GM	Gradient Magnitude
HL	Hinge Loss
HOG	Histogram of Oriented Gradient
IEEE	Institute of Electrical and Electronics Engineers
KNN	K-Nearest Neighbor
LBP	Local Binary Pattern
LF	Lost Function
LG	Logistic Regression
LSVC	Linear Support Vector Classifier
ML	Machine Learning
NB	Naïve Bayesian
ORB	Oriented FAST and rotated BRIEF
RF	Random Forest
RGB	Red Green Blue
SIFT	Scale Invariant Feature Transform
SK	Scikit
SURF	Speeded-Up Robust Features
SVC	Support Vector Classifier
SVM	Support Vector Machine
TN	True Negative
TP	True Positive

DEFINITION OF OPERATIONAL TERMS

Feature Extraction: The process of extracting principal components that represent an image.

Computer Vision: The field of artificial intelligence that illustrates how computer see images the way human being does.

Image Classification Accuracy: is the total number of images predicted correctly out of the total images in the test dataset in the field of computer vision.

Model Validation: is the process of evaluating the performance of the model.

Precision: It is a metric that is used to measure how often the model is correct when it classifies the image in the right class.

Recall: It is used to measure how often the model predicts yes when it is actually yes.

F1 Score: It is obtained by calculating the harmonic mean of precision and recall. The worst value for the F1 score is 0 and the best value is 1.

Confusion Matrix: is a table that is used to define the performance of a classification algorithm. A confusion matrix visualizes and summarizes the performance of a classification algorithm.

CHAPTER ONE

INTRODUCTION

1.1 Background of the Study

Maize leaf disease is one of the challenges that farmers usually encounter in the field of agriculture, especially during maize farming. Identifying the specific type of disease affecting maize is still a challenge hence if these diseases are not correctly identified it may lead to a farmer making an erroneous decision which is expensive. Image processing is one of the techniques that is used to extract features from images and then these features are passed to the image classification models which assist in classifying the images to the class they belong to hence this leads to easier identification of the diseases.

Image classification models are validated by calculating their accuracy which is an important metric since it shows the ratio of the total number of predictions with the total number of images subjected to the test. The field of computer vision still experiences a lot of challenges while trying to classify the images accurately since none of the models has ever achieved an accuracy of one hundred percent. Due to a lot of publicly available images on the internet, there are a lot of redundant images hence a method that can identify and classify the images with the highest accuracy need to be developed. Extracting features from an entire image is usually time-consuming and involves a lot of computation hence dividing the images into smaller parts and only using relevant parts of the image to extract features resulted in a balanced tradeoff between complexity and accuracy [1].

The purpose of this research is to extract features from the images, the features are in a form of a vector space which acts as an input value to the hybrid of ANN and SVM classifies which produces accurate results compared to a single model [2]. Dimensionality reduction was also considered to reduce the computational complexity and this was achieved by using the HOG feature descriptor which extracted only relevant features and through away information that was not necessary. This technique removes misleading data from images which makes the models struggle to understand the data during training, hence the model when fed with main features, it clearly distinguishes the images well leading to high accuracy score. The relevant features are considered as the key point since an image is differentiated from each other using the key points. The developed model input is the features extracted which are in a form of a vector space known as an array of numbers and each number represents a particular feature. Image processing is one of the techniques used during feature extraction where the image is first resized into a ratio of 1:2, mostly used ratio is 64 x 128 [3]. The images are divided into a patch of 8 x8 and 16 x 16 which makes the feature to be extracted easily and that is why resizing the images to a scale of 64 x 128 is very important. The application of filters to an image after calculating the vertical and horizontal gradient leads to the creation of the histogram of the oriented gradient. The calculation of the image gradient leaves only the shape and edges of the image and unnecessary information such as colored background is removed. The HOG method provides the edge direction by calculating the magnitude and direction of edges, unlike other methods which extract only the edge features to be able to recognize an image. The x and y pixel values are very important since they are used to calculate the image gradient by calculating the change in the x and y direction of every pixel. The gradient is always calculated from an image patch of 64 image pixels which is always an image of size 8 x 8. For every patch extracted from an image, the pixel matrix is generated for each patch [4]. Gx and Gy denote the change in the x and y direction which is calculated for each pixel matrix. The Gx and Gy will now store the new matrices formed after the gradient is obtained. All the pixels in an image are used to find the gradient direction and the magnitude. This is achieved by calculating the Total Gradient Magnitude and the mathematical equation used to calculate that is illustrated as shown in the equation below;

T.G.M=$\sqrt{((Gy)2+(Gx)2)}$

The direction of every image pixel is calculated using the equation;

Ɵ=arctan (Gx / Gy)

The magnitude and direction of the pixel are used to generate the histogram which in turn produces the Histogram of Oriented Gradient features [5]. Finally, the input value for the classifiers is the Histogram of Oriented Gradient features extracted from each image.

The predicted outcome for Artificial Neural Network is achieved by adjusting the weights during forward and backward propagation until the final score matches the actual score. Different nodes produce different scores and averaging the scores from various nodes reduces variance since Artificial Neural Network has low bias and high variance. Averaging the predicted outcome reduces the variance of the models significantly if the models are much uncorrelated. The same idea usually makes Random Forest bootstrap sampling features and observations by coming up with trees that are less correlated [6].

It should also be noted that image classification models usually fail to make predictions on the data that it has never been trained on. This usually makes the model perform badly which in turn results in large misclassification errors. To make the models perform well on the data it has never been trained on it is good to divide the images into training, validation, and testing images. Then use the training images to train the model and check the performance of the model using the test images. After that, the images are reshuffled by using the one used for testing to be used for training and vice versa until all the images are partly used for training and partly used for testing to enable the models to generalize well from training data to unseen data thus preventing overfitting problem from occurring and hence enhancing the image classification accuracy.

The preeminent approach to improving predictive performance is to combine the advantages of different models so that the models may work together and the size of the models has a relative improvement in the test results. Biasness is reduced by these models since the conclusion is made by combining outcomes from multiple models rather than a single model. Typically, multiple models use bagging, boosting, and voting algorithms to combine weak models to create a hybrid classifier that when subjected to a test dataset yields better predictive results hence it is considered as a supervised learning technique. The single opinion of one model is noisier than the aggregate opinion of a collection of models and this reduces the variance and the overfitting problem. Improving generalization ability is one of the most common reasons why the combination is done [7]. Measuring the extent to which the models are error-independent can tell how effective combining is. The best reason for combining is that when the models make an error on test data is that this error should not be replicated to other models, so each of the models generalizes the data well. When the model is created the researcher should consider the advantages of each member and choose the ones that seem to classify maize leaf disease images accurately, and this should result in a model that generalizes well. The image classification model should be trained on varying methods while holding the data constant since this is considered a promising approach that results in good generalization patterns thus improving the image classification accuracy.

1.2 Statement of the Problem

The advancement of information technologies has led to massive digital data which needs processing to inform business processes. A huge percentage of this data consists of images. Image processing has therefore attracted much attention from researchers recently. Many experts in the field of machine learning have developed image classification models used to classify maize leaf diseases and went further in measuring the accuracy of each model [5] [8] [10]. Unfortunately, these models are not one hundred percent accurate which leads to instances of wrong classification and hence erroneous decisions that are expensive.

Most of the studies have focused on single models which have a challenge with generalization error [6]. Meaning that the results cannot be fully relied on due to the misclassification of images. The single models like Artificial Neural Network tend to cram the features and hence be biased with the predictions since the model's disadvantages may outweigh the advantages, which in turn affects the model classification accuracy. The ANN classifies the images through trial and error during forward and backward propagation hence resulting in an overfitting problem that affects accuracy. The SVM does not perform well on large dataset since it doesn't have enough memory to store more training data and also it requires more time to be trained.

Therefore, the main goal of this study is to enhance the image classification accuracy for maize leaf diseases by creating a model that generalizes the data well.

1.3 Research Objectives

1.3.1 Main Objective

The main research objective was to develop an ensemble model for enhanced image classification of maize leaf diseases.

1.3.2 Specific Objectives

To achieve the main objective, the research was guided by the following specific objectives:

i. To analyze the existing maize leaf disease image classification models.
ii. To design an enhanced image classification model for maize leaf diseases based on objective (i) results.
iii. To validate the enhanced image classification model.

1.4 Research Questions

i. How do you analyze existing maize leaf disease image classification models?
ii. How do you design an enhanced image classification model for maize leaf diseases based on objective (i) results?
iii. How do you validate the enhanced image classification model?

1.5 Significance of the Study

The researcher developed an enhanced image classification accuracy model for maize leaf disease identification. The model was tested and validated based on machine learning metrics such as accuracy and generating a classification report together with the confusion matrix.

The research incorporated a feature extraction method that extracted relevant features i.e. the important information from an image before the images are classified. Extracting features from an entire image is usually time-consuming and involves a lot of computation hence dividing the image into smaller parts and only using relevant parts of the image to extract features resulted in a balanced tradeoff between complexity and accuracy.

The model was aimed at overcoming the generalization error associated with a single model by coming up with a model that combines the advantages of Support Vector Machine and Artificial Neural Network models which produced good predictions results. The models were subjected to data they have never seen before to increase their generalizability and this was done using the cross-validation method. The method was used to affirm if the classifiers have generalized the data well before being validated with the test data set. This assessment was very important since it enabled one to know if the models learned during the training process. The underfitting and overfitting of the model was also determined through cross-validation. The underfitting problem usually occurs when you have less training data which makes the model not to generalize the data well since the training data is used both for training and validation. Reducing the training data makes us lose hidden patterns in the data which in turn increases the error induced by bias hence losing important trends in the data. The research increased the generalizability of the model by using K-Fold cross-validation which ensured part of the training data is used both for training and validation purposes before subjecting the model to the test data set. The whole data set was considered as a k subset and 1 fold of the dataset was used for validation and the remaining k-1 fold was used for training. The total error estimation was done for all k trials to find the total effectiveness of the developed model. The method made the images to be in the test data set once and in the training data set k-1 times. It made most of the images be used both in training and testing hence making the model generalize the data well thus reducing the overfitting problem, variance, and bias significantly. The method interchanged the test set with the training set which made the method more effective and in turn improved the learning rate of the image classification models.

The best approach to improve predictive performance i.e. image classification accuracy was to use a model that combined the strengths of the Support Vector Machine and Artificial Neural Network and the size of the models had a relative improvement on the test results. Biasness was reduced by the developed model since the predicted result was from a combination of two models rather than a single model. Typically, combining models to come up with one that generalizes the data well so that it can give better predictive results enhanced the image classification accuracy obtained by a single model.

1.6 Scope of the Study

The study used the maize leaf disease dataset to test the developed model. The maize leaf disease images used included healthy, northern leaf blight, leaf spot, and common rust images. Therefore, other maize leaf disease images not in this dataset were not considered.

The metrics measured for the model were accuracy, precision, f1-score, and recall. Meaning any other metrics were not considered. The confusion matrix and classification report were also generated to see how the model classifies images when subjected to a test dataset.

The enhancement of image classification accuracy was achieved by creating a model that combined the strengths of Support Vector Machine and Artificial Neural Network models. The advantages of the two models resulted in a better model in terms of performance compared to a single model.

1.7 Limitations of the Study

A limitation of the study refers to the characteristics that will hinder the study conclusion validity and the generalizability of the results and the researcher has no control over them.

One of the challenges specifically with the image classification model is getting enough data that will be used both for training and testing purposes, which is always not enough. Collecting maize leaf disease was a challenge since it requires a camera with high specifications which will take clear images to enable the models to generalize the training data well hence producing accurate results when subjected to the test dataset. The researcher overcame this challenge by using the Kaggle website to download the maize leaf disease dataset. The dataset had images for both training and testing purposes for each category of maize leaf disease.

1.8 Contribution of the Thesis

The contribution to the thesis was made by achieving all the objectives stated in the introduction chapter as follows;

i. A better model was developed by combining the components of Support Vector Machine and Artificial Neural Network to interoperate for an enhanced image classification accuracy in maize leaf disease identification. The model was developed based on optimal parameters for Artificial Neural Network and Support Vector Machine. In the combination of Support Vector Machine and Artificial Neural Network, the SVM replaced the softmax layer in the Artificial Neural Network, and hence the SVM had the capability of minimizing the generalization error on unseen data which resulted in better prediction results. Combining the two models reduced the variance by fitting one component of each model at a time and an increase in the capacity of models reduced the bias. The combined strengths of the models offset individual model variances and biases since the generalized the data well and it provided a composite prediction where the final accuracy was better than the accuracy of individual models.
ii. A comparative analysis was done for the developed model with other existing models and the results showed that the model performed best.
iii. The developed model was validated by measuring accuracy, precision, recall, and f1 score and the result indicated that it performed well compared to single models.

1.9 Organization of the Thesis

The thesis is divided into five chapters, the first chapter is the introduction. It explains the problem statement, research objectives, research questions, scope of the study, and contribution of the thesis. It outlines a detailed problem and therefore the need of the study to address it. The objectives are the specific goals that were set out to be attained by the study. The scope on the other hand is the extent to which the study area was investigated.

The second chapter is literature review, is a detailed literature on the area of study and what other researchers have done in the studies related to the topic under review.

The third chapter is the research methodology, it outlines the steps the research process took in order to achieve the set objectives and research questions. It also explains how the data was collected, preprocessed, analyzed and research design used. The chapter also explores the research tools, experimental procedures and the findings.

The fourth chapter is results and discussion, it looks into the outcomes of the experiment as well as strengths and weakness of the proposed solution. The chapter also discussed the model accuracy and reliability.

The fifth chapter is the conclusion, recommendations and areas of possible future research work.

CHAPTER TWO

LITERATURE REVIEW

2.1 Introduction

This chapter provides a literature review of different image classification models, how the models have been used by different researchers. A literature review on different feature extraction methods has been carried out including how other authors have used the different feature extraction methods. The performance of both feature extraction methods together with image classification models has also been reviewed in the literature. The research also reviewed the advantages and disadvantages of feature descriptor methods and maize disease classifiers. Model validation was also reviewed to find out the process that is followed to validate the developed model. Based on the literature review that has been done, the research gaps have been identified which will form a basis under which this research will be established.

The identification of maize leaf diseases by extracting features from the images by using feature descriptors and the features acting as the input value to the machine learning algorithms is widely used. The machine learning algorithms are validated based on various metrics which differ for each algorithm used. The identification of images uses various methods in the field of computer vision depending on the nature of the data set used. Image classification accuracy is one of the major problems in the field of computer vision. Due to a lot of publicly available images on the internet, there is a lot of redundant images hence we need a method that can identify and classify the images with the highest accuracy. In recent studies, most machine learning algorithms classify images by using the features extracted from them. Extracting features from an entire image is usually time-consuming and involves a lot of computation hence dividing the image into smaller parts and only using relevant parts of the image to extract features usually results in a balanced tradeoff between complexity and accuracy [10].

The machine learning algorithms work hand in hand with the feature descriptor which extracts the features and passes them to the algorithms as a feature vector or an array of integers that represent the features. The dimensionality reduction technique is one of the methods that is used to reduce the number of features extracted by only considering the key points thus reducing the computational complexity during the process of extracting information from the images. The images are distinguished clearly from each other using the key points hence classifying the images using the distinctive features becomes easy and faster.

2.2 Image Classification Models

The image classification models used in the field of agriculture to classify maize leaf diseases include Support Vector Machine, Artificial Neural Network, K-Nearest Neighbour, Logistic Regression, Random Forest, Decision Tree e.t.c. The models use the features extracted from the images by the feature descriptors as their input values. The classifiers are then trained on the features to generalize the data well which assists them in prediction when subjected to a test data set. The validation process is done to affirm if the models identify the maize leaf diseases correctly [6],[8], [20]. Panigrahi et al, 2020 used machine learning algorithms to do an experiment that can detect and classify maize leaf diseases. Out of the supervised machine learning techniques used in his experiment, Random Forest emerged the best with a classification accuracy of 79.23%. During experimentation, the dataset was obtained from the plant village website and it contained a total of 3823 images. The images were labeled common rust, gray leaf spot, northern leaf blight, and healthy having 1192 images, 513 images, 956 images, and 1162 images respectively. Traning and testing of the image classification model were done using the respective images too.

2.2.1 Support Vector Machine

Cortes and Vapnik suggested a binary classification method which is a modified version of the Support Vector Machine. The main reason for the support vector machine is to identify a line that distinguishes between data points of different class categories. The lines H_1, H_2, and H_3 are the hyperplanes that are used to segregate the data points from various classes. The X_1 and X_2 are used to determine the positions of each data point.

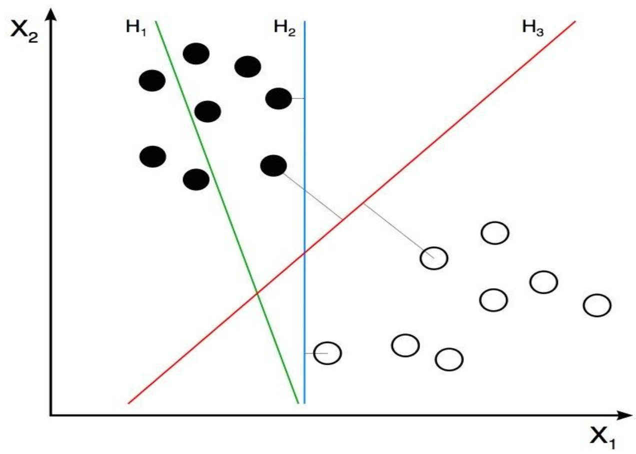

Figure 2.1 Hyper Planes (Source: [1])

The line with a large distance between the two classes of the data points is chosen and this makes the data points be classified with high accuracy. These lines are called hyperplanes and are the ones that determine the class the data point belongs to. The number of features is the determinant of the dimension of the hyperplane [4].

When there are two input features the line is drawn clearly that differentiates the two different data points. A two-dimensional plane is drawn when the input features are three and it is hard to draw the hyperplanes when the image features are more than three. The hyperplane divides the datapoint into separate classes and is usually represented by the following formula.

$$y=a.y+b$$

$$a.y+b-y=0$$

Let vector $X=(x,y)$ and $W=(a,-1)$ then in vector form hyperplane is

$$W.X+b=0$$

The support vector machine works well with high dimension space hence it provides greater accuracy. It also uses less memory since it uses a subset of training points. The major disadvantage with the support vector machine is that when using it with a large dataset it usually results in a high training time. The other disadvantage is that when used with overlapping classes it does not perform well.

2.2.2 K-Nearest Neighbors

It is a machine learning algorithm that learns from labeled data by taking the features of image X and tries to associate them with their label, the algorithm learns from the training data set and tries to classify features based on what it has learned. It classifies the features based on the majority k nearest neighbors calculated based on some distance metric [3].

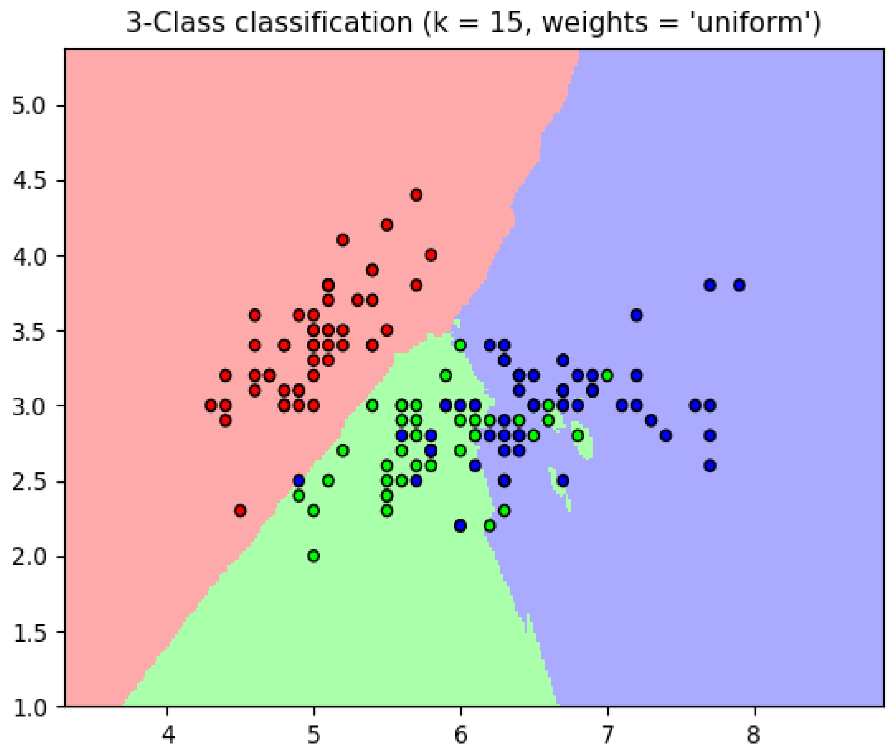

Figure 2.2 Data Points (Source: [2])

The algorithm work on the divided dataset which contains a training set and testing set. The testing set is used to see if the algorithm can classify data well best on prior knowledge during the training session by obtaining k images that are trained and closest to the validation image. The k neighbors' labels help in classifying the test images bases on voting methods and the image is classified based on many votes [2].

KNN produces high accuracy but relatively there are better classification models than KNN, the algorithm is also simple and easy to interpret and understand.

The disadvantage is that it stores all the training data which makes it computationally expensive and hence high memory storage is required compared to other image classification algorithms. Irrelevant features should not be used with KNN and also data that has been scaled since the algorithm is sensitive to irrelevant features and scaled data.

2.2.3 Random Forest Classifier

Random Forest is mostly used for classification problems since it is easy to use and produces better results because it is considered an ensemble method of many decision trees as we are aware a forest is made up of many trees and the more the trees the more the forest becomes robust. The Random Forest is made up of more decision trees and predicts results by creating decision trees on the data samples and picks the best decision tree that predicted the results with high accuracy through voting. It reduces the overfitting problem by averaging the result from every decision tree hence concluding the best result that is why it is called an ensemble method. The diagram below illustrates better how random forest works;

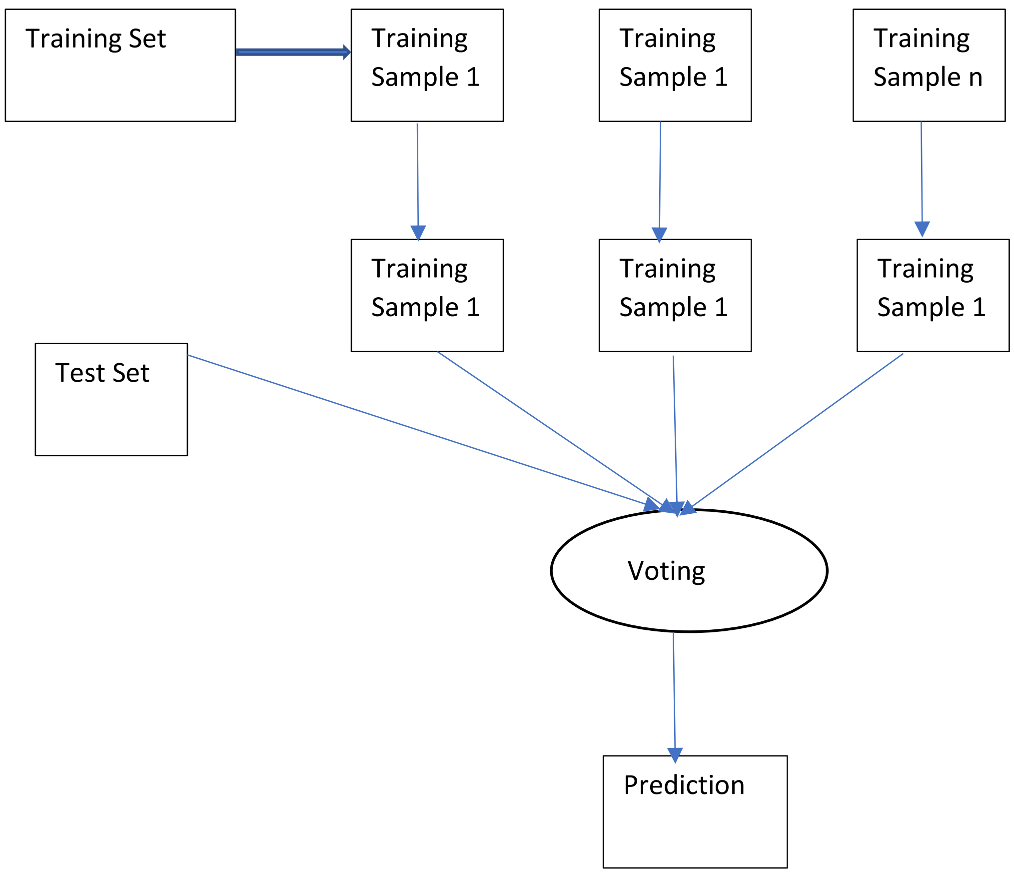

Figure 2.3 Random Forest Model

The main reason why the random forest is preferred in classification problems is that it is an ensemble method and produces more accurate prediction results by building multiple decision trees and combining them to get better results. And also it reduces the overfitting problem by averaging the results from different decision trees. The major disadvantage of the algorithms is it takes a lot of time to make predictions since it uses many decision trees to give better results hence consuming a lot of time [11].

2.2.4 Logistic Regression

It is a statistical machine learning model that is used to show the relationship between the independent and dependent variables. The model uses the logistic function which is derived from its name logistic, and the function is also known as the sigmoid function and the value of the function lies between 0 and 1. The Figure below demonstrates how the logistic function works by finding the probability that a vehicle will break since the last time it was serviced [19].

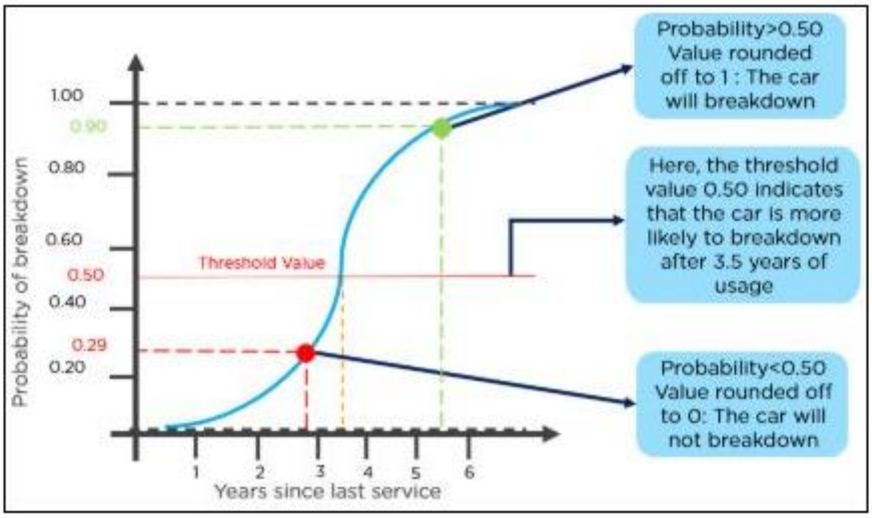

Figure 2.4 Sigmoid Function (Source: [19])

The logistic function is represented by the following equation.

$$P(x)= (e^{\beta_0 + \beta_1 x})/(1+ e^{\beta_0} + \beta_1 x)$$

$$P(x)= 1/(1 + e^{-(\beta_0 + \beta_1 x)})$$

When the data is linearly separatable the logistic regression performs better which results in high accuracy score. The good thing with the algorithm is that it does not require tuning and it is easy to develop and train a model using logistic regression since it is easy to implement and interpret it.

Logistic Regression should not be used when the number of features is more than the number of observations since it may lead to over fitting [19].

2.2.5 Artificial Neural Network

The Artificial Neural Network model is a model that mimics the way the human brain operates. The model has a three-layer and the first layer is called the input layer which works with the hidden layer by forwarding all the inputs to it for processing, the processed input is taken to the output layer which displays the final output. The hidden layer is in charge of feature extraction and all manner of calculations.

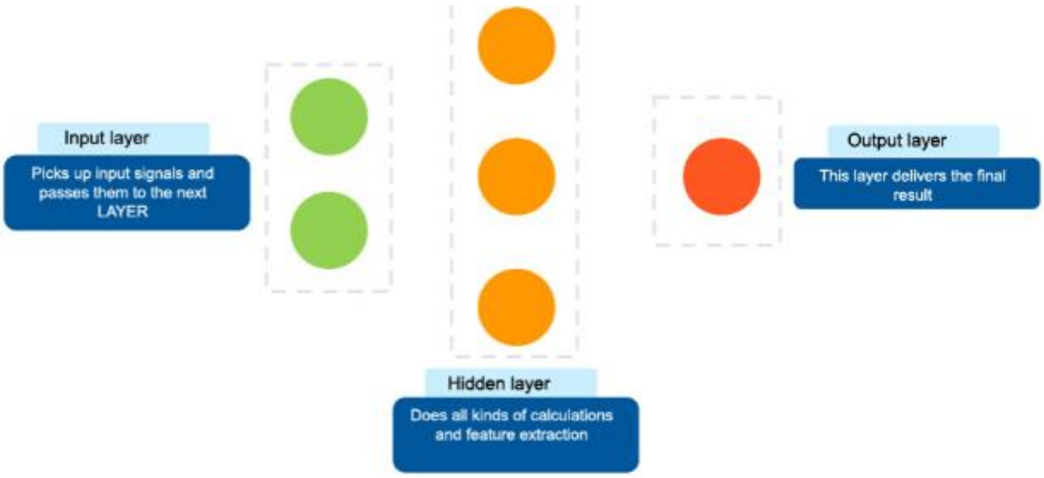

Figure 2.5 Artificial Neural Network Layers (Source: [20])

The ANN when used in image identification, the input layer takes the image of 28 by 28 pixels. The model has multiple neurons and each neuron has an activation that reads the image in form of a grayscale. The activation represents the corresponding pixel of the image with values ranging from 0 to 1, whereby 0 represents the black pixel and 1 represents the white pixel. The feature vector of the image pixels acts as the input value of the input layer. The size of the input layer is fixed and accepts only images of size 28 by 28 pixels and if your image is greater than that then you need to resize it because the size of the input layer cannot change [20]. The Figure below demonstrates how the ANN processes the image for identification purposes.

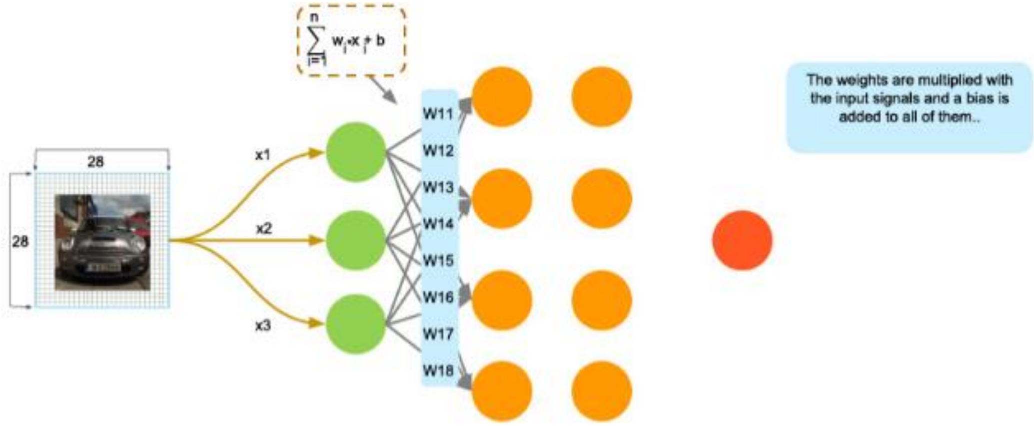

Figure 2.6 Artificial Neural Network Input Layer (Source: [20])

The advantage of ANN is that during training it generalizes the data well hence enabling it to make good predictions on unseen data. The other advantage is that it can model relationships that are complex which is good because in the real life the input and output relationships are usually complex.

The disadvantage with ANN is that you use trial and error until you get the appropriate network structure of the artificial neural network that will help you solve the problem hence losing trust in the model.

2.2.6 Decision Tree

The decision tree model is used to classify images and is a supervised learning method that is tree-based. The algorithm makes a prediction based on labeled data that's why it is called a supervised learning method. In training a model based on the labeled data the decision tree acts as the supervisor. The decision tree uses various data points to learn from simple decision rules. They also determine odds and in python, they can be used to both solve regression and classification problems. Figure 2.7 below is used to demonstrate how a Decision Tree is used to classify animals with animals' color and height acting as input features to the Decision Tree classification model.

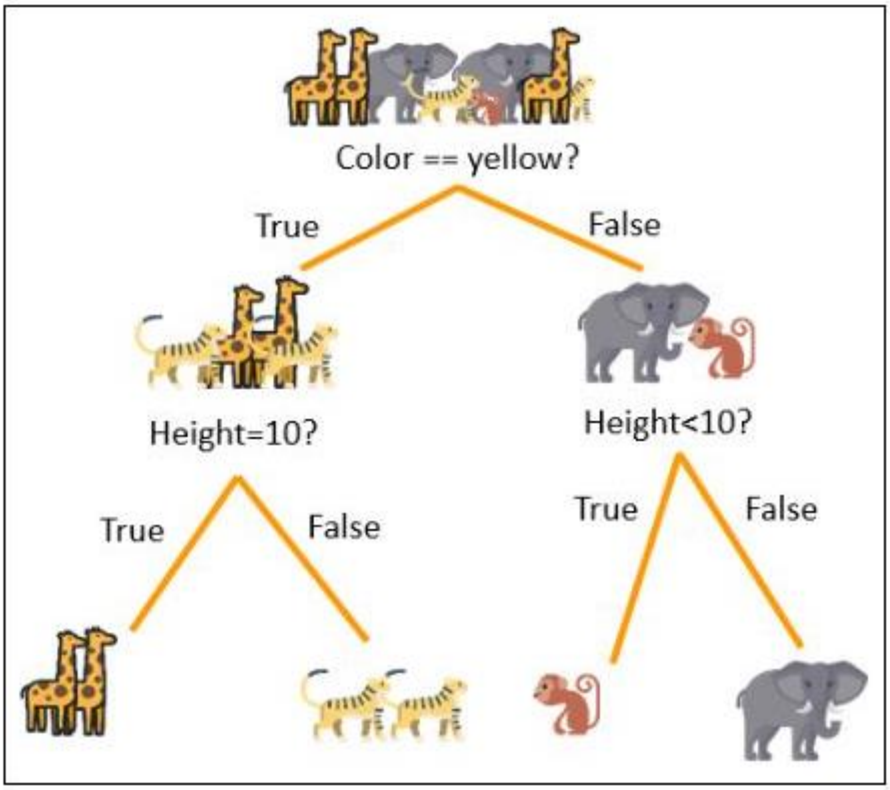

Figure 2.7 Classification of Animals Using Decision Tree (Source: [21])

The way the model splits data is usually determined by calculating the entropy of the model. The metric is applied to a dataset to measure its randomness and uncertainty.

The formula for calculating the entropy is as follows.

$$\sum_{i=1}^{k} P(valuei).log2(P(valuei)$$

One of the advantages of the Decision tree is that it can handle data that is both categorical and numerical. Even if the assumptions are violated by the actual model, the DT still performs well. Implementing, visualizing, interpreting, and understanding how the Decision Tree works is usually easy and simple [21].

2.3 Model Development Approaches

Image classification models usually require the right input for them to make correct predictions. In order to generalize the training data well, the selection of feature descriptors is important since it ensures that the right inputs are obtained for use. Data collection is very important since accurate, clear, and concise data affects the performance of machine learning algorithms. When developing these models the data collected needs to be split into training and testing to avoid biases since the testing data set should not be part of the training data. The nature of data and outcomes enables you to either create unlabelled data (unsupervised) or labeled data (supervised) models [20].

Some previous work was reviewed with an aim of identifying model development approaches as follows:

2.3.1 Hybrid of Deep Learning Using Support Vector Machine

Tang, Y 2013 developed a hybrid of deep learning using linear support vector machine for image classification and clearly showed the advantage of replacing soft max layer with linear support vector machine. Popular dataset like MNIST, CIFAR-10, and the ICML 2013 Representation Learning Workshop's face expression recognition challenge were used and the result showed that when linear support vector machine is used for classification it gives better prediction. Canadian Institute For Advanced Research 10 dataset was a 10 class object dataset with 50,000 images for training and 10,000 for testing. The colored images were 32×32 in resolution. The Convolutional Neural Network was trained with two alternating pooling and filtering layers. Horizontal reflection and jitter was applied to the data randomly before the weight was updated using a minibatch of 128 data cases. The test error was measured while using

ConvNet+Softmax and ConvNet+SVM and the results were 14.0% and 11.9% respectively [77].

2.3.2 Model Simulation Using MATLAB Tool

Yakkundimath et al, 2013 conducted an experimental work using SVM and ANN to classify three types of cereal plants. In particular, they used fungal symptoms associated with each leaf disease (jowar, maize, and wheat leaf disease) to train the classifier so as they may be able to identify the leaf diseases. The type of leaf diseases used during the research included normal, smut, powdery mildew, leaf spot, and leaf blight maize leaf disease and the researcher was only limited to those particular types of disease. The images of 750 JPG format were used and were normal and fungal affected and a procedure was used to identify and categorize the symptoms associated with each image. The images were then divided into smaller segments of the same size and this is referred to as image segmentation and preprocessing, to reduce the computational complexity. The MATLAB tool was used for the program interface, the features which acted as input to the machine learning algorithm were extracted from the maize leaf disease images using the color co-occurrence matrix algorithm. The validation for the machine learning algorithms was done and the accuracy of 83.83% and 77.75% were obtained for SVM and ANN respectively. The research results showed that SVM classified the cereal fungal diseases more accurately than ANN hence it recommended SVM be used for a similar experiment in the future [10]. A comparative analysis of feature extraction methods and other machine learning algorithms were recommended to be done in the future to have a more conclusive experimental work when it comes to the classification of cereal fungal disease images [66].

2.3.3 Genetic Algorithm and Support Vector Machine

Xiaoyang et al, 2017 created a model that classified four types of maize leaf diseases, the research was done on a china farm and the following steps were followed; The JPG types of maize leaf disease images were collected using digital cameras, the images were converted to BMP format to extract relevant features from them, and a threshold value was calculated to use it in segmenting the images. The images were classified using the GA-SVM algorithm where the images are converted to HIS from RGB and finally, the mean and standard deviation is calculated and this information is the one that assisted in the identification of the maize leaf diseases. The RBF kernel function and Support Vector Machine were among the classifiers used in the image classification. The validation was done by measuring how the machine learning algorithm performed with the test dataset and a precision of 88.72% to 92.59% was obtained for the GA-SVM algorithm and 69.63% to 90.09% for SVM was calculated which indicated that the GA-SVM generalized the training data well hence it produced more accurate results compared to Support Vector Machine [4].

2.3.4 Support Vector Machine and Artificial Neural Network

Pujari et al, 2016 developed a model of Support Vector Machine and Artificial Neural Network to identify maize leaf disease images. Feature descriptor was used to extract image features which were used to train the two classifiers. The validation of the models after training was done and the experimental results showed that SVM performs best compared to ANN. The accuracy of 0.9217 and 0.8748 was calculated for SVM and ANN respectively but unfortunately, the researcher did not mention the number of images used for both training and validation purposes [66].

2.3.5 Experimental Approach by Zhang et al, 2017

Zhang et al, 2017 did experimental work to identify five types of maize crop diseases using machine learning algorithms. The model was trained and validated with both 20 images collected and this can clearly show that there was not enough data for training and validation thus the models did not generalize the data well which led to both underfitting and overfitting problems. The maize leaf diseases used were not mentioned by the author hence the scope of the types of diseases that were being classified was not clear. Image preprocessing was done by resizing the images into 32 x 32 pixels and this was to scale and normalize them by considering their orientation and histogram equilibrium. The image pixel had a 255 grayscale level hence the images were converted to black and white color with each image having a white background. The images used were collected using digital cameras and image segmentation was done to reduce computational complexity during feature extraction. The images collected were divided into five disease categories both for training and validation. The KNN algorithm was trained and tested based on the features extracted and it was able to classify the five-leaf disease category to the class they belong to hence each image feature was associated with their respective class labels. The results for maize leaf disease classification produced a classification accuracy of above 80% after it was done 50 times. The key points clearly distinguish an image from one another and one of the future recommendations by the researcher is to use a feature descriptor that only extracts relevant information from the images hence reducing the training time of the machine learning algorithms. The training data should also be increased to enable the algorithms to generalize the data well hence avoiding the overfitting and underfitting problems [74].

2.3.6 Image Classification Using the Support Vector Machine

The model which uses the Support Vector Machine as the classifier and HOG and LBP as feature descriptors were created by Mohammad, Sayeed, and Billah in 2019 for plant disease detection. The training and testing were done using a public dataset known as the Flavia leaf dataset. The image features were extracted using Local Binary Pattern and Histogram of Oriented Gradient and they acted as the input value for the Support Vector Machine Classifier which assisted in maize leaf disease identification. The correct number of predictions made over the total number of images subjected for testing was calculated and an accuracy of 0.9125 was obtained, this was done by using the Support Vector Machine as the classifier and the hybrid of Local Binary Pattern and Histogram of Oriented Gradient as the feature descriptor. The Support Vector Machine was trained on the features extracted using the Local Binary Pattern Descriptor and after it generalized the data it was validated with the images it has never seen before and the model performed badly when subjected to a test data set by producing an accuracy of 0.406. The Histogram of Oriented Gradient algorithm was used and first the images were divided into the size of 2 x 2, 4 x 4, and 8 x 8, the features were extracted from the three segments using the HOG algorithm, and the information was fed to the Support Vector Machine classifier for identification of the images and an accuracy of 0.775, 0.8125 and 0.8531 was obtained respectively. From the results, it was concluded that when the Histogram of oriented Gradient extracts information from an image of size 8 x 8 and the Support Vector Machine is trained on the features extracted, it generalizes the data well which in turn reduces the overfitting problem hence producing more accurate results. In summary, the model which contains Support Vector Machine as the classifier for maize leaf disease identification and a hybrid of Histogram of oriented Gradient and Local Binary Pattern as the feature descriptor produces more accurate results compared to any other image classification model [39].

2.3.7 Random Forest Model by Panigrahi et al, 2020

Panigrahi et al, 2020 used machine learning algorithms to do an experiment that can detect and classify maize leaf diseases. Out of the supervised machine learning techniques used in his experiment, Random Forest emerged the best with a classification accuracy of 79.23%. During experimentation, the dataset was obtained from the plant village website and it contained a total of 3823 images. The images were labeled common rust, gray leaf spot, northern leaf blight, and healthy having 1192 images, 513 images, 956 images, and 1162 images respectively. Traning and testing of the image classification model were done using the respective images too. Image processing was done and this included converting the images into the same sizes as large images usually occupy a lot of memory and it entails a lot of computation. The colored images were also converted into grayscale images since this makes images contain only two colors which are black and white hence processing features from these images becomes easier and faster. Image segmentation was also done during the experiment and this involved dividing the images into smaller parts and then doing away with parts that are not important. The important segments of the images were used to extract features since these parts are the ones that contained relevant and unique features of the images. The features were extracted and categorized in terms of shape, color, and texture since the models can easily detect and classify maize disease images based on these features. Remember the features were extracted from the maize disease dataset which contained 3823 images and the dataset was split into two whereby 90% of the images were used as the training dataset to the machine learning algorithms and 10% of the dataset was used and the testing dataset to the image classification models. The classifiers used were SVM.KNN, RF, DT, and NB and the implementation of these classifiers were done using Windows 7 operating system installed with python 3.3. Pandas package and python machine learning library were used along with python software. The dataset had images of different sizes and hence the images were reduced to a size of 100 by 100 and this was done to make images used of the same size. Image library known as the CV in python was used to convert grayscale images to a CV2 format. The transformation of the images is performed again and again by feeding the formatted images into a pickle. Once this step is through the classification models are fed with the pickle file. The models then predict disease accordingly based on the data they are trained on. The models are then measured in terms of accuracy, precision, recall, and F1-score to determine which model performs best on the test data set. The models tend to classify maize leaf disease to be able to determine which leaves are diseased and healthy and in case the leaves have been infected with the disease the models will tell which particular disease the leaves are suffering from. This will enable the farmers to take appropriate measures to prevent maize leaves from being infected further and it will enable the farmers to know the right pesticide to apply to the infected maize leaves [37].

2.3.8 Principal Component Analysis and Support Vector Machine

Zixi et al, 2020 created a model for maize leaf identification based on principal component analysis and Support Vector Machine. The maize leaf diseases used in their experiment were healthy corn leaves, corn rust, corn big spot, and corn gray leaf spot. To outline the outline of the image and create a mask, OpenCV morphological transformation and morphological operation methods were used in the process of image background segmentation. To get a complete corn leaf image the difference is set between the background and the corn leaf by using the outline and then support vector machine and principal component analysis are applied to the processed image. The classification accuracy of four kinds of disease is 0.9050, 0.9264, 0.9123, and 0.9378 respectively when the support vector machine kernel is linear and penalty parameter C is 100. Principal Component Analysis is mainly used to reduce the image dimension which finally saves the storage space used. Its main objective was to calculate k-dimensional features from the n-dimensional features hence coming up with a new feature vector that acts as an input to the support vector classification model. The support vector machine was used to classify the processed dataset and good results were obtained [84].

2.4 Feature Generation Methods

The images are identified by looking at the features that accurately distinguish an image from one another. The features in an image can be the spots, corners, color, edges, and points of interest. Feature extraction is considered as a dimensionality reduction approach since the original image is reduced into manageable small dimensions and features are extracted which are used to represent the entire image with originality and accuracy. This approach reduces the computational complexity and balances the trade-off between accuracy and time for computing all the features in an entire image. The process enables you to use less information whenever the dataset is large without losing the actual representation of the original data. The feature generation methods at the end enable you to build a model that generalizes the data well which in turn speeds up the learning process. Reducing the number of features by using the dimensionality reduction technique makes one comes up with a model which has less machine learning effort.

2.4.1 Histogram of Oriented Gradients

HOG is a feature extraction method that is used to extract only important features from an image and hence the unnecessary information from an image is left out. The HOG first does the image processing method which involves resizing the image into a 64 x 128 image window which is an aspect ratio of 1:2[3]. The images need to be segmented into patches of an aspect ratio of 1:2 most probably 8 x8 or 16 x 16, this is to reduce the computational complexity during feature extraction and it is the main reason why image processing is done. The image filters are applied by calculating the horizontal and vertical image pixel gradients which are known as the histogram of the oriented gradient. The shape and the edges of the images are used for image representation after the colored background and unnecessary information are removed from the images. The HOG method provides the edge direction by calculating the magnitude and the direction of every edge unlike other feature descriptors which extract only the edge features from an image hence they are not concerned with the direction. Calculating and summing up the direction of x and y pixel values results in a total gradient for the entire image. The direction of every image pixel is determined from the image window extracted from the original image. The image segments are used to generate the feature matrix that acts as the input value to the image classification model [4]. The Gx and Gy denote the change in the x and y-axis for every image pixel and which is considered as the gradient magnitude for the x and y-axis pixel and this is obtained from the new matrix formed after image processing. The new matrices are then formed where one stores the Gx values and the other one stores the Gy values. The total gradient magnitude is calculated by summing up the square of the change in x and y pixels then the square root is done from the total summation. The total gradient magnitude is calculated as shown in the equation below;

$$T.G.M= \surd\ [((Gy)2 + Gx)2 +]$$

The direction of each image pixel is calculated as shown in the equation below;

$$\Theta = \arctan\ (Gx / Gy)$$

The gradient magnitude and direction are finally used to come up with the HOG features and this is the result of the histogram generated [16]. The input value for the machine learning algorithms now becomes the HOG features generated from the feature extraction process.

The HOG method is different from other feature extraction methods simply because, the method does image segmentation whose primary goal is to divide the images into smaller parts before calculating the magnitude and direction of every image pixel. The small segments are the ones that are used to come up with the histogram after generating the magnitude and direction of every image pixel. The HOG method provides the edge direction by calculating the magnitude and direction of every edge thus concentrating on the shape of the image.

One of the disadvantages of the Histogram of Oriented Gradient is that it does not perform well with rotated images, therefore it is not good to use it with images that can be detected as rotated.

2.4.2 Oriented FAST and Rotated Brief

The feature extraction method was developed at OpenCV laboratories called ORB and was free to use to detect key points and descriptors from images, which was a more viable and efficient alternative to SIFT and SURF feature extraction methods. ORB is open-source and that is why it was developed since SURF and SIFT are licensed algorithms [5].

When it comes to feature detection ORB is better than SURF and it performs as well as SIFT. Contributions made toward ORB are; the FAST feature selection method has been added as an accurate component and Oriented BRIEF features are computed efficiently.

2.4.3 Features from Accelerated Segment Test

The FAST algorithm matches the brilliance of a given pixel p in an array to neighboring sixteen others that are in a small circle near p. Three classes; lighter than p, darker than p, and similar to p are then formed after sorting the pixels in the circle. A key point is selected when more than eight pixels are brighter or darker than p [10].

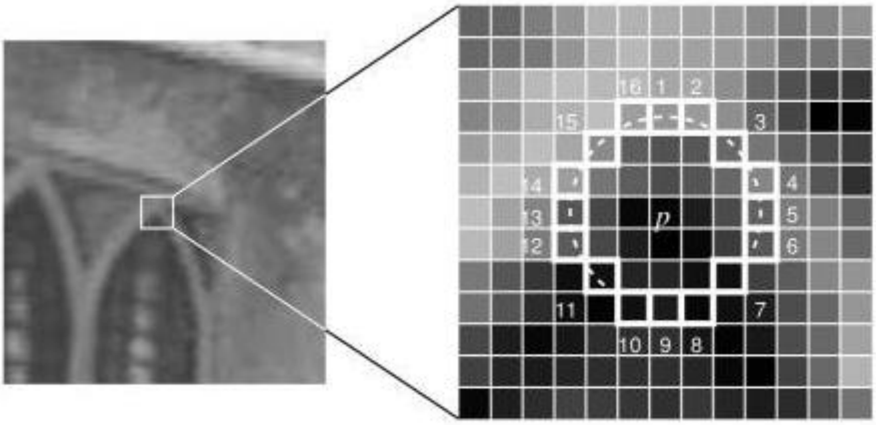

Figure 2.8 Selecting a Key Point (Source: [10])

Multi-scale features and orientation components are not part of the FAST features. A multiscale image pyramid is used by the ORB method for feature detection. Arrangements of pictures altogether of which are types of the picture at diverse resolutions is an image pyramid with multiscale representation. This algorithm for feature detection is used to detect key points in the image. The feature selection method called ORB is effective in locating key points at a different scale of each level of the pyramid and this case makes the ORB method partial scale-invariant [9].

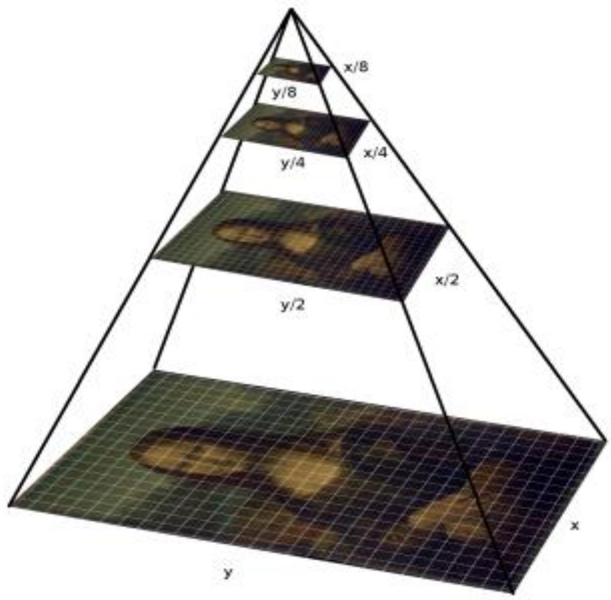

Figure 2.9 Partial Scale-Invariant (Source: [9])

Based on the degree of changing intensity around each key point, the orb assigns an orientation to each key point and the orb uses an intensity centroid for detecting intensity.

2.4.4 Binary Robust Independent Elementary Features

An object remains represented by BRIEF taking the features found by the FAST method and then converting the features to a dual characteristic vector which is a vector that contains only 1 and 0.

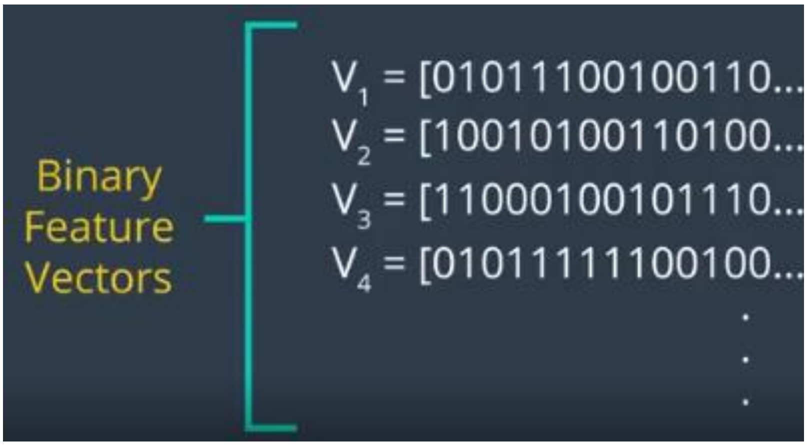

Figure 2.10 Binary Feature Vectors (Source: [7])

The Gaussian kernel is used by BRIEF to smoothen an image and this avoids key points from being affected by noise. A random pair of pixels is chosen by BRIEF in a defined neighborhood around every descriptor. Digit 1 is assigned to the first matching pixel if it is brighter than the second else digit 0 is assigned [7].

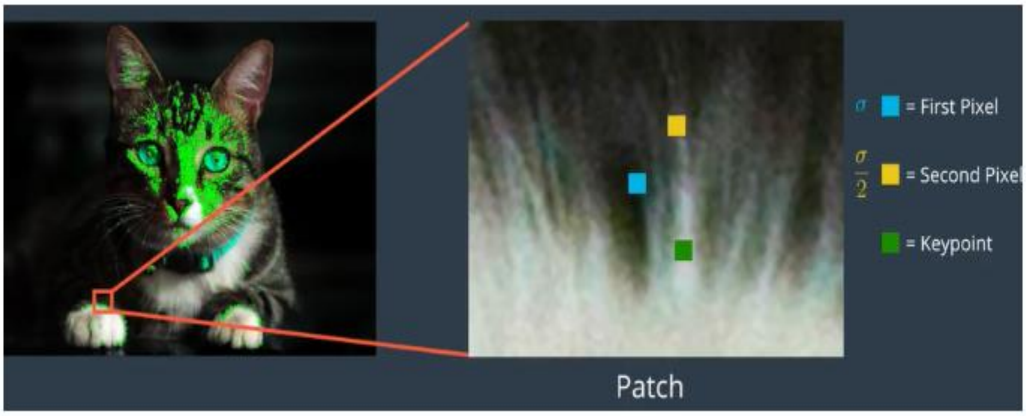

Figure 2.11 Image Patch (Source: [7])

A random pair is selected and assigned a value by the BRIEF algorithm. In the case of a keypoint BRIEF repeats, the process for 128 times for a 128-bit vector, and a vector is created like this by BRIEF for every keypoint in an image. ORB uses Rotation-aware BRIEF because BRIEF is not invariant to rotation. This functionality is added to ORB to avoid it losing out on the speed aspect of BRIEF.

The main advantage of the ORB method is that it is noise resistant and rotation invariant. ORB is a local feature detector that is fast robust. The quality of the corresponding algorithm is poor but it is a very fast method in generating image features.

2.5 Maize Leaf Disease Feature Extraction

Feature extraction is the process of obtaining the main features from images while still maintaining the intrinsic dimension of the original images and thus it is considered a dimensionality reduction technique. The features extracted include color, edges, shape, ridges, and texture. The above-named features are discussed in detail as shown below;

2.5.1 Shape of the Leaf

The shape provides co-ordinates of points such that the entire leaf area can be conveniently established through the use of convex hull algorithm. The shape of the leaf corresponds to the aspect ratio, area, and rectangular features. The Euclidean distance between the leaf tip (apex) and the base which makes the major axis defines the length of the leaf that is the main vein to the tip of the leaf. The leaf geometry defines various features with regards to shape. The diameter is the longest distance in the covered area between two points of the leaf. The length and breadth are used in finding the aspect ratio by dividing the length by the width. The end-to-end distance between the leaf margins which makes the minor axis defines the breadth (width) i.e., the distance between the leftmost to rightmost side of the leaf [20].

2.5.2 Leaf Texture

The spatial distribution of tonal variations in the neighborhood characterizes the texture. It comprises texture elements known as texels. A texel has pixel intensity and structure features. The texture is an attribute that partitions an image into regions of interest and then provides spatial arrangement information concerning color and color intensities in the image. Intensity determines the tone while the texel structure signifies the spatial connection among the texels. On the other hand, the structural approach uses the texels in certain repeated or regular patterns. Lastly, the statistical approach takes texture as a measurable aspect of the organization of intensities in a section. A fine texture results with small texels that have a large tonal difference while the contrary results in coarse texture. Texture can be defined in three ways; statistical, modeling and structural [25].

2.5.3 Leaf Color

The color of a particular leaf matters a lot since it clearly shows the type of disease affecting maize leaves. The complexity of the image is determined by the variation of pixel color at various levels. The open computer vision library reads the color in form of BGR color code hence it needs to be converted to Red, Green, and Blue color code which is mostly used by feature descriptors [21].

2.6 Model Validation

Model Validation is the process whereby the developed model is evaluated by measuring its generalization ability after it has been trained on the training dataset.

The model is evaluated by subjecting it to an unseen dataset known as the testing dataset and then evaluated in terms of the performance [9]. The first step that needs to be done is to create the model. Once the model has been constructed it needs to be tuned to have the right parameters associated with the model which will enable it to learn fast and generalize the data well during training. The training of the classifier is done by using of training dataset which is always a secondary dataset with a huge amount of data to make the algorithm learn the hidden pattern in the dataset. The validation is done once the model has learned and it can be able to make the predictions. This is achieved by subjecting it to a test dataset and this data should not be the one that was used during training. When many models are used and an exhaustive comparative study is done, the model that usually makes accurate predictions is the one that is usually selected and used even in the future when a similar problem of classification arises. The selected classifier is also validated using an independent data set to affirm that it can also be used in a different domain to provide a solution to the classification problem. Figure 2.12 illustrates the process followed during model validation.

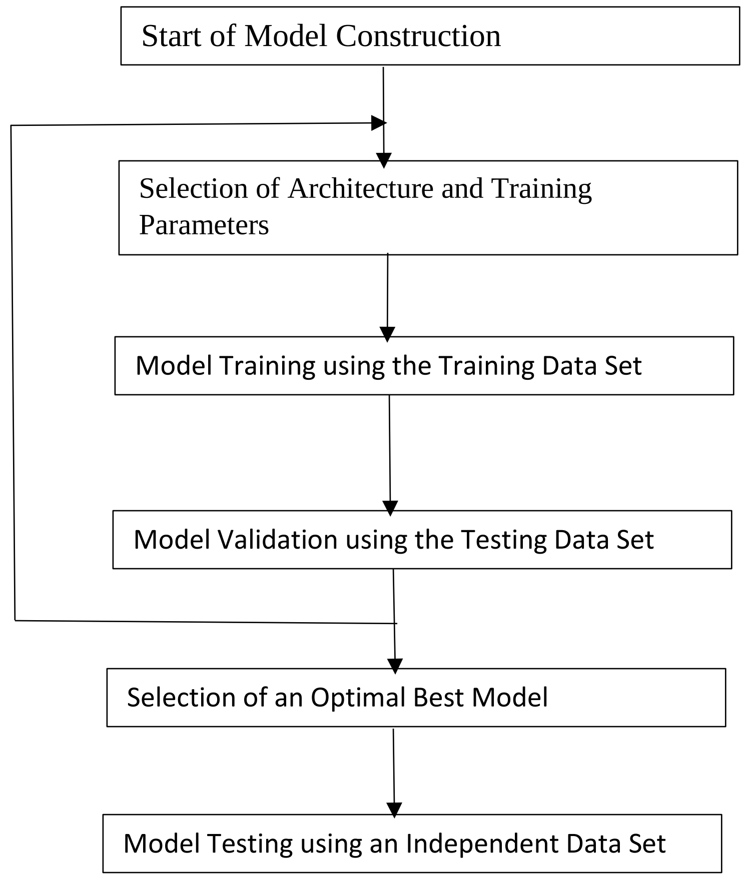

Figure 2.12 Model Validation

Model validation is a very important process during model development since it will enable the researcher to know how well the developed models make predictions when subjected to unseen data. Accuracy, precision, f1 score, and recall are some of the metrics that are usually calculated during the model validation process [13].

2.7 Research Gaps

There are several models developed to measure the classification accuracy of maize leaf diseases. The models have largely involved a single machine learning algorithm such as support vector machine algorithm, ANN, Random Forest, and K-NN, meaning there has been little effort to explore the potential of combining two or more models.

The comparative analysis needs to be done with more image classification models to ascertain the performance whether it is still the best for Support Vector Machine and Artificial Neural Network in classifying and identification of maize leaf disease images. A comparative analysis needs to be done for the enhanced model with other models to see if there is an improvement in accurately classifying the images.

A lot of research has been done with image classification models but no single researcher has provided the optimal parameters to be used with these algorithms. The research needs to be explored by tuning the parameters for these models until optimal parameters can be obtained which in turn improves the prediction results with the highest accuracy.

The data collected needs to be looked at carefully since the images have been taken based on different light conditions. Some of the images have been taken during morning, afternoon, evening, and some at night. The variation in light usually makes the model struggle in classifying the images and most researchers have not considered this aspect while developing these models. During the feature extraction process, the researcher will calculate the total magnitude and direction of every image pixel and by doing so it will make the images less susceptible to light which reduces the effort of the machine learning models in classifying the images to the class they belong to.

The images collected have variations in size, some are large while others are small, this usually affects the feature extraction process due to size variation that generates the same features that are considered different due to the image pixel position adjusted due to size. Most researchers have not yet considered this but for this research, the researcher will subject the extracted features to vector normalization process and once an image has been scaled by a factor of 2 or 0.5, the features generated won’t be affected in case of any scale variation.

Generalizability is usually important since it enables the models to learn the hidden pattern of data which helps them during the validation process. One of the methods used to make the model generalize the data well is K-Fold cross-validation. During training, cross-validation needs to be done to reduce the generalization problem and this will increase the robustness of image classification algorithms. Before subjecting your model to a test data set, you need some assurance that your model will perform best with the test data set and that is why cross-validation is needed which most researchers usually don't do. However, this method only gives an idea of the generalizability of the model with the training data and the training errors the model generates. It clearly shows the difference between the predicted images and the actual responses. The technique however does not tell how the learner will generalize when subjected to unseen or independent data set. Training data is not always enough so when you reduce the data by dividing it into training and testing usually results in an underfitting problem. This is because the learner uses fewer data to learn hence it does not generalize the hidden and important patterns in the data set. The research left part of the data for training and the other part for testing and this was achieved by using a method known as K Fold cross-validation. The method divides the data into k groups and the training is repeated k times such that one k group is used for training and the other k-1 group is used for testing. Averaging error estimation over all k trials measures the effectiveness of the model. As it can be seen, every data point gets to be k-1 times in the training set and once in the testing set. Henceforth most of the data is used in fitting the model which reduces bias and variance since the validation process uses part of the test data set only. The training data set is interchanged for training and testing purposes which eventually improves the effectiveness of the image classification models.

The Support Vector Machine has been widely used in the field of computer vision, especially in image recognition. The model has been used to classify maize leaf diseases and accuracy and precision calculated which clearly shows that the model classifies the images accurately compared to other machine learning algorithms. An exhaustive comparative analysis needs to be done to affirm if the Support Vector Machine is the best when it comes to maize leaf disease identification. The metrics such as recall, f1-score, hinge loss function, gradient descent, and ROC curve needs also to be used to measure the performance of the image classification model during the training and testing process. The current study needs to increase the number of machine learning algorithms used in the experimental work and also the training and testing dataset needs to be increased to come to a clear conclusion on the best algorithm when it comes to maize leaf disease identification. Image classification accuracy is still a challenge in the field of computer vision since there is no single model that has achieved an accuracy of 100% hence more research still needs to be done to come up with an enhanced model that produces more accurate results than the Support Vector Machine and Artificial Neural Network. The dataset used will also be an augmented dataset which will be able to tell if the model can still classify images under complex conditions. The strengths of different models need to be combined to work together and the main aim of doing this is to use a combination of base models in making predictions rather than single models since single models tend to be biased and have high variance. The other reason why the combination needs to be done is to be able to determine if it can improve the accuracy, precision, recall, and f1-score since single models are biased.

The feature extraction method, which is less susceptible to light and the one that extracts distinctive features from an image needs to be used to classify images with improved accuracy level. The feature extraction method used need also to be compared with other feature extraction methods to see if it performs best with image classification models. During training, cross-validation needs to be done to reduce the generalization problem and this will increase the robustness of image classification algorithms.

The enhanced image classification model needs to be used to arrive at the optimal prediction. The enhanced model because of its low variance and unbiasedness than the single model will result in more accurate predictions hence fewer classification errors. The model should be most preferred in making predictions than using a single base learner. Table 2.1 below shows a summary of the some of the research gaps in publications done by different authors.

Table 2.1 Summary of the Research Gaps

Publication	Techniques used	Research Gaps
Yakkundimath et al, 2013 [29]	Support Vector Machine and Artificial Neural Network	Overfitting and underfitting problem, parameter tuning.
Xiaoyang et al, 2015 [52]	Image preprocessing, GA-SVM, RBF kernel function and Support Vector Machine	Feature extraction, feature selection, feature normalization, light factor not considered and confusion matrix not used to validate the models.
Pujari et al, 2016 [30]	Support Vector Machine and Artificial Neural Network	Soft margin error, high loss function, overfitting problem, bias and variance
Zhang et al, 2017 [11]	K-Nearest Neighbor	Underfitting and overfitting problem, cross validation and dimensionality reduction
Mohammad, Sayeed, and Billah, 2019 [13]	Local Binary Pattern, Histogram of Oriented Gradient and Support Vector Machine	Vector normalization, hyper parameter tuning, outlier treatment and feature engineering.
Panigrahi et al, 2020 [14]	Support Vector Machine, K-Nearest Neighbor, Random Forest, Decision Tree, and Naïve Bayes.	Image gradient, feature selection, principal component analysis and ensembling.
Zixi et al, 2020 [15]	Principal Component Analysis and Support Vector Machine	Cross validation, hyper parameter tuning , image gradient, soft margin error, underfitting problem and ensembling

2.8 Summary

In this chapter, different image classification models and feature descriptors have been reviewed and the weakness and the strengths identified to help the researcher to identify the knowledge gaps. The researcher has reviewed how the models work together with the feature extraction methods. The study found out the best approach to extract features is to use the HOG feature extraction method since the method is used to extract only important features from an image and hence the unnecessary information from an image is left out which reduces the computational complexity of feature extraction.

The survey found out that combining models will reduce variance by fitting one component of each model at a time and an increase in the capacity of models will reduce biases while classifying images. The combined strengths of the models will offset individual model variances and biases and this will provide a composite prediction where the final accuracy will be better than the accuracy of individual models. The SVM generalizes well the training dataset compared to ANN since it scales relatively well to high dimensional data [16]. The optimal plane is also known as the hyperplane and is a line that separates data belonging to different classes which makes SVM classify data accurately. The higher the margin around the decision boundary leads to an increase in the classification accuracy. The Artificial neural network also contains a multilayer component with several neurons. During training, the information is distributed to all neurons which makes the network learn faster and store more information which assists for reference purposes.

This formed the motivation to combine the advantages of Artificial Neural Network and Support Vector Machine for an enhanced image classification accuracy in maize leaf disease identification since the ANN feature of multilayer component with several neurons will assist SVM to correlate with the ANN whenever SVM forgets the data that it was trained on and hence it enhances the image classification accuracy of the model. Combining SVM and ANN will make them work together and the advantages of SVM and ANN as highlighted above when brought together will enable the developed model to produce better image classification accuracy.

CHAPTER THREE

METHODOLOGY

3.1 Introduction

This chapter describes the methodology followed in this study to achieve the set objectives. It presents the research design, and research process describing the steps taken to achieve research objectives, research strategy, data collection, data analysis, and ethical issues.

3.2 Research Design

It entails a conceptual structure for collecting and analyzing data used during the research process. The researcher used an experimental research design since an investigation was carried out to examine the impact of combining the advantages of two image classification models and single models on image classification accuracy. The research design combined the advantages of two classification models; Support Vector Machine and Artificial Neural Network to examine whether a stronger learner created through experimental work enhances the image classification accuracy. Machine learning research is experimental since it uses quantitative methods to solve a particular problem. Quantitative research uses quantitative methods and experimental research design is one of the categories of quantitative research. This type of research uses quantifiable collected data to investigate a phenomenon systematically by using computational, mathematical, and statistical techniques which are types of quantitative methods. Machine learning requires empirical studies since no one can predict how the models will perform despite carrying out a mathematical analysis. The structure of the world helps in determining the performance of machine learning algorithms when it matches their assumptions and that is why this type of research is inherently empirical [24].

3.3 Research Process

The research process illustrated the steps that were followed to achieve each of the research objectives being investigated in this study. The process entailed three main steps which were used to achieve objective one up to objective three. The first objective was achieved by analyzing existing maize leaf disease image classification models to find out which image classification model classifies maize leaf diseases with the highest accuracy. The second objective was addressed in the second step by designing an enhanced image classification model for maize leaf diseases based on objective one results. The enhanced model was a result of the combination of the Support Vector Machine and Artificial Neural Network based on the empirical analysis carried out in objective one. The last step of the research process was to validate the developed model to affirm that it enhances image classification in maize leaf disease compared to single models. Figure 3.1 below illustrates the steps the research process followed.

Analyze Existing Maize Leaf Disease Image Classification Models (Empirical Analysis)

Design an Enhanced Image Classification Model for Maize Leaf Diseases Based on Objective One Results (Experiment)

Validate the Enhanced Image Classification Model (Experiment) `

Figure 3.1 Research Process

3.4 Description of the Training and Test Data Set

The Kaggle website was used to download the maize leaf disease data augmented dataset which was public and was divided into training and test images. The literature review done by various researchers showed that the dataset consisted of maize leaf disease images and healthy images. The 1600 images were in the whole test data set. A total of 7308 images for leaf spot, common rust, northern leaf blight, and healthy leaf were in the training data set [16]. The dataset had images of different sizes hence it was converted to images of same size i.e 256 x 256 pixel window.

3.5 Environment Setup

The environment used to create the model was done by downloading anaconda which is an IDE for running the python programs. Under anaconda, jupyter notebook application was selected which allowed the researcher to type python codes as input and display the results of the executed codes as output in a well-presentable environment. The library for locating the path of the dataset using jupyter notebook was imported by using the code below;

```
#to access system resources such as directories
import os
```

The operating system module in python known as OS allowed us to fetch the images contained in the training and test data set folders residing in the computer directory. The code for locating the path to our dataset with the help of the imported OS module was written as shown below;

```
#Set this to point to the project root; all paths will be relative to this one
project_dir = 'C:/Users/VINCENT/Dropbox/PC/Documents/maize-disease-detection'
```

```
def set_up_directories(project_dir=project_dir):
    """Sets up the paths to important directories

    Parameters
    ----------
    project_dir : string; default is the current working directory
        The path to the project root i.e '/home/lyle/projects/maize-disease-detection'

    returns
    -------
    base_dir : string
        The project directory path
    data_folder : string
        The data subfolder path
    maize_data_folder :
        The path to the subdirectory containing the maize images

    example usage
    -------------
    base_dir, data_folder, maize_data_folder = set_up_directories()
    """

    #set our base directory. This should point to the location of the plant-diseases folder
    base_dir = project_dir
    #set the path to our data folder
    data_folder = os.path.join(base_dir, 'data')
    #set the path to the maize folder and list the various categories available
    maize_data_folder = os.path.join(data_folder, 'maize')

    return base_dir, data_folder, maize_data_folder
```

3.6 Feature Extraction as an Array of Integers

Features before being extracted from the images, image processing was done and first, a library was imported known as open cv using the code import cv2 which is a library that was used in computer vision i.e enabling the computer to see the images the way human being sees them. Open CV library was then used to read the color of the images and since the library reads the color in form of BGR format, they had to be converted to a standard RGB color which is known and used by machine learning algorithms. Displaying the images from the data set and analyzing them, first a library for displaying the images was imported by using the code import matplotlib.pyplot as plt. The HOG, KAZE, and ORB feature descriptors were used to extract features from the maize leaf disease images. The images were first retrieved from their respective folders. Open CV library was then used to read the color of the images and since the library reads the color in form of BGR format, they had to be converted to a standard RGB color, and finally stored in a variable known as image as shown in the code below.

```
def get_images(disease, image_count=200, offset=0):
    """Loads a specified number of images for a given maize disease

    parameters
    ----------
    disease: string
        A string that could be common_rust, healthy, leaf_spot, nothern_leaf_blight
    image_count : int
        Number of images to return
    returns
    -------
    disease_images: list
        A list of images for the selected disease
    offset : int
        Where to begin
    """

    #this list will contain the  images returned
    disease_images = []
    #path to the images
    disease_images_path = os.path.join(maize_data_folder, disease)
    count = 0
    image_paths = os.listdir(disease_images_path)
    for image_path in image_paths[offset:]:
        if count == image_count:
            break
        image_path = os.path.join(disease_images_path, image_path)
        image = cv2.imread(image_path, cv2.IMREAD_COLOR)
        image = cv2.cvtColor(image,cv2.COLOR_BGR2RGB)
        disease_images.append(image)
        count += 1
    return disease_images
```

Once the images were retrieved a function was created that extracted the hog training and test features from the images. The HOG method is a feature descriptor that was imported from the OpenCV module. Feature extraction is a dimensionality reduction technique hence using feature descriptors such as HOG extracted the main features from the image hence converting the image from high to low dimensional space. Computation of the training and test features was done using the compute function contained in the feature descriptor. Association of features with their respective image labels was also done. The code below summarizes the steps followed during the computation of the input values to the image classification models and how they were related to their labels.

```
def extract_train_features(algorithm='hog', dataset_size=1600):

    features = []
    labels = []
    disease_names = ['common_rust', 'healthy', 'leaf_spot', 'nothern_leaf_blight']
    for disease_name in disease_names:
        images = get_images(disease_name, image_count=dataset_size)
        for image in images:
            try:
                if algorithm == 'kaze':
                    image_features = extract_features_kaze(image)
                    features.append(image_features)
                elif algorithm == 'orb':
                    image_features = extract_features_orb(image)
                    features.append(image_features)
                else:
                    image_features = extract_features_hog(image)
                    features.append(image_features)
                labels.append(disease_name)
            except AttributeError as e:
                continue
    features = np.array(features)
    labels = np.array(labels)
    features = StandardScaler().fit_transform(features)
    labels = LabelEncoder().fit_transform(labels)

    return  features, labels
```

The output of the feature extraction process was an array of integers and each feature was represented in integer form and the final results of the process acted as the input value to the machine learning algorithms and the code below demonstrates the out from the feature extraction process.

```
In [7]: hog_test_features, hog_test_labels = extract_test_features(dataset_size=400)
        hog_test_features.shape, hog_test_labels.shape

Out[7]: ((1600, 4096), (1600,))
```

It can be seen clearly from the code above that features were extracted from 400 images for each disease category in the test data set which resulted in 1600 images in total and the code when executed returned a total of 1600 images, 4096 features, and 1600 labels. The machine learning algorithms were fed with the key points extracted from each image since they are the ones that differentiated an image from one another. Despite the change in an image, the key point can still be detected since it is unique for every image and that is why the researcher used the HOG method since it only extracts the distinctive feature from an entire image. The code below shows how the features were passed to the models for training.

```
models = [
    RandomForestClassifier(n_estimators=100),
    LogisticRegression(solver='lbfgs', multi_class='auto'),
    KNeighborsClassifier(),
    MLPClassifier(),
    SVC(gamma='scale'),
    DecisionTreeClassifier(),
    LinearSVC()
]

names = [
    'Random Forest Classifier',
    'Logistic Regression',
    'K-Nearest Neighbors',
    'Artificial Neural Networks',
    'Support Vector Classifier',
    'Decision Tree Classifier',
    'Linear SVC'
]
```

First, the list of models and models' names was created and this list was passed to the train function together with the train and test features.

```
def train_base_models_timed(train_set, test_set, models=models, names=names):
```

3.7 Training and Testing of the Image Classification Models

Training is very important in the model development life cycle since it helps in generalizing the data well hence leading to better prediction results. The researcher first identified the models to be trained by storing them in a form of a list and then associating the list with the models' names. A training function was designed and passed with the classifiers, names, train, and test features as arguments. Classifier.fit() function was then used for training the models by assigning it the train features as follows.

```
classifier.fit(train_set[0], train_set[1])
```

Training time was also calculated by subtracting the start time from the stop time by using the function time.time();

```
start_time = time.time()
classifier.fit(train_set[0], train_set[1])
stop_time = time.time()
train_time.append(stop_time - start_time)
```

Testing was done using the classifier.predict() function and test features were passed to the function as an argument as indicated below.

```
predictions = classifier.predict(test_set[0])
```

A comparative analysis of the three feature extraction methods was done by passing the extracted features to the machine learning algorithms and finally measuring the image classification accuracy to compare the performance of the feature descriptors with the classifiers. The comparative analysis of the feature descriptor with the machine learning algorithms showed that the HOG method performs best with the classifier followed by KAZE and lastly the ORB method.

3.8 Feature Extraction Algorithm

Algorithm 1: Histogram of Oriented Gradient

Input: Image

Output: HOG features

Begin algorithm

1. Read Image in the form of pixels
2. Extract the HOG features:

   2.1 Divide the image pixel window by 8X8 pixel cell and calculate the gradient components with respect to each pixel (x, y) in vertical and horizontal directions

   2.2 Calculate number of blocks in vertical and horizontal directions taking block step size of 8 pixels.

   2.3 Histogram of 9 gradient directions is then calculated for each cell and extract feature vectors i.e., HOG features.

   2.4 Save all the features in a matrix.

   End algorithm

3.9 Tuning the Parameters

The set of optimal parameters for each classification algorithm was investigated to improve the performance of the classifiers. The learning rate of any machine learning algorithm is controlled by the parameters you assign to the model, optimal parameters usually improve the learning rate and they are used to determine the convergence point of the classifiers. Some of the examples for the hyperparameters for the machine learning algorithm include loss for the stochastic gradient descent and penalty for the logistic regression. The grid search is one of the optimization algorithms or tuning strategies used to get the optimal parameters for each machine learning algorithm since the method is to implement.

According to machine learning research and published work, hyperparameter tuning improves the predictive results for the image classification models. The hyperparameters were searched exhaustively over a specified set of constraints until the best parameters were arrived at through a method known as grid search. The grid search method helped the researcher to arrive at the set of optimal parameters but the process took a lot of time and it consumed a lot of computing resources since arriving at the value associated with the parameter that can make the model give accurate results is not easy. The researcher first defined the number of parameters to search over in the Jupiter notebook under anaconda IDE during the implementation of the hyperparameter tuning process since only the parameters that affect the learning rate of the machine learning algorithms were considered [33]. Figure 3.6 shows clearly some of the parameters that were tuned which include alpha, leaning rate, maximum iteration e.t.c.

```
neural_network_tuned = MLPClassifier(activation='relu', alpha=0.001, batch_size='auto', beta_1=0.9,
              beta_2=0.999, early_stopping=False, epsilon=1e-08,
              hidden_layer_sizes=(100,), learning_rate='constant',
              learning_rate_init=0.001, max_fun=15000, max_iter=100,
              momentum=0.9, n_iter_no_change=10, nesterovs_momentum=True,
              power_t=0.5, random_state=None, shuffle=True, solver='adam',
              tol=1e-05, validation_fraction=0.1, verbose=False,
              warm_start=False)
```

Figure 3.2 Hyper parameter Tuning

The regularization term in Artificial Neural Network known as the alpha parameter reduces the overfitting problem by constraining the weights. The ANN performs well on unseen data when the overfitting problem is reduced by increasing the value of the alpha parameter to 0.001.

The forward and backward propagation done by the network is shown by the number of times the data passes through the algorithm and the parameter that regulates this is the maximum iteration. The network makes accurate predictions by iteratively adjusting the weights until the right weights are assigned to the network through forward and backward propagation. From the tuned results, 100 maximum iterations produced better prediction results.

The tolerance parameter is used to show the convergence point and this is the point the network has learned and it cannot learn beyond that point hence the algorithm has increased its generalizability thus it can make accurate predictions and at that particular point, the overfitting problem is reduced. The training stopped and the network convergence reached when tol=1e-05.

3.10 K- Fold Cross-Validation

The classifiers before being introduced to the images they have never seen before the K-Fold cross-validation was done to increase the generalizability of the machine learning algorithms using the code;

```
train_scores = scores = cross_val_score(model, train_features, train_labels, scoring='accuracy', cv=10)
```

It is a good idea to evaluate the machine learning algorithms before subjecting them to the data they have never seen before and that is why the researcher used K-Fold cross-validation that tries to subdivide the training data into training and validation. This method made the algorithms to generalize the data well before making predictions with the test data set.

The underfitting problem usually comes as a result of the model not being subjected to enough training data since part of the data is used for training and the other part is used for validation. The training data needs to be more to make the models generalize the data well and this reduces variance and biasness thus the classifiers recognize the hidden patterns in the dataset. The K-Fold cross-validation used in the research increased the robustness of the model by training and validating the model from the training dataset before subjecting it to the test dataset. The training dataset was divided into 10 folds, whereby 10 folds are used for testing and the remaining 1590 are used for training. This ensured that part of the data was used for training and the remaining used for validation. All the 160 trial error estimations were averaged to obtain the total effectiveness of the model. The method ensured that data appears in the test set once. The bias was reduced significantly since the method increased the generalizability of the model by ensuring the models were trained and validated using the training dataset and this reduced the variance too. The interchanging of the data to be partly in the training set and once in the validation set increased both the robustness, learnability, and effectiveness of the machine learning algorithms.

K- Fold Cross-Validation Algorithm

i. Unsystematically interchange the set of data.

ii. Break the set of the data into groups (K groups)

iii. For every distinctive group;

    i. Let k subset to be used as a testing or validating set

    ii. Let the other k-1 subset to be put together to act as the training set

    iii. Discard the model and withhold the evaluation score

iv. Obtain the total effectiveness of the model by averaging the error estimation of all the k trials.

3.11 The Comparative Analysis of the Existing Models

Once the models were trained, a comparative analysis was done to analyze performance, which assisted the researcher to select the best two models to be combined to improve the classification accuracy. The results of the comparative study was displayed on a data frame and sorting the accuracy of the respective models' in descending order through the following code;

```
for i, classifier in enumerate(models):
    try:
        #Let us train the model and get the training time
        start_time = time.time()
        classifier.fit(train_set[0], train_set[1])
        stop_time = time.time()
        train_time.append(stop_time - start_time)
        predictions = classifier.predict(test_set[0])
        accuracy = accuracy_score(test_set[1], predictions)
        model_accuracy.append(round(accuracy, 3))
        model_names.append(names[i])
        print(f'{names[i]}: {round(accuracy, 3)}')
    except Exception as e:
        print(f'Could not train {names[i]} because of {e}')

df = pd.DataFrame({'Model':model_names, 'Accuracy':model_accuracy, 'Train Time':train_time})
df = df.sort_values(by=['Accuracy'], ascending=False)

return df
```

As you can see from the code above, the DataFrame library imported from the pandas module is passed with arguments which are the models, accuracy results and train time results. The second last statement shows that the models' accuracy is sorted in descending order before being displayed on a data frame.

3.12 The Developed Model

The developed model was a combination of the best two models; Artificial Neural Network and Support Vector Machine from the comparative analysis done for objective one. The python program that helped the researcher to combine the best two models is as follows;

```
models = [
            linear_svc_tuned,
            logistic_regression_tuned,
            svc_tuned,
            neural_network_tuned,
            random_forest_tuned
]
names = [
         'Linear SVC',
         'Logistic Regression',
         'Support Vector Classifier',
         'Neural Network',
         'Random Forest'
]
estimators = [
            ('Logistic Regression', logistic_regression_tuned),
            ('Support Vector Classifier', svc_tuned),
            ('Neural Network', neural_network_tuned),
            ('Random Forest', random_forest_tuned)
]
```

```
best_two_estimators = [
        ('Support Vector Classifier', svc_tuned),
        ('Neural Network', neural_network_tuned),
]
```

```
best_two = VotingClassifier(
    estimators=best_two_estimators,
    voting='soft'
)
```

The code above shows that first, you need to identify the best two models after conducting a comparative analysis of image classification models and store them in a defined variable known as best_two_estimators then the variable was passed to a function that combines the two classifiers known as the VotingClassifier and after the combination, the resultant model will be stored in a variable called best_two which was trained on the features extracted and validated with the test data set. The validation done indicated that the developed model had the highest classification accuracy compared to the existing ones.

The major reason why the combination was done is that the SVM had the capability of minimizing the generalization error on unseen data which resulted in better prediction results. The SVM generalizes well the training dataset compared to ANN since it scales relatively well to high dimensional data. The optimal plane is also known as the hyperplane and is a line that separates data belonging to different classes which made SVM classify maize leaf disease images accurately.

The Artificial Neural Network also contains a multilayer component with several neurons. During training, the information was distributed to all neurons which made the network learn faster and store more information which assisted for image classification purposes. This feature assisted SVM to correlate with the ANN whenever the training dataset increased since ANN had several neurons which accommodated more training data and hence enhanced the image classification accuracy results. Combining SVM and ANN made them work together and the advantages of SVM and ANN as highlighted above when brought together enabled the model to produce better image classification accuracy.

3.13 Model Validation

Model validation was done to evaluate the performance of the model with other existing image classification models. It was done mostly to measure the accuracy of the proposed model when subjected to the unseen data. A matrix representation was created to clearly show the correct predicted images and the ones that were misclassified and this was done using a confusion matrix. The confusion matrix assisted in calculating the accuracy, precision, f1 score, and recall.

		Predicted Values	
		Negative	Positive
Actual Values	Negative	True Negative TN	False Positive FP
	Positive	False Negative FN	True Positive TP

Figure 3.3 Confusion Matrix

Each prediction was based on how it matches with the actual value and it was as one of the below outcomes.

True Positive (TP) = Predicted True and True in reality.

True Negative (TN) = Predicted False and False in reality.

False Positive (FP) = Predicted True and False in reality.

False Negative (FN) = Predicted False and True in reality.

3.13.1 Classification Accuracy

Classification accuracy was a metric that measured the model performance in terms of how correct the model made the right predictions.

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

$$Accuracy = \frac{Number\ of\ correct\ predictions}{Total\ number\ of\ predictions\ made}$$

3.13.2 Precision

It was a metric that was used to measure how often the model was correct when it classified the image in the right class.

$$Precision = \frac{True\ Positives}{True\ Positives + False\ positives}$$

3.13.3 Recall

It was used to measure how often the model predicted yes when it was actually yes.

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives}$$

3.13.4 F1 Score

It was obtained by calculating the harmonic mean of precision and recall. The worst value for the F1 score is 0 and the best value is 1.

$$F1\ Score = 2 * \frac{Precision * Recall}{Precision + Recall}$$

$$F1 = \frac{TP}{TP + \frac{1}{2}(FP + FN)}$$

3.14 Ethical Considerations

To ensure that there was no conflict of interest in the research undertaken, the researcher ensured that the moral principles for doing the research were followed by first acquiring the research letter from Murang'a University of Technology (see appendix 11) and a research permit from NACOSTI (see appendix 12). The researcher had to abide by research ethics by obtaining a research license permit from NACOSTI (see appendix 12) since according to NACOSTI all the research conducted in Kenya, research ethical approval is desirable.

CHAPTER FOUR

RESULTS AND DISCUSSION

4.1 Introduction

This chapter presents the results of the set objectives to indicate how they were achieved and what the results looked like. Some of the findings discussed in this chapter include how features were extracted, a comparative analysis of the existing image classification models with the feature descriptors, tuning parameters of the models to see if it improves the classification accuracy, and combining the best two models after a comparative analysis of the existing models. The developed model was validated by generating the confusion matrix and calculating the accuracy, precision, recall, and f1 score. Finally the researcher discussed the developed model components in detail and even went further to discuss why it produced the highest classification accuracy compared to existing ones.

4.2 Feature Extraction

Figure 4.1 illustrates how ORB extracts the key points from the common rust disease image. It clearly shows that the key points are more concentrated on the left-hand side of the image and that is that point that has the image features that clearly distinguish it from the other images.

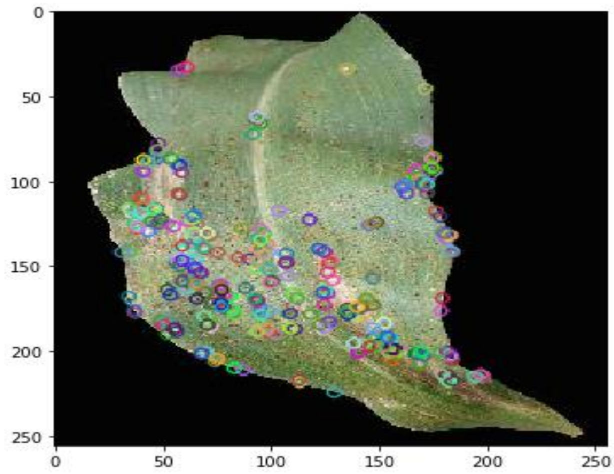

Figure 4.1 ORB Extracting Key Points from Common Rust Disease Image

KAZE feature descriptor method extracts features at the edge of the image since from Figure 4.2 it can be seen that the features are more concentrated at the edges of the image. This shows clearly that KAZE extracts image features mostly at the edges of the image.

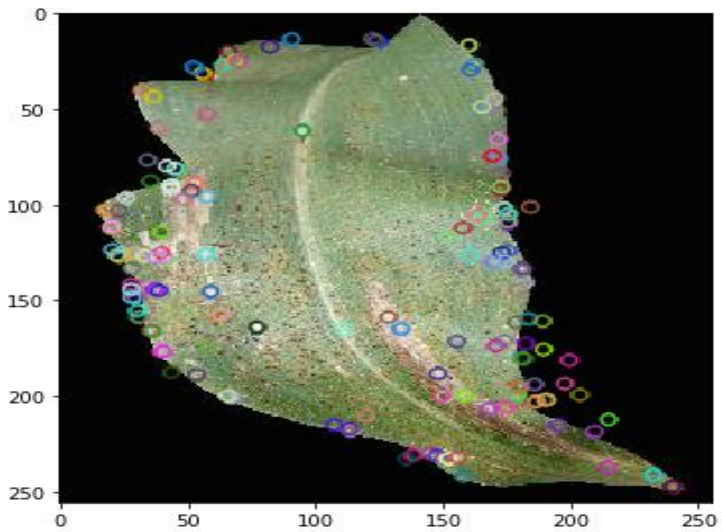

Figure 4.2 KAZE Extracting Key Points from Common Rust Disease Image

The healthy leaves, northern leaf blight, leaf spot, and common rust maize leaf diseases were the images contained in the dataset used during the research. During training, the researcher used 1600 images from each disease category and since four types of disease were used this resulted in 6400 images in total. The 4096 distinctive features were extracted using the three feature extraction methods from the total images used and the features were used to train the model hence acting as the input values to the machine learning algorithms. The features included color,edges, ridges, shape e.t.c .The features and labels were converted into a NumPy array which was usually faster than traditional python lists. Utilization of computer resources and speed was usually important and that was why NumPy array was most preferred in python. The main reason why this array was more efficient unlike lists was that in memory they are stored in one continuous place. The NumPy array both for features extracted and labels acted as input values for the image classification models. The association of each feature with their respective disease type to increase the generalizability and learning rate of the models was done during training. Note that the labels for image diseases before being converted to NumPy array were first converted to numerical values using the label encoder function in python. This process tried to normalize the non-numerical labels to numerical labels hence making the numerical labels faster for processing.

Table 4.1 Images Used During Feature Extraction.

Maize Disease Type	Images Used During Training	Images Used During Testing
Common Rust	1600	400
Healthy	1600	400
Leaf Spot	1600	400
Northern Leaf Blight	1600	400
Total Images Used	6400	1600

After downloading the dataset, a snipping tool was used to preview a sample of the images contained in each maize leaf disease category. Figures 4.3, 4.4, 4.5, and 4.6 show the images for common rust, healthy leaves, leaf spot, and northern leaf blight respectively as in the used dataset.

Figure 4.3 Common Rust

Figure 4.4 Healthy Leaves

Figure 4.5 Leaf Spot

Figure 4.6 Northern Leaf Blight

4.3 Image Classifiers

The Random Forest, Artificial Neural Network, Support Vector Classifier, K-Nearest Neighbors, Linear SVC, Logistic Regression, and Decision Tree were the image classifiers used in comparison with feature extraction methods. The researcher looked at the best feature extraction method by considering how they performed with the classifier in terms of accuracy and training time. Features were generated using KAZE, ORB, and HOG methods, and how the classifier performed in terms of accuracy with each of the feature extraction methods was measured. The features were extracted from 1600 images for each disease type to compare the accuracy performance of feature extraction methods with the classifiers. Table 4.2 shows the results of KAZE, ORB, and HOG feature descriptors accuracy performance with the image classification models.

Table 4.2 Feature Descriptors' Accuracy Performance with the Classifiers

	Accuracy		
Models	KAZE	ORB	HOG
Random Forest	0.675	0.376	0.730
Logistic Regression	0.695	0.361	0.790
K-Nearest Neighbors	0.609	0.397	0.680
Artificial Neural Networks	0.716	0.443	0.830
Linear SVC	0.690	0.361	0.730
Decision Tree	0.579	0.289	0.630
Support Vector Machine	0.706	0.423	0.820
Average Accuracy:	0.667	0.379	0.744

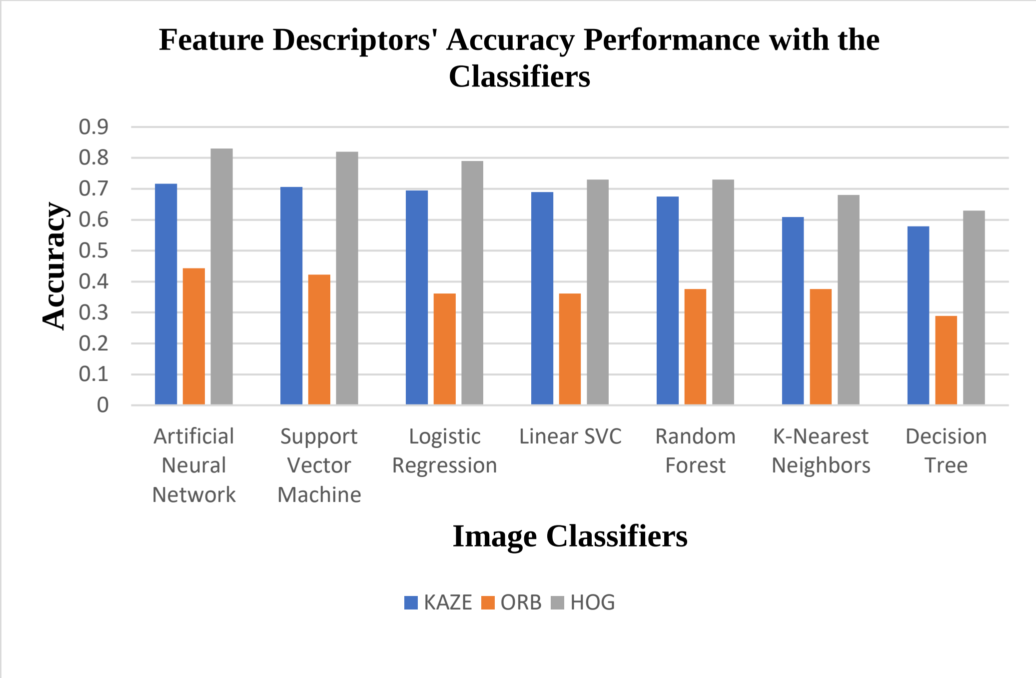

Figure 4.7 Feature Descriptors' Accuracy Performance with the Classifiers

As seen in Table 4.2 and Figure 4.7, the ORB feature extraction model performs badly with the classifiers. The average accuracy performance shown in Figure 4.7 and table 4.2 indicates ORB at 0.379, KAZE at 0.667, and HOG at 0.744 which is the best feature extraction method. The researcher decided to work with the HOG feature extraction method for the entire process of research due to its good performance.

Table 4.3 KAZE, ORB, and HOG Training Time with the Classifiers.

Models	KAZE	ORB	HOG
Random Forest	0.652460	0.487986	1.124743
Logistic Regression	0.640971	0.372273	0.767419
K-Nearest Neighbors	0.100726	0.098091	0.102654
Artificial Neural Networks	2.020296	3.488178	4.690680
Linear SVC	1.626427	0.379449	1.608843
Decision Tree	0.429705	0.255710	0.949240
Support Vector Machine	0.911466	0.947222	0.951323
Average Training Time (seconds)	0.911722	0.861273	1.456415

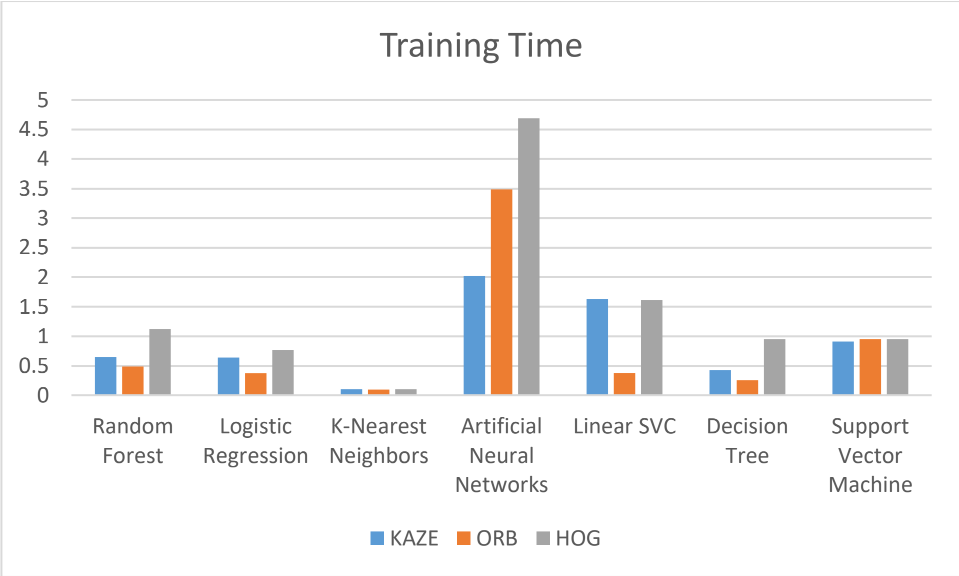

Figure 4.8 KAZE, ORB, and HOG Training Time with the Classifiers.

As seen in Table 4.3 and Figure 4.8, the ORB feature extraction method has the lowest training time among the base models. The ORB produces the average training time of 0.861273 seconds, KAZE at 0.911722 seconds, and HOG at 1.456415 seconds which indicates that it has the highest training time compared to others.

A comparison of training accuracy, test accuracy, and training time for Random Forest, Support Vector Classifier, Linear SVC, Logistic Regression, K-Nearest Neighbor, Artificial Neural Network, and Decision Tree was also done using cross-validation. The main reason was to see if cross-validation reduces the generalization error. For the above model, the training was done using the cross-validation method.

4.4 Hyper Parameter Tuning and Cross-Validation

The researcher looked at the optimal hyperparameters to work with the image classification models. During hyperparameter tuning the same data set size was used, then both models were tested on a dataset of 400 images. The hyperparameter is a parameter that was set for each image classification model before it started to learn from the given dataset. The hyperparameters were set for the classifier that accepts hyperparameter tuning and these were some of the hyperparameters for the classifiers using the HOG features;

The regularization parameter in Linear SVC is known as the C parameter and is the one that ensures that it classifies data points to the correct class. A low value of the C parameter is the one that classifies data correctly compared to a large value for C. When C=0.0001 it is considered an optimal value as shown in Figure 4.9 after grid search for optimal parameters is done and thus improves the classification accuracy.

Maximum iteration is the number of iterations run across all classes until the model learned and made accurate predictions. The optimal value for maximum iteration was 500 as shown in Figure 4.9.

```
linear_svc = LinearSVC(C=1.0, class_weight=None, dual=True,
fit_intercept=True,intercept_scaling=1, loss='squared_hinge', max_iter=1000,
multi_class='ovr', penalty='l2', random_state=None, tol=0.0001,verbose=0)


linear_svc_tuned = LinearSVC(C=0.0001, class_weight=None,        dual=True,
fit_intercept=True,  intercept_scaling=1,  loss='squared_hinge',  max_iter=500,
multi_class='ovr', penalty='l2', random_state=None, tol=0.01,verbose=0)
```

Figure 4.9 Default and Tuned Hyper-parameters values for Linear Support Vector Classifier

In logistic regression, the optimization algorithm is used to find the correct class to which the data points belong which in turn improves the classification accuracy. The solver parameter has different optimization algorithms assigned to it and based on Figure 4.10 the liblinear optimization algorithm was best suited hence resulting in better prediction results.

```
logistic_regression = LogisticRegression(C=1.0, class_weight=None,
dual=False,

fit_intercept=True,

           intercept_scaling=1, l1_ratio=None, max_iter=100,

           multi_class='auto', n_jobs=None, penalty='l2',

           random_state=None, solver='lbfgs', tol=0.0001, verbose=0,

           warm_start=False)

logistic_regression_tuned = LogisticRegression(C=0.01, class_weight=None,
dual=False, fit_intercept=True,

           intercept_scaling=1, l1_ratio=None, max_iter=500,

           multi_class='auto', n_jobs=None, penalty='l2',

           random_state=None, solver='liblinear', tol=0.01, verbose=0,

           warm_start=False)
```

Figure 4.10 Default and Tuned Hyper-parameters values for Logistic Regression

The class of the target point is usually selected depending on the number of the neighbors point through voting. To avoid the tie an odd number is usually preferred and 5 is usually the default value. The n_neighbors=11 as shown in Figure 4.11 and that was the optimal value for the tuned model which resulted in more accurate predictions.

```
k_nearest = KNeighborsClassifier(algorithm='auto', leaf_size=30,
metric='minkowski',

                metric_params=None, n_jobs=None, n_neighbors=5, p=2,

                weights='uniform')

k_nearest_tuned = KNeighborsClassifier(algorithm='auto', leaf_size=30,
metric='minkowski',

                metric_params=None, n_jobs=None, n_neighbors=11, p=1,

                weights='distance')
```

Figure 4.11 Default and Tuned Hyper-parameters values for K-Nearest Neighbor

The tolerance parameter was used to measure the point where the model has learned and it cannot learn beyond that point hence the training stops. The optimal parameter was when tol=1e-07 and that is when the model classified the data points to the correct class.

```
svc = SVC(C=1.0, break_ties=False, cache_size=200, class_weight=None,
coef0=0.0,    decision_function_shape='ovr', degree=3, gamma='scale',
kernel='rbf',
    max_iter=-1, probability=False, random_state=None, shrinking=True,
    tol=0.001, verbose=False)
svc_tuned = SVC(C=1.0, break_ties=False, cache_size=200, class_weight=None,
coef0=0.0,    decision_function_shape='ovr', degree=3, gamma='scale',
kernel='rbf',
    max_iter=-1, probability=False, random_state=None, shrinking=True,
    tol=1e-07, verbose=False)
```

Figure 4.12 Default and Tuned Hyper-parameters values for Support Vector Classifier

The best split of data points is considered by looking at the maximum features and the default value is always none. Figure 4.13 indicate that data was classified accurately when the maximum feature is 500.

```
decision_tree = DecisionTreeClassifier(ccp_alpha=0.0, class_weight=None,
criterion='gini',
                        max_depth=None, max_features=None, max_leaf_nodes=None,
                        min_impurity_decrease=0.0, min_impurity_split=None,
                        min_samples_leaf=1, min_samples_split=2,
                        min_weight_fraction_leaf=0.0, presort='deprecated',
                        random_state=None, splitter='best')


decision_tree_tuned = DecisionTreeClassifier(ccp_alpha=0.0,
class_weight=None, criterion='gini',
                        max_depth=None, max_features=500, max_leaf_nodes=200,
                        min_impurity_decrease=0.0, min_impurity_split=None,
                        min_samples_leaf=10, min_samples_split=10,
                        min_weight_fraction_leaf=0.0, presort='deprecated',
                        random_state=None, splitter='best')
```

Figure 4.13 Default and Tuned Hyper-parameters values for Decision Tree

The regularization term in Artificial Neural Network known as the alpha parameter reduces the over fitting problem by constraining the weights. The ANN performs well on unseen data when the over fitting problem is reduced by increasing the value of the alpha parameter to 0.001 after grid search for optimal parameters is done.

The forward and backward propagation done by the network is shown by the number of times the data passes through the algorithm and the parameter that regulates this is the maximum iteration. The network makes accurate predictions by iteratively adjusting the weights until the right weights are assigned to the network through forward and backward propagation. From the tuned results, 100 maximum iterations produced better prediction results.

The tolerance parameter is used to show the convergence point and this is the point the network has learned and it cannot learn beyond that point hence the algorithm has increased its generalizability thus it can make accurate predictions and at that particular point, the overfitting problem is reduced. Figure 4.14 shows that the training stopped and the network convergence reached when tol=1e-05.

```
neural_network = MLPClassifier(activation='relu', alpha=0.0001,
batch_size='auto', beta_1=0.9, beta_2=0.999, early_stopping=False, epsilon=1e-
08, hidden_layer_sizes=(100,), learning_rate='constant',
learning_rate_init=0.001, max_fun=15000, max_iter=200, momentum=0.9,
n_iter_no_change=10, nesterovs_momentum=True,  power_t=0.5,
random_state=None, shuffle=True, solver='adam',  tol=0.0001,
validation_fraction=0.1, verbose=False, warm_start=False)

neural_network_tuned = MLPClassifier(activation='relu', alpha=0.001,
batch_size='auto', beta_1=0.9, beta_2=0.999, early_stopping=False, epsilon=1e-
08, hidden_layer_sizes=(100,), learning_rate='constant',
learning_rate_init=0.001, max_fun=15000, max_iter=100, momentum=0.9,
n_iter_no_change=10, nesterovs_momentum=True, power_t=0.5,
random_state=None, shuffle=True, solver='adam',  tol=1e-05,
validation_fraction=0.1, verbose=False, warm_start=False)
```

Figure 4.14 Default and Tuned Hyper-parameters values for Artificial Neural Network

The training usually takes a lot of time when the number of trees is high and the n estimator is a parameter that is used to measure the trees used. The trees are used during voting and the average of each prediction done by the tree is calculated which reduces the generalization error experienced by one decision tree. Figure 4.15 shows that the random forest classifier classified images accurately when the n_estimators=200.

```
random_forest = RandomForestClassifier(bootstrap=True, ccp_alpha=0.0,
class_weight=None,
                       criterion='gini', max_depth=None, max_features='auto',
                       max_leaf_nodes=None, max_samples=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=2,
                       min_weight_fraction_leaf=0.0, n_estimators=100,
                       n_jobs=None, oob_score=False, random_state=None,
                       verbose=0, warm_start=False)
random_forest_tuned = RandomForestClassifier(bootstrap=True, ccp_alpha=0.0,
class_weight=None,
                       criterion='gini', max_depth=None, max_features=2000,
                       max_leaf_nodes=None, max_samples=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=2,
                       min_weight_fraction_leaf=0.0, n_estimators=200,
                       n_jobs=None, oob_score=False, random_state=None,
                       verbose=0, warm_start=False)
```

Figure 4.15 Default and Tuned Hyper-parameters values for Random Forest

4.5 Classification Report

The classification report enabled the researcher to know if the image classification models can classify the images well and be able to measure the quality of prediction the algorithms used. And this report was done for the algorithms with untuned and tuned parameters. The F1-Score, Recall, and Precision classification metrics are calculated and shown on the report. False and true negatives and false and true positives are used to calculate the metrics. The testing of the classifiers as shown in the Figures below was done using 400 images for each category of the disease from the testing data set and the results are shown in Fig 4.16, Fig 4.17, Fig 4.18, Fig 4.19, Fig 4.20, Fig 4.21, Fig 4.22, Fig 4.23, Fig 4.24, Fig 4.25, Fig 4.26, Fig 4.27, Fig 4.28 and Fig 4.29.

	Precision	Recall	F1-Score	Support
Common Rust	0.93	0.94	0.93	400
Healthy	0.64	0.67	0.65	400
Leaf Spot	0.76	0.72	0.74	400
Northern Leaf Blight	0.60	0.60	0.60	400
Accuracy			0.73	1600
Macro avg	0.73	0.73	0.73	1600
Weighted avg	0.73	0.73	0.73	1600

Figure 4.16 Linear SVC Classification Report

The linear SVC classifier with optimal parameters classified the common rust images with the highest accuracy of 0.98 compared to the normal Linear SVC which had an accuracy score of 0.93 as shown in Figure 4.16. The regularization parameter in Linear SVC is known as the C parameter and is the one that ensured that it classified data points to the correct class. A low value of the C parameter ensures that the model classifies data correctly compared to a large value for C. When C=0.0001 it was considered an optimal value as shown in Figure 4.9 and thus improved the classification accuracy.

	Precision	Recall	F1-Score	Support
Common Rust	0.98	0.86	0.91	400
Healthy	0.73	0.74	0.73	400
Leaf Spot	0.81	0.78	0.80	400
Northern Leaf Blight	0.68	0.78	0.73	400
Accuracy			0.79	1600
Macro avg	0.80	0.79	0.79	1600
Weighted avg	0.80	0.79	0.79	1600

Figure 4.17 Tuned Linear SVC Classification Report

Logistic regression with optimal parameters improves the classification accuracy by 0.04. In logistic regression, the optimization algorithm is used to find the correct class to the data points belong which in turn improves the classification accuracy. The solver parameter has different optimization algorithms assigned to it and based on Figure 4.10 the liblinear optimization algorithm was best suited hence resulting in better prediction results.

	Precision	Recall	F1-Score	Support
Common Rust	0.94	0.94	0.94	400
Healthy	0.73	0.78	0.75	400
Leaf Spot	0.81	0.77	0.79	400
Northern Leaf Blight	0.69	0.68	0.69	400
Accuracy			0.79	1600
Macro avg	0.79	0.79	0.79	1600
Weighted avg	0.79	0.79	0.79	1600

Figure 4.18 Logistic Regression Classification Report

	Precision	Recall	F1-Score	Support
Common Rust	0.98	0.90	0.94	400
Healthy	0.75	0.82	0.78	400
Leaf Spot	0.86	0.88	0.87	400
Northern Leaf Blight	0.77	0.74	0.76	400
Accuracy			0.83	1600
Macro avg	0.84	0.83	0.84	1600
Weighted avg	0.84	0.83	0.84	1600

Figure 4.19 Tuned Logistic Regression Classification Report

K-Nearest Neighbor Model with n-neighbors of 11 results in a precision score of 1.00. The class of the target point is usually selected depending on the number of the neighbors' point through voting. To avoid the tie an odd number is usually preferred and 5 is usually the default value. The n_neighbors=11 as shown in Figure 4.11 and was the optimal value for the tuned model which resulted in more accurate predictions.

	Precision	Recall	F1-Score	Support
Common Rust	0.99	0.86	0.92	400
Healthy	0.51	0.83	0.63	400
Leaf Spot	0.70	0.73	0.71	400
Northern Leaf Blight	0.68	0.29	0.41	400
Accuracy			0.68	1600
Macro avg	0.72	0.68	0.67	1600
Weighted avg	0.72	0.68	0.67	1600

Figure 4.20 K-Nearest Neighbor Classification Report

	Precision	Recall	F1-Score	Support
Common Rust	1.00	0.90	0.95	400
Healthy	0.89	0.78	0.83	400
Leaf Spot	0.88	0.86	0.87	400
Northern Leaf Blight	0.74	0.92	0.82	400
Accuracy			0.86	1600
Macro avg	0.88	0.86	0.87	1600
Weighted avg	0.88	0.86	0.87	1600

Figure 4.21 Tuned K-Nearest Neighbor Classification Report

The optimal parameter for Support Vector Machine makes it at a global minimum to classify the images to the right class they belong to. As seen in Figure 4.23 the precision for common rust disease is 1.00 which indicates that the 400 images of common rust disease were all classified accurately. The tolerance parameter was used to measure the point where the model has learned and it cannot learn beyond that point hence the training stops. The optimal parameter was when tol=1e-07 and that is when the model classified the data points to the correct class.

	Precision	Recall	F1-Score	Support
Common Rust	0.98	0.94	0.96	400
Healthy	0.79	0.82	0.80	400
Leaf Spot	0.92	0.72	0.81	400
Northern Leaf Blight	0.67	0.81	0.73	400
Accuracy			0.82	1600
Macro avg	0.84	0.82	0.83	1600
Weighted avg	0.84	0.82	0.83	1600

Figure 4.22 Support Vector Classifier Classification Report

	Precision	Recall	F1-Score	Support
Common Rust	1.00	0.90	0.95	400
Healthy	0.89	0.82	0.85	400
Leaf Spot	0.89	0.84	0.87	400
Northern Leaf Blight	0.73	0.90	0.80	400
Accuracy			0.87	1600
Macro avg	0.88	0.86	0.87	1600
Weighted avg	0.88	0.86	0.87	1600

Figure 4.23 Tuned Support Vector Classifier Classification Report

The Decision Tree model does not perform well with the test dataset because of the high generalization error that makes it misclassify a lot of images. The model works well when classifying data belonging to two classes as compared to more than two classes.

	Precision	Recall	F1-Score	Support
Common Rust	0.83	0.66	0.74	400
Healthy	0.59	0.63	0.61	400
Leaf Spot	0.68	0.73	0.70	400
Northern Leaf Blight	0.49	0.52	0.50	400
Accuracy			0.63	1600
Macro avg	0.65	0.63	0.64	1600
Weighted avg	0.65	0.63	0.64	1600

Figure 4.24 Decision Tree Classification Report

	Precision	Recall	F1-Score	Support
Common Rust	0.88	0.58	0.70	400
Healthy	0.49	0.78	0.60	400
Leaf Spot	0.65	0.64	0.65	400
Northern Leaf Blight	0.59	0.46	0.52	400
Accuracy			0.61	1600
Macro avg	0.65	0.61	0.62	1600
Weighted avg	0.65	0.61	0.62	1600

Figure 4.25 Tuned Decision Tree Classification Report

The forward and backward propagation done by the network is shown by the number of times the data passes through the algorithm and the parameter that regulates this is the maximum iteration. The network makes accurate predictions by iteratively adjusting the weights until the right weights are assigned to the network through forward and backward propagation. From the tuned results 100 maximum iterations produced better prediction results.

The tolerance parameter is used to show the convergence point and this is the point the network has learned and it cannot learn beyond that point hence the algorithm has increased its generalizability thus it can make accurate predictions and at that particular point, the overfitting problem is reduced.

The regularization term in Artificial Neural Network known as the alpha parameter reduces the overfitting problem by constraining the weights. The ANN performs well on unseen data when the overfitting problem is reduced by increasing the value of the alpha parameter.

The tuned three parameters for Artificial Neural Network are the ones that made the model classify maize leaf disease images with an accuracy of 0.88 as shown in Figure 4.27.

	Precision	Recall	F1-Score	Support
Common Rust	0.97	0.96	0.96	400
Healthy	0.78	0.83	0.80	400
Leaf Spot	0.87	0.79	0.83	400
Northern Leaf Blight	0.72	0.76	0.74	400
Accuracy			0.83	1600
Macro avg	0.84	0.83	0.83	1600
Weighted avg	0.84	0.83	0.83	1600

Figure 4.26 Artificial Neural Network Classification Report

	Precision	Recall	F1-Score	Support
Common Rust	0.98	0.92	0.95	400
Healthy	0.84	0.86	0.85	400
Leaf Spot	0.88	0.90	0.89	400
Northern Leaf Blight	0.82	0.84	0.83	400
Accuracy			0.88	1600
Macro avg	0.88	0.88	0.88	1600
Weighted avg	0.88	0.88	0.88	1600

Figure 4.27 Tuned Artificial Neural Network Classification Report

The training usually takes a lot of time when the number of trees is high and the n estimator is a parameter that was used to measure the trees used. The trees are used during voting and the average of each prediction done by the tree was calculated which reduced the generalization error experienced by one decision tree. Figure 4.15 shows that the random forest classifier classified images accurately when the n_estimators=200.

	Precision	Recall	F1-Score	Support
Common Rust	0.96	0.65	0.77	400
Healthy	0.65	0.81	0.72	400
Leaf Spot	0.87	0.69	0.77	400
Northern Leaf Blight	0.62	0.79	0.69	400
Accuracy			0.73	1600
Macro avg	0.77	0.73	0.74	1600
Weighted avg	0.77	0.73	0.74	1600

Figure 4.28 Random Forest Classification Report

	Precision	Recall	F1-Score	Support
Common Rust	1.00	0.94	0.97	400
Healthy	0.84	0.92	0.88	400
Leaf Spot	0.86	0.84	0.85	400
Northern Leaf Blight	0.80	0.78	0.79	400
Accuracy			0.86	1600
Macro avg	0.86	0.86	0.86	1600
Weighted avg	0.86	0.86	0.86	1600

Figure 4.29 Tuned Random Forest Classification Report

Table 4.4 Images Used During Feature Extraction for Enhanced Model

Maize Disease Type	Images Used During Training	Images Used During Testing
Common Rust	1600	400
Healthy	1600	400
Leaf Spot	1600	400
Northern Leaf Blight	1600	400
Total Images Used	6400	1600

	Precision	Recall	F1-Score	Support
Common Rust	0.99	0.97	0.98	400
Healthy	0.95	0.94	0.95	400
Leaf Spot	0.95	0.95	0.95	400
Northern Leaf Blight	0.88	0.91	0.92	400
Accuracy			0.95	1600
Macro avg	0.95	0.95	0.95	1600
Weighted avg	0.95	0.95	0.95	1600

Figure 4.30 Enhanced Model Classification Report

The training took a lot of time for the combined two models since the generalization of the two models with data is more compared to single ones. The hyperplane component in the support vector machine increased the classification accuracy by clearly segregating data between different classes. Figure 4.30 shows that the developed model classified images accurately when the hyperplane component replaced the softmax layer in the artificial neural network.

Table 4.5 Classification Accuracy for Default and Tuned Models

Models	Untuned Model Accuracy	Tuned Model Accuracy
Random Forest	0.73	0.850
Logistic Regression	0.790	0.830
K-Nearest Neighbors	0.68	0.860
Artificial Neural Networks	0.830	0.890
Linear SVC	0.73	0.790
Decision Tree	0.63	0.730
Support Vector Machine	0.82	0.870
Average Accuracy:	0.713	0.831

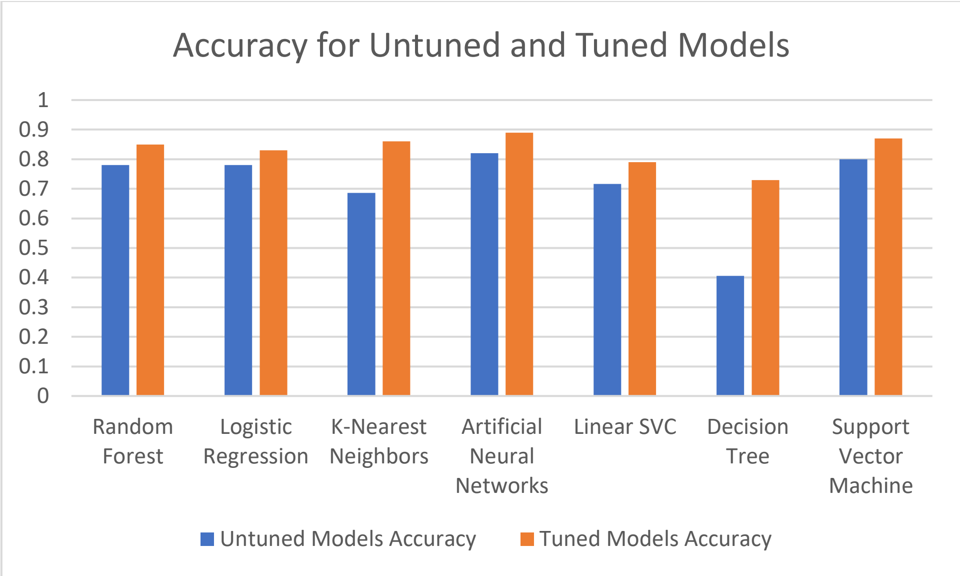

Figure 4.31 Accuracy for Default and Tuned Models

Table 4.5 and Figure 4.31 above indicate that there was improved performance in terms of accuracy for all of the tuned models. The average accuracy as per table 4.5 was for the tuned models which indicates that the tuned models outperformed the untuned models. The accuracy was obtained by using the untuned and tuned models together with the HOG feature extraction method. These helped the researcher to get more insights into the models' performance thus guiding him on which models to combine to improve more on the image classification accuracy. The parameters used by each model are usually internal to the model and from the given data set their values were estimated to know the best parameter values that can work based on a given dataset. The parameters that are set to these image classification models usually affect the prediction accuracy of these models. When good parameters are set for each model then definitely the models will have better predictions on the new dataset they are subjected to.

Table 4.6 Classification Accuracy for Enhanced Model

Model	Accuracy	F1 Score	Precision	Recall
Artificial Neural Networks + Support Vector Machine	0.95	0.95	0.95	0.95

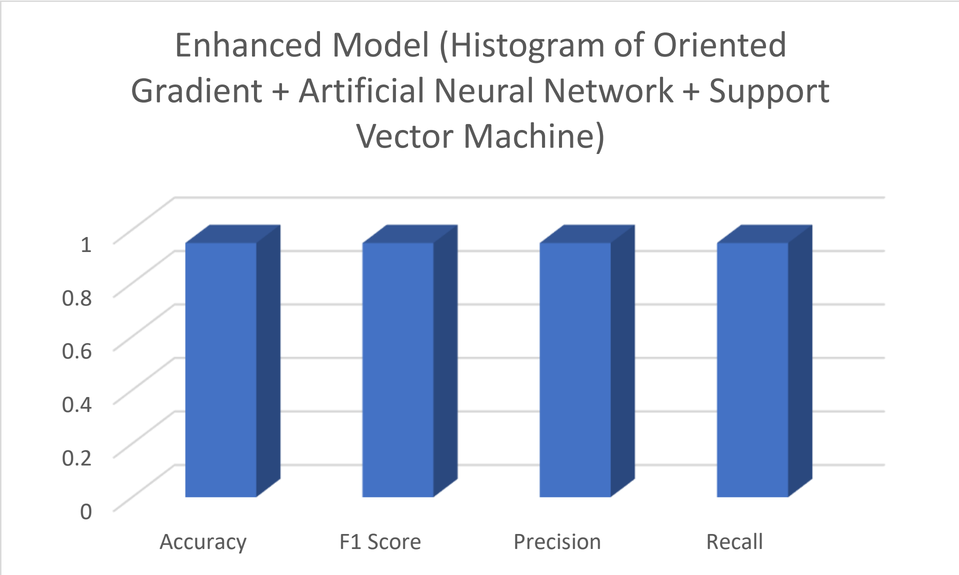

Figure 4.32 Metrics for the Enhanced Model

Table 4.6 and Figure 4.32 shows clearly that the developed model produced an average accuracy of 0.95 which is better compared to existing models. The hyperplane component in the support vector machine increased the classification accuracy by clearly segregating data between different classes. It can be seen clearly from table 4.6 and Figure 4.32 that the developed model classified images accurately when the hyperplane component replaced the softmax layer in the artificial neural network.

4.6 Confusion Matrix

The matrix was used to be able to visualize how each model performed based on the test data set and the main aim was to visualize in terms of how many images are classified and misclassified by each model. The following are some of the confusion matrices for the models used.

Table 4.7 Linear SVC Confusion Matrix Visualization

	Predicted			
Known	Common Rust	Healthy	Leaf Spot	Northern Leaf Blight
Common Rust	371	13	2	14
Healthy	9	251	41	99
Leaf Spot	5	37	311	47
Northern Leaf Blight	11	83	68	238

Tables 4.7 and 4.8 indicate the confusion matrix for linear svc, the numbers placed diagonally for each table are the prediction for the disease category for northern leaf blight, leaf spot, healthy, and common rust respectively. It can also be seen from tables 4.7 and 4.8 that common rust disease was the one which was well predicted with the highest common rust image predictions of 371 and 377 images disease for linear svc and tuned linear svc respectively.

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

$$Accuracy = \frac{Number\ of\ correct\ predictions}{Total\ number\ of\ predictions\ made}$$

Linear SVC Accuracy $= \frac{371+251+311+238}{1600}$

$= 0.73$

Linear SVC Precision$= \frac{True\ Positives}{True\ Positives+False\ Positives}$

$= \frac{1171}{1171+429}$

$$= 0.73$$

Linear SVC Recall=$\frac{True\ Positives}{True\ Positives+False\ Negatives}$

$$= \frac{1171}{1171+429}$$

$$= 0.73$$

Linear SVC F1 Score =$2 * \frac{Precision*Recall}{Precision+Recall}$

$$= 2 * \frac{0.73 * 0.73}{0.73 + 0.73}$$

$$= 0.73$$

Table 4.8 Linear SVC Tuned Confusion Matrix Visualization

	Predicted			
Known	Common Rust	Healthy	Leaf Spot	Northern Leaf Blight
Common Rust	377	3	6	14
Healthy	14	269	25	92
Leaf Spot	2	24	362	12
Northern Leaf Blight	15	56	77	252

The confusion matrix was used to calculate the f1 score, recall, precision, and accuracy as shown by the formula for the equation below.

Linear SVC Tuned Accuracy =$\frac{377+269 + 362 +252}{1600}$

$$= 0.79$$

Linear SVC Tuned Precision=$\frac{True\ Positives}{True\ Positives+False\ Positives}$

$= \frac{1260}{1535+65}$

$= 0.79$

Linear SVC Tuned Recall=$\frac{True\ Positives}{True\ Positives+False\ Negatives}$

$= \frac{1260}{1535+65}$

$= 0.79$

Linear SVC Tuned F1 Score =$2 * \frac{Precision*Recall}{Precision+Recall}$

$= 2 * \frac{0.79 *0.79}{0.79 +0.79}$

$= 0.79$

Tables 4.9 and 4.10 indicate the confusion matrix for logistic regression, the numbers placed diagonally for each table are the predictions for the disease category for northern leaf blight, leaf spot, healthy, and common rust respectively. It can also be seen from tables 4.9 and 4.10 that northern leaf blight and healthy images were the ones that were badly predicted. Logistic Regression predicted 283 and 280 images for healthy and northern leaf blight disease respectively and the tuned one predicted 291 and 294 images for healthy and northern leaf blight disease respectively.

Table 4.9 Logistic Regression Confusion Matrix Visualization

	Predicted			
Known	Common Rust	Healthy	Leaf Spot	Northern Leaf Blight
Common Rust	385	10	3	12
Healthy	10	283	30	77
Leaf Spot	3	30	327	40
Northern Leaf Blight	11	49	60	280

Logistic Regression Accuracy $=\frac{385+283+327+280}{1600}$

$= 0.79$

Logistic Regression Precision$=\frac{True\ Positives}{True\ Positives+False\ Positives}$

$= \frac{1275}{1275+325}$

$= 0.79$

Logistic Regression Recall$=\frac{True\ Positives}{True\ Positives+False\ Negatives}$

$= \frac{1275}{1275+325}$

$= 0.79$

Logistic Regression F1 Score $=2 * \frac{Precision*Recall}{Precision+Recall}$

$= 2 * \frac{0.79 * 0.79}{0.79 + 0.79}$

$= 0.79$

Table 4.10 Logistic Regression Tuned Confusion Matrix Visualization

	Predicted			
Known	Common Rust	Healthy	Leaf Spot	Northern Leaf Blight
Common Rust	385	5	2	8
Healthy	10	291	23	76
Leaf Spot	3	18	358	21
Northern Leaf Blight	7	49	50	294

Logistic Regression Tuned Accuracy $= \frac{385+291+358+294}{1600}$

$= 0.83$

Logistic Regression Tuned Precision$= \frac{True\ Positives}{True\ Positives + False\ Positives}$

$= \frac{1328}{1328+272}$

$= 0.83$

Logistic Regression Tuned Recall$= \frac{True\ Positives}{True\ Positives + False\ Negatives}$

$= \frac{1328}{1328+272}$

$= 0.83$

Logistic Regression Tuned F1 Score $= 2 * \frac{Precision * Recall}{Precision + Recall}$

$$= 2 * \frac{0.83 * 0.83}{0.83 + 0.83}$$

$$= 0.83$$

The true positives which indicate the predicted values for K-Nearest Neighbors are shown diagonally in tables 4.11 and 4.12 for the disease category for northern leaf blight, leaf spot, healthy, and common rust. The predictions include 396, 78, 272, and 344 for the above-named diseases respectively. It can also be seen from tables 4.11 and 4.12 that healthy images were the ones that were badly predicted. K-Nearest Neighbors predicted 78 images for healthy and the tuned one predicted 292 images for the healthy category.

Table 4.11 K-Nearest Neighbors Confusion Matrix Visualization

	Predicted			
Known	Common Rust	Healthy	Leaf Spot	Northern Leaf Blight
Common Rust	396	0	3	1
Healthy	40	78	79	203
Leaf Spot	11	41	272	76
Northern Leaf Blight	5	25	26	344

K-Nearest Neighbors Accuracy $= \frac{396 + 78 + 272 + 344}{1600}$

$$= 0.68$$

K-Nearest Neighbors Precision $= \frac{True\ Positives}{True\ Positives + False\ Positives}$

$= \frac{1090}{1090+690}$

$= 0.68$

K-Nearest Neighbors Recall=$\frac{True\ Positives}{True\ Positives+False\ Negatives}$

$= \frac{1090}{1090+690}$

$= 0.68$

K-Nearest Neighbors F1 Score =$2 * \frac{Precision*Recall}{Precision+Recall}$

$= 2 * \frac{0.68 * 0.68}{0.68 + 0.68}$

$= 0.68$

Table 4.12 K-Nearest Neighbors Tuned Confusion Matrix Visualization

	Predicted			
Known	Common Rust	Healthy	Leaf Spot	Northern Leaf Blight
Common Rust	399	0	1	0
Healthy	14	292	27	67
Leaf Spot	9	19	329	43
Northern Leaf Blight	6	12	27	355

Tuned K-Nearest Neighbors Accuracy =$\frac{399+292 + 329 +355}{1600}$

$= 0.86$

Tuned K-Nearest Neighbors Precision=$\frac{True\ Positives}{True\ Positives+False\ Positives}$

$$= \frac{1375}{1375+225}$$

$$= 0.86$$

Tuned K-Nearest Neighbors Recall=$\frac{True\ Positives}{True\ Positives+False\ Negatives}$

$$= \frac{1375}{1375+225}$$

$$= 0.86$$

Tuned K-Nearest Neighbors F1 Score =$2 * \frac{Precision*Recall}{Precision+Recall}$

$$= 2 * \frac{0.86 * 0.86}{0.86 + 0.86}$$

$$= 0.86$$

The values not in diagonal matrix indicate the misclassified images which resulted in a total of 284 images for the support vector classifier. The true positives which indicate the predicted values for the support vector classifier are shown diagonally in tables 4.13 and 4.14 for the disease category for northern leaf blight, leaf spot, healthy, and common rust. The predictions for the tuned model include 397, 333, 387, and 273 for the above-named diseases respectively. It can also be seen from tables 4.13 and 4.14 that common rust images were correctly classified compared to other diseases. The support vector classifier predicted 392 images and the tuned one predicted 397 images for the common rust disease category.

Table 4.13 Support Vector Classifier Confusion Matrix Visualization

	Predicted			
Known	Common Rust	Healthy	Leaf Spot	Northern Leaf Blight
Common Rust	392	0	2	6
Healthy	3	310	20	67
Leaf Spot	1	20	375	4
Northern Leaf Blight	22	51	88	239

Support Vector Classifier Accuracy $=\frac{392+310+375+239}{1600}$

$= 0.82$

Support Vector Classifier Precision$=\frac{True\ Positives}{True\ Positives+False\ Positives}$

$= \frac{1316}{1316+284}$

$= 0.82$

Support Vector Classifier Recall$=\frac{True\ Positives}{True\ Positives+False\ Negatives}$

$= \frac{1316}{1316+284}$

$= 0.82$

Support Vector Classifier F1 Score $=2 * \frac{Precision*Recall}{Precision+Recall}$

$= 2 * \frac{0.82 * 0.82}{0.82 + 0.82}$

$= 0.82$

Table 4.14 Support Vector Classifier Tuned Confusion Matrix Visualization

	Predicted			
Known	Common Rust	Healthy	Leaf Spot	Northern Leaf Blight
Common Rust	397	0	0	3
Healthy	5	333	18	44
Leaf Spot	1	11	387	1
Northern Leaf Blight	18	36	73	273

Tuned Support Vector Classifier Accuracy $=\frac{397+333+387+273}{1600}$

$= 0.87$

Tuned Support Vector Classifier Precision$=\frac{True\ Positives}{True\ Positives+False\ Positives}$

$= \frac{1390}{1390+210}$

$= 0.87$

Tuned Support Vector Classifier Recall$=\frac{True\ Positives}{True\ Positives+False\ Negatives}$

$= \frac{1390}{1390+210}$

$= 0.87$

Support Vector Classifier F1 Score $=2 * \frac{Precision*Recall}{Precision+Recall}$

$= 2 * \frac{0.87 * 0.87}{0.87 + 0.87}$

$= 0.87$

The decision tree is one of the image classification models that use the concept of entropy to classify the images. The main reason why it does not classify data well is that it suffers from the overfitting problem which is a result of the model not generalizing the data well. The true positives which indicate the predicted values for the decision tree are shown diagonally in tables 4.15 and 4.16 for the disease category for northern leaf blight, leaf spot, healthy, and common rust. The predictions include 347, 217, 259, and 187 for the above-named diseases respectively. It can also be seen from tables 4.15 and 4.16 that northern leaf blight images were the ones that were badly predicted. The decision tree predicted 187 images and the tuned one predicted 232 images for the northern leaf blight maize leaf disease category.

Table 4.15 Decision Tree Confusion Matrix Visualization

	Predicted			
Known	Common Rust	Healthy	Leaf Spot	Northern Leaf Blight
Common Rust	347	12	19	22
Healthy	39	217	33	111
Leaf Spot	30	46	259	65
Northern Leaf Blight	67	91	55	187

Decision Tree Accuracy $=\frac{347+217+259+187}{1600}$

$= 0.63$

Decision Tree Precision $=\frac{True\ Positives}{True\ Positives+False\ Positives}$

$= \frac{1010}{1010+590}$

= 0.63

Decision Tree Recall=$\frac{True\ Positives}{True\ Positives+False\ Negatives}$

$= \frac{1010}{1010+590}$

= 0.63

Decision Tree F1 Score =$2 * \frac{Precision*Recall}{Precision+Recall}$

$= 2 * \frac{0.63 * 0.63}{0.63 + 0.63}$

= 0.63

Table 4.16 Decision Tree Tuned Confusion Matrix Visualization

	Predicted			
Known	Common Rust	Healthy	Leaf Spot	Northern Leaf Blight
Common Rust	371	5	11	13
Healthy	41	238	45	76
Leaf Spot	19	26	322	33
Northern Leaf Blight	44	71	53	232

Tuned Decision Tree Accuracy =$\frac{371+238+322+232}{1600}$

= 0.73

Tuned Decision Tree Precision=$\frac{True\ Positives}{True\ Positives+False\ Positives}$

$= \frac{1163}{1163+437}$

$$= 0.73$$

$$\text{Tuned Decision Tree Recall}=\frac{True\ Positives}{True\ Positives+False\ Negatives}$$

$$= \frac{1163}{1163+437}$$

$$= 0.73$$

$$\text{Tuned Decision Tree F1 Score} =2 * \frac{Precision*Recall}{Precision+Recall}$$

$$= 2 * \frac{0.73 *0.73}{0.73 +0.73}$$

$$= 0.73$$

The weights and biased are adjusted by ANN during the forward and the backward propagation to get the right values of weight and bias that gives the correct predictions. The network also has more neurons to store more data during training which assists in image classification. Tables 4.17 and 4.18 indicate the confusion matrix for artificial neural network, the numbers placed diagonally for each table are the prediction for the disease category for northern leaf blight, leaf spot, healthy, and common rust respectively. It can also be seen from tables 4.17 and 4.18 that common rust disease was the one which was well predicted with the highest common rust image predictions of 389 and 394 images disease for artificial neural network and tuned artificial neural network respectively.

Table 4.17 Artificial Neural Network Confusion Matrix Visualization

	Predicted			
Known	Common Rust	Healthy	Leaf Spot	Northern Leaf Blight
Common Rust	389	0	1	10
Healthy	6	307	26	61
Leaf Spot	2	18	355	25
Northern Leaf Blight	9	51	59	281

Artificial Neural Network Accuracy $=\frac{389+307+355+281}{1600}$

$= 0.83$

Artificial Neural Network Precision$=\frac{True\ Positives}{True\ Positives+False\ Positives}$

$= \frac{1332}{1332+268}$

$= 0.83$

Artificial Neural Network Recall$=\frac{True\ Positives}{True\ Positives+False\ Negatives}$

$= \frac{1332}{1332+268}$

$= 0.83$

Artificial Neural Network F1 Score $=2 * \frac{Precision*Recall}{Precision+Recall}$

$= 2 * \frac{0.83 * 0.83}{0.83 + 0.83}$

$= 0.83$

Table 4.18 Artificial Neural Network Tuned Confusion Matrix Visualization

	Predicted			
Known	Common Rust	Healthy	Leaf Spot	Northern Leaf Blight
Common Rust	394	0	1	5
Healthy	4	341	16	39
Leaf Spot	1	14	367	18
Northern Leaf Blight	8	31	39	322

Tuned Artificial Neural Network Accuracy $=\frac{394+341+367+322}{1600}$

$= 0.89$

Tuned Artificial Neural Network Precision$=\frac{True\ Positives}{True\ Positives+False\ Positives}$

$= \frac{1424}{1424+176}$

$= 0.89$

Tuned Artificial Neural Network Recall$=\frac{True\ Positives}{True\ Positives+False\ Negatives}$

$= \frac{1424}{1424+176}$

$= 0.89$

Tuned Artificial Neural Network F1 Score $=2 * \frac{Precision*Recall}{Precision+Recall}$

$= 2 * \frac{0.89 * 0.89}{0.89 + 0.89}$

$= 0.89$

The Random Forest is made up of more decision trees and predicts results by creating decision trees on the data samples and picking the best decision tree that predicted the results with high accuracy through voting. It reduces the overfitting problem by averaging the result from every decision tree hence concluding the best result that is why it is called an ensemble method. Tables 4.19 and 4.20 indicate the confusion matrix for the random forest, the numbers paced diagonally for each table are the predictions for the disease category for northern leaf blight, leaf spot, healthy, and common rust respectively. It can also be seen from tables 4.19 and 4.20 that northern leaf blight and healthy images were the ones that were badly predicted. The Random Forest predicted 222 and 205 images for healthy and northern leaf blight disease respectively and the tuned one predicted 316 and 248 images for healthy and northern leaf blight disease respectively.

Table 4.19 Random Forest Confusion Matrix Visualization

	Predicted			
Known	Common Rust	Healthy	Leaf Spot	Northern Leaf Blight
Common Rust	389	0	10	1
Healthy	71	222	37	70
Leaf Spot	5	22	359	14
Northern Leaf Blight	66	54	75	205

Random Forest Accuracy $=\frac{389+222+359+205}{1600}$

$= 0.73$

Random Forest Precision$=\frac{True\ Positives}{True\ Positives+False\ Positives}$

$$= \frac{1175}{1175+425}$$

$$= 0.73$$

Random Forest Recall=$\frac{True\ Positives}{True\ Positives+False\ Negatives}$

$$= \frac{1175}{1175+425}$$

$$= 0.73$$

Random Forest F1 Score =2 $* \frac{Precision*Recall}{Precision+Recall}$

$$= 2 * \frac{0.73 *0.73}{0.73 +0.73}$$

$$= 0.73$$

Table 4.20 Random Forest Tuned Confusion Matrix Visualization

	Predicted			
Known	Common Rust	Healthy	Leaf Spot	Northern Leaf Blight
Common Rust	398	0	1	1
Healthy	9	316	15	60
Leaf Spot	1	4	392	3
Northern Leaf Blight	44	47	61	248

Tuned Random Forest Accuracy =$\frac{398+316 + 392 +248}{1600}$

$$= 0.85$$

Tuned Random Forest Precision=$\frac{True\ Positives}{True\ Positives+False\ Positives}$

$$= \frac{1354}{1354+246}$$

$= 0.85$

Tuned Random Forest Recall=$\frac{True\ Positives}{True\ Positives+False\ Negatives}$

$= \frac{1354}{1354+246}$

$= 0.85$

Tuned Random Forest F1 Score =$2 * \frac{Precision*Recall}{Precision+Recall}$

$= 2 * \frac{0.85 *0.85}{0.85 +0.85}$

$= 0.85$

The 1600 images were used for training for each disease category resulting in a total of 6400. Testing the model was part of the validation process which involved 400 images for each maize leaf disease category resulting in 1600 images in total as clearly shown in table 4.21 below.

Table 4.21 Images Used for Enhanced Model

Maize Disease Type	Images Used During Training	Images Used During Testing
Common Rust	1600	400
Healthy	1600	400
Leaf Spot	1600	400
Northern Leaf Blight	1600	400
Total Images Used	6400	1600

The combination of Artificial Neural Network + Support Vector Machine produced good results and the main reason was that the SVM replaced the softmax layer in the Artificial Neural Network, and hence it had the capability of minimizing the generalization error on unseen data which resulted in better prediction results. The true positives which indicate the predicted values for the developed model are shown diagonally in table 4.22 with a background of blue color for the disease category for northern leaf blight, leaf spot, healthy, and common rust. The predictions include 398, 371, 384, and 382 for the above-named diseases respectively. It can also be seen from table 4.22 that healthy images were the ones that were badly predicted. The developed model predicted 371 images for healthy category.

Table 4.22 Enhanced Model Confusion Matrix Visualization

	Predicted			
Known	Common Rust	Healthy	Leaf Spot	Northern Leaf Blight
Common Rust	398	0	0	2
Healthy	0	371	11	18
Leaf Spot	0	8	384	8
Northern Leaf Blight	0	15	13	382

Developed Model Accuracy $=\frac{398+371+384+382}{1600}$

$= 0.95$

Precision$=\frac{True\ Positives}{True\ Positives+False\ Positives}$

$= \frac{1535}{1535+65}$

$= 0.95$

Recall$=\frac{True\ Positives}{True\ Positives+False\ Negatives}$

$= \frac{1535}{1535+65}$

$= 0.95$

$$\text{F1 Score} = 2 * \frac{Precision * Recall}{Precision + Recall}$$

$$= 2 * \frac{0.95 * 0.95}{0.95 + 0.95}$$

$= 0.95$

4.7 Overall Classification Accuracy

After finding the classification report for each model the overall classification metrics were determined and this helped us know the best performing model in terms of classifying the images from the test data set. The developed model emerged as the best with a classification accuracy of 0.95 while the worst model was the decision tree with an accuracy score of 0.73. The following are the results gotten for each model;

Table 4.23 Models' Accuracy

Models	Accuracy
Artificial Neural Networks + Support Vector Machine	0.95
Artificial Neural Networks	0.89
Support Vector Machine	0.87
K-Nearest Neighbors	0.86
Random Forest	0.85
Logistic Regression	0.83
Linear SVC	0.79
Decision Tree	0.73

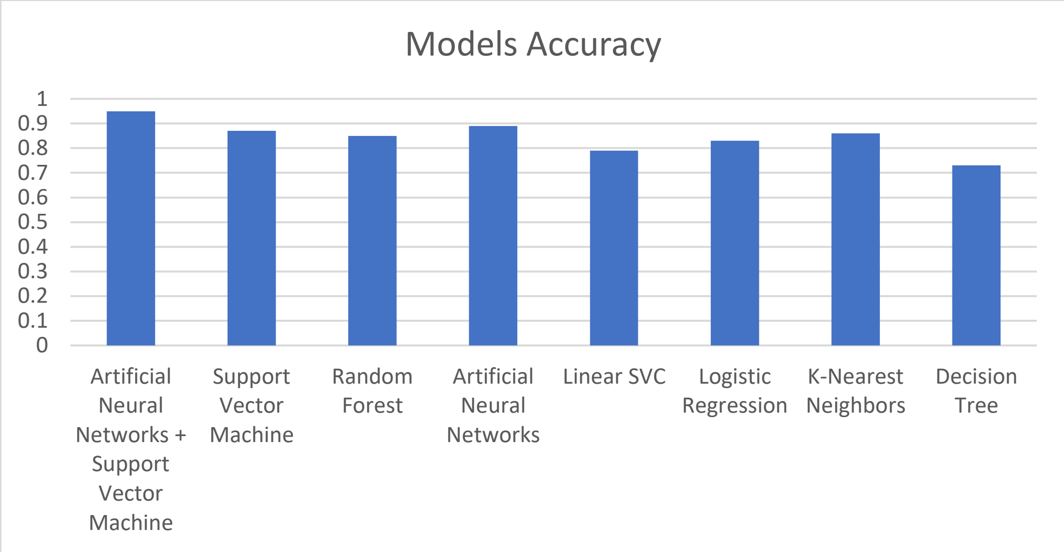

Figure 4.33 Models' Accuracy

The results in Figure 4.33 above show the best performing model (Artificial Neural Networks + Support Vector Machine) with the test data set while Decision Tree performed badly with the test data set. The combination of Artificial Neural Network + Support Vector Machine produced good results and the main reason was that the SVM replaced the softmax layer in the Artificial Neural Network, and hence it had the capability of minimizing the generalization error on unseen data which resulted in better prediction results.

4.8 Developed Model for Maize Leaf Diseases

Image classification accuracy is an image classification problem in which images are classified to the class they don't belong to hence leading to decisions that are erroneous and expensive. The enhanced image classification model consisted of two modules; the feature extraction module and the image classification module. The feature extraction module was integrated to work together with the classification module and the features extracted by the feature extraction module were normalized to make them scale-invariant and less susceptible to light which is one of the factors that usually affects image classification accuracy. The classification module was also adjusted by combining two classifiers; Artificial Neural Network and Support Vector Machine and the main reason were for the SVM to replace the softmax layer used for classification in the Artificial Neural Network since the SVM has the hyperplane component which is a line that accurately separates data belonging to different classes and this made Support Vector Machine classify maize leaf disease images accurately. The Support Vector Machine also has the capability of minimizing the generalization error on unseen data which resulted in better prediction results. The conceptual representation of the enhanced model is shown in Figure 4.34 below.

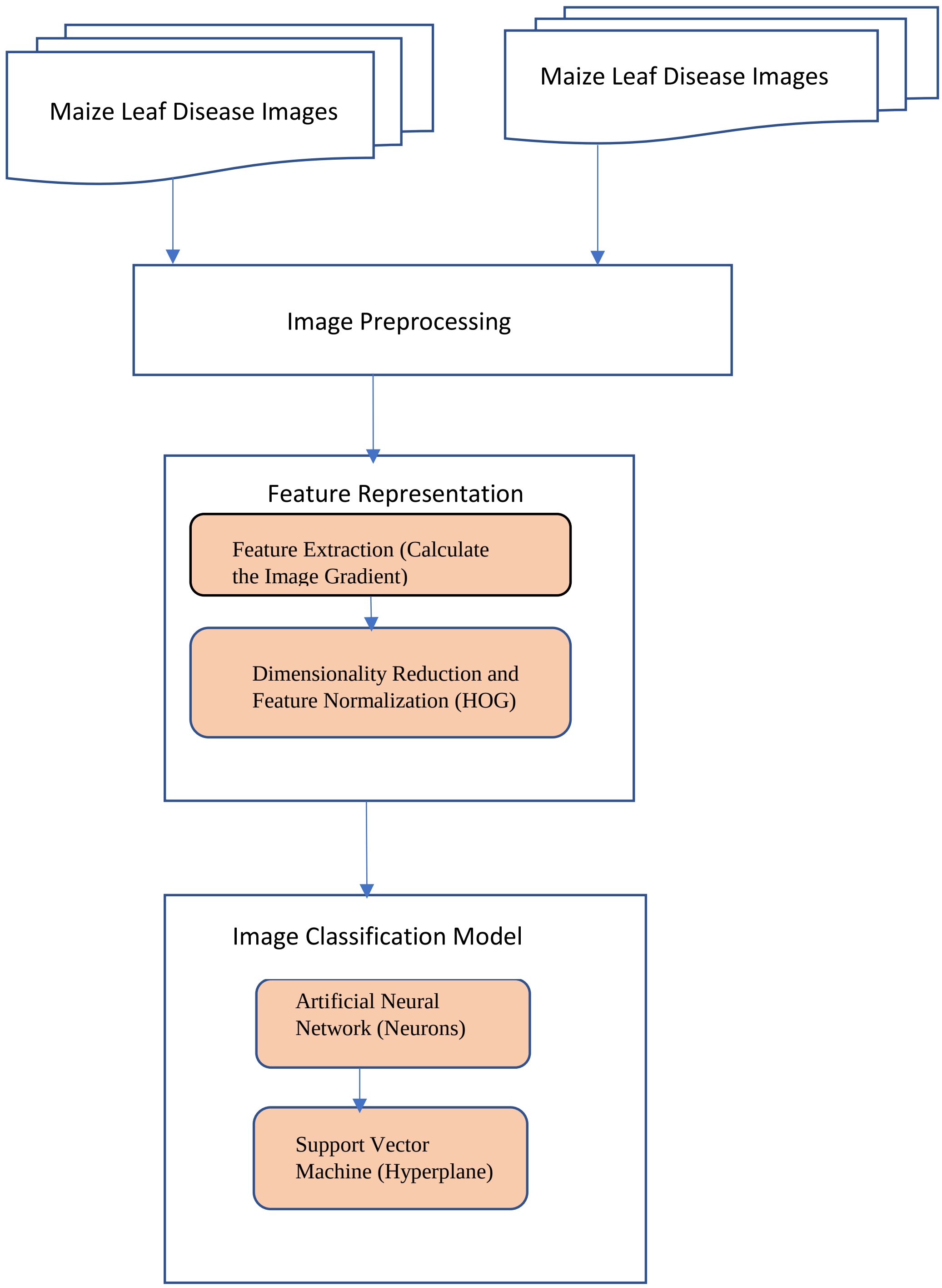

Figure 4.34 Enhanced Image Classification Accuracy Model

Figure 4.34 shows that the feature representation module extracted features by calculating the image gradient which made the features less susceptible to light. The dimensionality reduction was also applied to the images by using the histogram of the oriented gradient algorithm which reduced the vector space. Finally, feature normalization was done by the histogram of the oriented gradient algorithm which made the images scale-invariant hence the images were classified correctly despite the scale variations. The image classification module had a combination of two classifies; the artificial neural network had several neurons that stored more training data and once the training data was captured it was passed to the support vector machine which had the hyperplane component that accurately classified data from different classes. Each component from Figure 4.34 is discussed in detail as shown below.

4.8.1 Maize Leaf Disease Images Input and Preprocessing

The train and test images were of 256 x 256 size and shape which increases the computational complexity of the feature extraction process. The first step done with the feature descriptor was to resize the image into a ratio of 1:2 and most probably the image was resized to 64 × 128, this process was known as image preprocessing. Image preprocessing was important since the images were broken further into 8 by 8 and 16 by 16-pixel windows to be able to generate the features from the images. An image size of pixel ratio of 1:2 made the calculation of feature extraction easier and faster. Finally, the feature vector construction and representation were done from the preprocessed images.

4.8.2 Feature Representation: Calculate the Image Gradient

The change in the x and y direction of every pixel was calculated after the image has been resized to a pixel ratio of 1:2. For example, let us take a small image window and calculate the gradient. Let us work with a matrix pixel of the generated image window taken from the whole image.

121	10	78	96	125
48	152	68	125	111
145	78	85	89	65
154	214	56	200	66
214	87	45	102	45

Figure 4.35 Image Pixels

As shown in Figure 4.35, in the pixel matrix, the value of pixel 85 is highlighted in light orange and that is the one that was used to demonstrate how the gradient of a pixel was calculated. The change in the x-direction of pixel 85 was calculated by subtracting the value of the pixel that is on the left of pixel value 85 from the value of the pixel that is immediately on the right side of pixel value 85. The same thing happens for the change in y-direction for the pixel value 85 which was calculated by subtracting the value immediately at the bottom of pixel value 85 from the value immediately at the top of pixel value 85. The G_x and G_y of 85-pixel value is;

G_x =89-78=11

G_y =68-56=8

This calculation was done for all the pixel values in the matrix and a new matrix was obtained with these new values which helped the researcher calculate the direction and the magnitude.

4.8.3 Orientation and Magnitude

The orientation and magnitude of each pixel value were determined by using the values obtained for the new matrix and this was achieved by using the Pythagoras theorem.

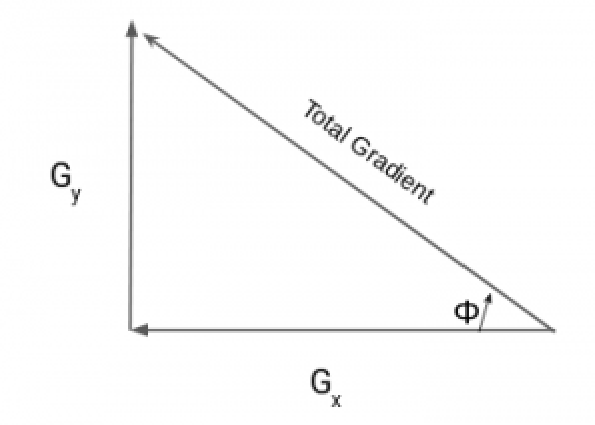

Figure 4.36 Orientation and Magnitude

Figure 4.36 indicates that the height and the base are G_y and G_x respectively and as for the previous example the value for G_y and G_x was 8 and 11 respectively.

Total Gradient Magnitude= $\sqrt{[(G_x)^2 + (G_y)^2]}$

$= \sqrt{[(11)^2 + (8)^2]}$

$= 13.6$

The pixel direction was calculated as;

$\tan(\Theta) = (G_x / G_y)$

$\Theta = \arctan(G_x / G_y)$

$\Theta = \arctan(11 / 8)$

$= 36$

4.8.4 Dimensionality Reduction and Feature Normalization

After calculating the direction and magnitude of each image element value now the magnitude and direction were used to come up with the histogram.

121	10	78	96	125
48	152	68	125	111
145	78	85	89	65
154	214	56	200	66
214	87	45	102	45

Frequency						1									
Angle	1	2	3	4..	35	36	37	38	39..	175	176	177	178	179	180

Figure 4.37 Generating Histogram

As you can see from Figure 4.37 the pixel value 85 has the direction of 36 and in the frequency table the occurrence of 36 is 1 and this was done for each image element value. The values on the y and x-axis were obtained from the frequency distribution table.

4.8.5 Histogram of Gradients in 8 × 8 Image Patch

The histogram of the whole image was obtained from the image segments of size 8 x 8 after the image pixel gradient and direction was calculated. The histograms obtained from the image window of 8 x 8 were used to generate a feature matrix of size 9 x 1. The HOG features were then normalized for the 9 x 1 feature vector to make them scale-invariant.

4.8.6 Normalize Gradients

The variation in light was considered by normalizing the gradient which was done by taking 16 x 16 blocks of images since in the 8 x 8 image patch some parts of the images appeared brighter than others and this procedure helped in reducing light variation. To create 16 × 16 blocks a combination was done for the 8 × 8 cells into one and remember all eight by eight cells had a matrix of 9 × 1 for a histogram so the combination of the four 9 × 1 matrix ended up with a single 36 × 1 matrix. The sum of the square of each value in the matrix was done and the square root was calculated and the results were divided by each of these values. For a given F vector:

F= [x1, x2, x3 …x36]

Determine the root of the sum of squares:

$Y = \sqrt{}\ (x1)^2 + (x2)^2 + (x3)^2 + \ldots\ (x36)^2$

Vector F values are divided by y value

Normalized vector = $\left(\frac{x1}{y}, \frac{x2}{y}, \frac{x3}{y}, \ldots \frac{x36}{y} \right)$

And this resulted in a 36 × 1 matrix normalized vector size.

The normalized feature vector made the images be classified correctly despite the scale variation of the images.

The feature extraction process can be represented using the below algorithms as follows;

Algorithm 1: Histogram of Oriented Gradient

Input: Image

Output: HOG features

Begin algorithm

3. Read Image in the form of pixels
4. Extract the HOG features:

2.1 Divide the image pixel window by 8X8 pixel cell and calculate the gradient components with respect to each pixel (x, y) in vertical and horizontal directions

2.2 Calculate number of blocks in vertical and horizontal directions taking block step size of 8 pixels.

2.3 Histogram of 9 gradient directions is then calculated for each cell and extract feature vectors i.e., HOG features.

2.4 Save all the features in a matrix.

End algorithm

4.8.7 Image Classification Model: Artificial Neural Network

The neurons in the artificial neural network are one of the important components in the model since it acts as the component for storing information. During training the information was distributed over various nodes on the network, this helped more information to be stored on the network which assisted in the learning process of the model. The fault of one neuron cannot affect the model since it has several neurons distributed over the network which will assist during the failure of one node thus the loss of the data cannot affect its working.

4.8.8 Image Classification Model: Support Vector Machine

The SVM had the capability of minimizing the generalization error on unseen data which resulted in better prediction results. The SVM generalizes well the training dataset compared to ANN since it scaled relatively well to high dimensional data. The optimal plane is also known as the hyperplane and is a line that separates data belonging to different classes which made SVM classify maize leaf disease images accurately.

The hyperplane equation is;

w.x + b =0

where b is an offset and w represents the vector which is usually normal to the hyperplane. Separating the data points using the hyperplane, the following steps were followed;

i. Start with a line, and two equidistant parallel lines to it.
ii. Pick a large number i.e 1000(number of repetitions, or epochs)
iii. Pick a number close to 1 i.e 0.99(the expanding factor)
iv. Repeat 1000 times

Pick a random point

If the point is correctly classified: Do nothing

If the data point is incorrectly classified: Move the line towards the point

Separate the lines using the expanding factor

v. Use the lines that separate the data accurately.

The higher the margin around the decision boundary led to an increase in the classification accuracy since the hyperplane and the data points margin were maximized using the following equation.

$$c(x, y, f(x))=\begin{cases} 0, & if\ y * f(x) \geq 1 \\ 1 - y * f(x), & else \end{cases}$$

The above equation is known as the hinge loss function that assisted in maximizing the margin between the hyperplane and the data points. The actual value and the predicted value are of the same sign if the cost function is equal to zero. The loss value is calculated if they are not the same. The cost function is the average of the loss function. To minimize the cost function, an optimization algorithm is used which is known as Gradient Descent. The parameters of the learning model are updated using the optimization algorithm. The gradient was calculated from the loss function by finding the derivatives with respect to weights.

$w_2 = w_1 - L * \frac{dj}{dw}$

$b_2 = b_1 - L * \frac{dj}{db}$

w ---> weight

b ---> bias

L ---> Learning Rate

$\frac{dj}{dw}$ ---> Partial derivative of the cost function with respect to w

$\frac{dj}{db}$ ---> Partial derivative of the cost function with respect to b

Note:

$\frac{dj}{dw}$ is how much your cost function changes when your weight changes

$\frac{dj}{db}$ is how much your cost function changes when you change your bias.

$\frac{dj}{dw}$ and $\frac{dj}{db}$ are known as gradient or derivatives.

The margin maximization and the loss are balanced by adding the regularization parameter to the cost function which looked as follows;

if(y_i . (w.x + b) $\geq$ 1):

$\frac{dj}{dw} = 2\lambda w$

$\frac{dj}{db} = 0$

else(y_i . (w.x + b) <1):

$\frac{dj}{dw} = 2\lambda w - y_i.x_i$

$\frac{dj}{db} = y_i$

The model accurately predicts the class of our data points if there is no misclassification at all. In this case, the regularization parameter was used to update the gradient.

$w = w - \alpha \cdot (2\lambda w)$

The gradient update was calculated by including the regularization parameter along with the loss function when the proposed model misclassifies the images.

$w = w + \alpha \cdot (y_i \cdot x_i - 2\lambda w)$

The Artificial Neural Network also contains a multilayer component with several neurons. During training, the information was distributed to all neurons which made the network learn faster and store more information which assisted for image classification purposes. This feature assisted SVM to correlate with the ANN whenever the training dataset increased since ANN had several neurons which accommodated more training data and hence enhanced the image classification accuracy results. Combining SVM and ANN made them work together and the advantages of SVM and ANN as highlighted above when brought together enabled the model to produce better image classification accuracy.

The model was developed to come up with a stronger learner that has both low bias, variance, and better predictive results compared to a single learner hence enhancing the image classification accuracy. The research also incorporated the best qualities of ANN and SVM hence coming up with a better model which resulted in an enhanced image classification accuracy. Combining ANN and SVM reduced variance by fitting one component of each model at a time and an increase in the capacity of models reduced biases. The combined strengths of the two models offset individual model variances and biases and this provided a composite prediction where the final accuracy was better than the accuracy of individual learners. With the combination of Support Vector Machine and Artificial Neural Network, the SVM replaced the softmax layer in the Artificial Neural Network, and hence the SVM had the capability of minimizing the generalization error on unseen data which resulted in better prediction results.

4.9 Summary

The research aimed to come up with an enhanced model and through the experimental results, it clearly showed that the model developed classifies maize leaf disease with the highest accuracy as compared to single models. The distinctive features were extracted from the images through the feature extraction methods as shown in Figures 4.5 and Figure 4.6. The image classification models used the distinctive feature as the input values. This means once the features were extracted they were passed to the classifiers which were trained on the extracted key points. The performance of three feature extraction methods with the machine learning algorithms is shown in table 4.2. KAZE, ORB, and HOG methods performed well with Artificial Neural Network as seen in table 4.2. The HOG method performed well when the average accuracy was calculated for feature extraction methods when used with different machine learning algorithms. The HOG method reduced the variation in light of the image by finding the gradient magnitude and direction of every pixel. This process usually makes the image less susceptible to light hence making the method detect distinctive features despite the variation in image light. The HOG features once they are generated they were normalized and this made them scale-invariant hence increasing the scale of the images won't affect the identification of maize leaf disease images. Image processing is also one of the processes whereby the images were divided into smaller equal parts hence extracting the features became easier and faster which in turn reduced the computational complexity of the feature extraction process. The above-listed characteristics of the HOG method are the ones that made it produce accurate results compared to KAZE and ORB methods.

The researcher throughout the entire experimental work decided to use the HOG feature extraction method due to its performance compared to other methods. After choosing the HOG the dataset was increased from 200 to 1600 images for each maize disease category hence subjecting the image classification models to a more increased training dataset which enabled the models to make predictions with minimal errors. Figure 4.20, 4.21, 4.22, 4.23, 4.24, 4.25, 4.26, Fig 4.27, 4.28, 4.29, 4.30, 4.31, 4.32, and 4.33 shows the classification reports, and these reports were used to tell whether the image classification models were making good or bad predictions based on the testing dataset they were subjected to. The test dataset of 400 images from each category of maize diseases was used for validating each image classification model and a classification report was obtained after the predictions. And from Figure 4.37 the enhanced model when it came to classifying the maize leaf disease images, emerged the best compared to others. The combination of Artificial Neural Network + Support Vector Machine produced good results and the main reason was that the SVM replaced the softmax layer in the Artificial Neural Network, and hence it had the capability of minimizing the generalization error on unseen data which resulted in better prediction results. The SVM generalizes well the training dataset compared to ANN since it scales relatively well to high dimensional data. The optimal plane is also known as the hyperplane and is a line that separates data belonging to different classes which made SVM classify maize leaf disease images accurately. The developed model also reduced the overfitting problem by averaging the results from the integration of the Support Vector Machine and Artificial Neural Network. The major disadvantage of it was that it took a lot of time to make predictions since it averaged the results from the hybrid of the Support Vector Machine and Artificial Neural Network to give better results hence consuming a lot of time. The new conclusion also needs to be made by exploring further investigation, which will be as a result of increasing the dataset, specifically the training and test dataset, and an empirical analysis be done based on the increased dataset. The new dataset also needs to be used with a different type of plant leaf disease to be able to verify if the newly developed model can still classify the images accurately.

CHAPTER FIVE

CONCLUSION, RECOMMENDATIONS AND FUTURE WORK

5.1 Conclusion

In summary, image classification accuracy is the total number of images predicted correctly out of the total images in the test dataset in the field of computer vision. Classifying the images accurately is still a challenge due to single image classification models being biased and having high variance. The research has presented a good feature extraction method and an enhanced model which is used to classify maize leaf disease images with high accuracy compared to existing models.

This research uses a dimensionality reduction known as histogram of oriented gradient for extracting distinctive features from maize leaf images and leaving out part of the information that is irrelevant and hence passing the extracted features for classification to the image classification model which is a combination of Support Vector Machine + Artificial Neural Network. The method therefore reduces the images from high to low dimensional space but still maintains the intrinsic dimension of the images. The research found that machine learning algorithms can perform better with the Histogram of Oriented Gradient feature descriptor method after a comparative study was done with other feature extraction methods. KAZE, ORB, and HOG methods performed well with Artificial Neural Network as seen in table 4.2. The HOG method performed well when the average accuracy was calculated for feature extraction methods when used with different machine learning algorithms. The HOG method reduced the variation in light of the images by finding the gradient magnitude and direction of every pixel. This process made the images less susceptible to light hence making the method detect distinctive features despite the variation in image light. The

HOG features once they are generated they were normalized and this made them scale-invariant hence increasing the scale of the images did not affect the identification of maize leaf disease images. Image processing was also one of the processes whereby the images were divided into smaller equal parts hence extracting the features became easier and faster which in turn reduced the computational complexity of the feature extraction process. The above-listed characteristics of the HOG method are the ones that made it produce accurate results compared to KAZE and ORB methods. The researcher throughout the entire experimental work decided to use the HOG feature extraction method due to its performance compared to other methods.

The classification reports were generated and these reports were used to tell whether the models were making good or bad predictions based on the test dataset they were subjected to. Each model was subjected to a test data set of 400 images from each category of maize diseases and a classification report was obtained after the predictions. And from the results, the proposed model of Histogram of Oriented Gradient + Support Vector Machine + Artificial Neural Networks classified maize leaf diseases with the highest accuracy.

The combination of Support Vector Machine + Artificial Neural Network performed best after comparative analysis was done with the other image classification models. The accuracy score of 0.95 was produced by the developed model. The main reason why it produced good results is that with the combination of Histogram of Oriented Gradient + Support Vector Machine + Artificial Neural Network, the features were extracted from maize leaf disease images using the Histogram of Oriented Gradient, and the combination of Artificial Neural Network and Support Vector Machine used the features as their input value during training, the SVM replaced the softmax layer in the Artificial Neural Network, and hence the SVM had the capability of minimizing the generalization error on unseen data which resulted in better prediction results. From the classification report, the enhanced model also was seen to be the best, and in conclusion, the developed model classified maize leaf disease images with the highest accuracy when the HOG method acted as the feature descriptor and the combination of Support Vector Machine and Artificial Neural Network as the classifier which made the researcher propose the model to be used today and in the future when it comes to classifying maize leaf disease images.

5.2 Recommendations

The research recommends that other existing image classification model needs to be analyzed specifically convolution neural network and a hybrid of other image classification models. A better model that uses deep learning and transfer learning need to be developed and compared with the developed model to see if accuracy improves.

The ensembling needs to be done with deep learning models to see if they generalize well the data compared to machine learning models. Ensembling is one of the approaches that reduces bias, variance and the overfitting problem which in turn improves the image classification accuracy.

The study also recommends the penalization of the wrong prediction by using the logarithmic loss metric needs to be measured. The probability of the image classification model randomly classifying an image to the wrong class rather than randomly classifying it to the right class can also needs to be measured by the Area Under the Curve (AUC) metric which is one of the metrics that the model can be subjected to.

5.3 Future Work

The developed model which was a combination of Support Vector Machine and Artificial Neural Network produced an accuracy of 0.95, unfortunately, this model is not one hundred percent accurate which leads to instances of the wrong classification hence erroneous decisions that are expensive. This is due to generalization error, biases, and variance which are still associated with the model hence the research recommends in the future to use the latest technology like deep learning specifically convolution neural network to see if the accuracy will be better than the current model. The combined two model needs to be subjected to further hyperparameter tuning by using advanced methods such as Conjugate Gradient to be able to compare the current optimal parameters with the new parameters obtained since optimal parameters when used along with the models' results to accurate predictions. During the feature extraction process before the features are fed to image classification models, feature selection needs to be done by using methods such as recursive feature elimination which usually ensures redundant features are eliminated thus making the machine learning algorithm perform faster during training and validation. The researcher also needs to explore other validation metrics that can exhaustively verify the prediction results of the machine learning algorithms. The new dataset also needs to be used with a different type of plant leaf disease to be able to verify if the newly developed model can still classify the images accurately. The researcher in the future can look at the penalization of the wrong prediction by using the logarithmic loss metric. The probability of the image classification model randomly classifying an image to the wrong class rather than randomly classifying it to the right class can also be measured by the Area Under the Curve (AUC) metric which is one of the metrics that the model can be subjected to in the future research. The noise can still be introduced in both the training and testing dataset by using noise functions in python such as the Gauss and Poisson function that distorts the images and makes them difficult for the classifier to recognize them and this will assist in comparing the performance of the classifiers both in normal and noisy conditions. The new conclusion also needs to be made by exploring further investigation, which will be as a result of increasing the dataset, specifically the training and test dataset, and an empirical analysis be done based on the increased dataset.

REFERENCES

[1] J. V. Rissati, P. C. Molina and C. S. Anjos, "Hyperspectral Image Classification Using Random Forest and Deep Learning Algorithms," *2020 IEEE Latin American GRSS & ISPRS Remote Sensing Conference (LAGIRS)*, 2020, pp. 132-132, doi: 10.1109/LAGIRS48042.2020.9165588.

[2] S. K. Sen, M. Foskey, J. S. Marron and M. A. Styner, "Support vector machine for data on manifolds: An application to image analysis," 2008 5th IEEE International Symposium on Biomedical Imaging: From Nano to Macro, 2008, pp. 1195-1198, doi: 10.1109/ISBI.2008.4541216.

[3] Gan, G., & Cheng, J. (2014, December). Pedestrian detection based on HOG-LBP feature. In *2014 Seventh International Conference on Computational Intelligence and Security* (pp. 1184-1187). IEEE.

[4] Z Zhiyong, H Xiaoyang, et al. Image recognition of maize leaf disease based on GA-SVM. *Chemical Engineering Transactions 2017*; 46:199-204.

[5] Moghimi.A, T. Celik, A. Mohammadzadeh and H. Kusetogullari, "Comparison of Keypoint Detectors and Descriptors for Relative Radiometric Normalization of Bitemporal Remote Sensing Images," *in IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 14, pp. 4063-4073, 2021, doi: 10.1109/JSTARS.2021.3069919.

[6] Zheng-Jian Ding, Yang Zhang, A-Qing Yang and Dai-Li, "Image matching of Gaussian blurred image based on SIFT algorithm," 2012 International Conference on Wavelet Active Media Technology and Information Processing (ICWAMTIP), 2012, pp. 121-124, doi: 10.1109/ICWAMTIP.2012.6413454.

[7] S. Manthira Moorthi, I. Misra, R. Kaur, N. P. Darji and R. Ramakrishnan, "Kernel based learning approach for satellite image classification using support vector machine," 2011 IEEE Recent Advances in Intelligent Computational Systems, 2011, pp. 107-110, doi: 10.1109/RAICS.2011.6069282.

[8] Upadhyay, A. Upadhyay and S. Maurya, "Regression and Artificial Neural Network based Improved Classification of LISS-III Satellite Image," 2017 International Conference on Current Trends in Computer, Electrical, Electronics and Communication (CTCEEC), 2017, pp. 917-921, doi: 10.1109/CTCEEC.2017.8455083.

[9] Sabrol. H and K. Satish, "Tomato plant disease classification in digital images using classification tree," 2016 International Conference on Communication and Signal Processing (ICCSP), 2016, pp. 1242-1246, doi: 10.1109/ICCSP.2016.7754351.

[10] B. Hou, X. Luo, S. Wang, L. Jiao and X. Zhang, "Polarimetric SAR images classification using deep belief networks with learning features," 2015 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), 2015, pp. 2366-2369, doi: 10.1109/IGARSS.2015.7326284.

[11] A. Hemalatha and J. Vijayakumar, "Automatic Tomato Leaf Diseases Classification and Recognition using Transfer Learning Model with Image Processing Techniques," 2021 Smart Technologies, Communication and Robotics (STCR), 2021, pp. 1-5, doi: 10.1109/STCR51658.2021.9588993.

[12] Bosch, A., Zisserman, A., & Munoz, X. (2017, October). Image classification using random forests and ferns. In *2017 IEEE 11th international conference on computer vision* (pp. 1-8). IEEE.

[13] C. O. Dumitru, G. Schwarz, S. Cui and M. Datcu, "Improved image classification by proper patch size selection: TerraSAR-X vs. Sentinel-1A," 2016 International Conference on Systems, Signals and Image Processing (IWSSIP), 2016, pp. 1-4, doi: 10.1109/IWSSIP.2016.7502739.

[14] A. Sutrisno, Indrabayu, I. S. Areni and A. Bustamin, "Reducing Area Recognition for Vehicle Model Classification using Car's Front Side," 2021 4th International Conference on Information and Communications Technology (ICOIACT), 2021, pp. 290-294, doi: 10.1109/ICOIACT53268.2021.9564018.

[15] C. S. Janadri, B. G. Sheeparamatti and V. Kagawade, "Multiclass classification of kirlian images using svm technique," 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI), 2017, pp. 2246-2250, doi: 10.1109/ICACCI.2017.8126180.

[16] Bertalya, Prihandoko, D. Kerami and T. M. Kusuma, "Classification of X-Ray Images Using Grid Approach," 2008 IEEE International Conference on Signal Image Technology and Internet Based Systems, 2008, pp. 314-319, doi: 10.1109/SITIS.2008.76.

[17] D. A. Padilla, R. A. I. Pajes and J. T. De Guzman, "Detection of Corn Leaf Diseases Using Convolutional Neural Network With OpenMP Implementation," 2020 IEEE 12th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment, and Management (HNICEM), 2020, pp. 1-6, doi: 10.1109/HNICEM51456.2020.9400004.

[18] Chen, P. Y., Huang, C. C., Lien, C. Y., & Tsai, Y. H. (2018). An efficient hardware implementation of HOG feature extraction for human detection. *IEEE Transactions on Intelligent Transportation Systems*, *15*(2), 656-662.

[19] D. Baicu, M. Crăciunescu, Ş. Mocanu and M. Cîrciumaru, "Real time Image Recognition Based on low cost Processing Platform," 2019 14th International Conference on Advanced Technologies, Systems and Services in Telecommunications (TELSIKS), 2019, pp. 241-246, doi: 10.1109/TELSIKS46999.2019.9002364.

[20] B. S. Kusumo, A. Heryana, O. Mahendra and H. F. Pardede, "Machine Learning-based for Automatic Detection of Corn-Plant Diseases Using Image Processing," 2018 International Conference on Computer, Control, Informatics and its Applications (IC3INA), 2018, pp. 93-97, doi: 10.1109/IC3INA.2018.8629507.

[21] D. J. M. Bonifacio, A. M. I. E. Pascual, M. V. C. Caya and J. C. Fausto, "Determination of Common Maize (Zea mays) Disease Detection using Gray-Level Segmentation and Edge-Detection Technique," 2020 IEEE 12th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment, and Management (HNICEM), 2020, pp. 1-6, doi: 10.1109/HNICEM51456.2020.9399998.

[22] Chen, P. Y., Huang, C. C., Lien, C. Y., & Tsai, Y. H. (2018). An efficient hardware implementation of HOG feature extraction for human detection. *IEEE Transactions on Intelligent Transportation Systems*, *15*(2), 656-662.

[23] Dawei Qi, Hongbo Mu, Mingming Zhang and Lei Yu, "Detection of wood defects from X-ray image by ANN," 2008 IEEE International Conference on Automation and Logistics, 2008, pp. 23-28, doi: 10.1109/ICAL.2008.4636113.

[24] E. Akanksha, N. Sharma and K. Gulati, "OPNN: Optimized Probabilistic Neural Network based Automatic Detection of Maize Plant Disease Detection," 2021 6th International Conference on Inventive Computation Technologies (ICICT), 2021, pp. 1322-1328, doi: 10.1109/ICICT50816.2021.9358763.

[25] Cheng Xing and Jingjing Huang, "An improved mosaic method based on SIFT algorithm for UAV sequence images," 2010 International Conference On Computer Design and Applications, 2010, pp. V1-414-V1-417, doi: 10.1109/ICCDA.2010.5540824.

[26] D. Patidar, N. Jain and A. Parikh, "Performance analysis of artificial neural network and K Nearest neighbors image classification techniques with wavelet features," Proceedings of IEEE International Conference on Computer Communication and Systems ICCCS14, 2014, pp. 191-194, doi: 10.1109/ICCCS.2014.7068192.

[27] F. Özyurt, M. Toğaçar, E. Avcı and D. Avcı, "Classification of Breast Cancer Images by Using of Convolutional Attribute of ANN," 2018 3rd International Conference on Computer Science and Engineering (UBMK), 2018, pp. 420-423, doi: 10.1109/UBMK.2018.8566624.

[28] N. A. Mahmon and N. Ya'acob, "A review on classification of satellite image using Artificial Neural Network (ANN)," 2014 IEEE 5th Control and System Graduate Research Colloquium, 2014, pp. 153-157, doi: 10.1109/ICSGRC.2014.6908713.

[29] Dietterich, T. G. (2017). Fundamental experimental research in machine learning. A section of the document Basic Topics in Experimental Computer Science edited by John McCarthy.

[30] K. Aparna and P. Supriya, "Precision Agriculture in Maize Fields," 2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS), 2018, pp. 1407-1410, doi: 10.1109/ICCONS.2018.8662936.

[31] Foody, G. M., & Mathur, A. (2016). The use of small training sets containing mixed pixels for accurate hard image classification: Training on mixed spectral responses for classification by a SVM. *Remote Sensing of Environment*, *103*(2), 179-189.

[32] H. Riri, A. Elmoutaouakkil, A. Beni-Hssane and F. Bourezgui, "Classification and Recognition of Dental Images Using a Decisional Tree," 2016 13th International Conference on Computer Graphics, Imaging and Visualization (CGiV), 2016, pp. 390-393, doi: 10.1109/CGiV.2016.82.

[33] M. V. Overbeek, Y. R. Kaesmetan and F. A. T. Tobing, "Identification of Maize Leaf Diseases Cause by Fungus with Digital Image Processing (Case Study: Bismarak Village Kupang District - East Nusa Tenggara)," 2019 5th International Conference on New Media Studies (CONMEDIA), 2019, pp. 125-128, doi: 10.1109/CONMEDIA46929.2019.8981843.

[34] H. Liu et al., "Unsupervised classification of polarimetric SAR images integrating color features," 2014 IEEE Geoscience and Remote Sensing Symposium, 2014, pp. 2762-2765, doi: 10.1109/IGARSS.2014.6947048.

[35] P. Bertalya, "Grid approach for X-ray image classification," International Conference on Instrumentation, Communication, Information Technology, and Biomedical Engineering 2009, 2009, pp. 1-5, doi: 10.1109/ICICI-BME.2009.5417256.

[36] Millard, K., & Richardson, M. (2015). On the importance of training data sample selection in random forest image classification: A case study in peatland ecosystem mapping. *Remote sensing*, *7*(7), 8489-8515.

[37] Panigrahi, K. P., Das, H., Sahoo, A. K., & Moharana, S. C. (2020). Maize Leaf Disease Detection and Classification Using Machine Learning Algorithms. In *Progress in Computing, Analytics and Networking* (pp. 659-669). Springer, Singapore.

[38] H. Sabrol and K. Satish, "Tomato plant disease classification in digital images using classification tree," 2016 International Conference on Communication and Signal Processing (ICCSP), 2016, pp. 1242-1246, doi: 10.1109/ICCSP.2016.7754351.

[39] Islam, M. A., Yousuf, M. S. I., & Billah, M. M. (2019). Automatic plant detection using HOG and LBP features with SVM. *International Journal of Computer (IJC)*, *33*(1), 26-38.

[40] K Song, Z liu, et al. A Research of maize disease image recognition of Corn Based on BP Networks. Measuring Technology and Mechatronics Automation China 2011: 246-249.

[41] J. Guo and X. Wang, "Image Classification Based on SURF and KNN," 2019 IEEE/ACIS 18th International Conference on Computer and Information Science (ICIS), 2019, pp. 356-359, doi: 10.1109/ICIS46139.2019.8940198.

[42] H. Riri, A. Elmoutaouakkil, A. Beni-Hssane and F. Bourezgui, "Classification and Recognition of Dental Images Using a Decisional Tree," 2016 13th International Conference on Computer Graphics, Imaging and Visualization (CGiV), 2016, pp. 390-393, doi: 10.1109/CGiV.2016.82.

[43] J. -D. Lee, Y. -S. Chen and J. -C. Chien, "Classification and tracking of large vehicles for night driving," 2016 IEEE 5th Global Conference on Consumer Electronics, 2016, pp. 1-2, doi: 10.1109/GCCE.2016.7800320.

[44] K. Mishiba and T. Yoshitome, "Image resizing with SIFT feature preservation," 2013 IEEE International Conference on Image Processing, 2013, pp. 991-995, doi: 10.1109/ICIP.2013.6738205.

[45] K. T. Islam, S. Wijewickrema, M. Pervez and S. O'Leary, "Road Trail Classification using Color Images for Autonomous Vehicle Navigation," 2018 Digital Image Computing: Techniques and Applications (DICTA), 2018, pp. 1-5, doi: 10.1109/DICTA.2018.8615834.

[46] N. Jayanthi, T. Sharma, V. Sharma, S. Tyagi and S. Indu, "Classification of ancient inscription images on the basis of material of the inscriptions," 2021 3rd International Conference on Signal Processing and Communication (ICPSC), 2021, pp. 422-427, doi: 10.1109/ICSPC51351.2021.9451641.

[47] Kobayashi, T. (2013). BFO meets HOG: feature extraction based on histograms of oriented pdf gradients for image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 747-754).

[48] Liu, Z., Du, Z., Peng, Y., Tong, M., Liu, X., & Chen, W. (2020, June). Study on Corn Disease Identification Based on PCA and SVM. In 2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC) (Vol. 1, pp. 661-664). IEEE.

[49] J. Fan, Y. Wu, F. Wang, Q. Zhang, G. Liao and M. Li, "SAR Image Registration Using Phase Congruency and Nonlinear Diffusion-Based SIFT," in IEEE Geoscience and Remote Sensing Letters, vol. 12, no. 3, pp. 562-566, March 2015, doi: 10.1109/LGRS.2014.2351396.

[50] M. D. Lakshmi, M. V. Raj and S. S. Murugan, "Feature Matching and Assessment of Similarity rate on geometrically Distorted Side Scan Sonar Images," 2019 TEQIP III Sponsored International Conference on Microwave Integrated [50] [51] Circuits, Photonics and Wireless Networks (IMICPW), 2019, pp. 208-212, doi: 10.1109/IMICPW.2019.8933268.

[51] Ying Li and Bo Cheng, "An improved k-nearest neighbor algorithm and its application to high resolution remote sensing image classification," 2009 17th International Conference on Geoinformatics, 2009, pp. 1-4, doi: 10.1109/GEOINFORMATICS.2009.5293389.

[52] M. Lv, G. Zhou, M. He, A. Chen, W. Zhang and Y. Hu, "Maize Leaf Disease Identification Based on Feature Enhancement and DMS-Robust Alexnet," in IEEE Access, vol. 8, pp. 57952-57966, 2020, doi: 10.1109/ACCESS.2020.2982443.

[53] S. Zhong, C. -I. Chang and Y. Zhang, "Iterative Support Vector Machine for Hyperspectral Image Classification," 2018 25th IEEE International Conference on Image Processing (ICIP), 2018, pp. 3309-3312, doi: 10.1109/ICIP.2018.8451145.

[54] Mahima and N. B. Padmavathi, "Comparative study of kernel SVM and ANN classifiers for brain neoplasm classification," 2017 International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT), 2017, pp. 469-473, doi: 10.1109/ICICICT1.2017.8342608.

[55] J. Sun, Y. Yang, X. He and X. Wu, "Northern Maize Leaf Blight Detection Under Complex Field Environment Based on Deep Learning," in IEEE Access, vol. 8, pp. 33679-33688, 2020, doi: 10.1109/ACCESS.2020.2973658.

[56] O. A. Máximo and D. Fernandez, "Classification of a reference image using auxiliary images," 2009 IEEE International Geoscience and Remote Sensing Symposium, 2009, pp. III-697-III-700, doi: 10.1109/IGARSS.2009.5417858.

[57] J. Blahuta, P. Cermak and J. Skacel, "The Concept of an Artificial Neural Network for the Classification of Atheromous Plaques from Digitized Segmented [59] Histological Images," 2018 14th International Computer Engineering Conference (ICENCO), 2018, pp. 22-25, doi: 10.1109/ICENCO.2018.8636116.

[58] P. Tamrakar and H. Amhia, "ANN and AdaBoost Based Effective Face Recognition and Improvement of Classification Training," 2021 International Conference on System, Computation, Automation and Networking (ICSCAN), 2021, pp. 1-5, doi: 10.1109/ICSCAN53069.2021.9526499.

[59] O. Mahendra, H. F. Pardede, R. Sustika and R. B. Suryo Kusumo, "Comparison of Features for Strawberry Grading Classification with Novel Dataset," 2018 International Conference on Computer, Control, Informatics and its Applications (IC3INA), 2018, pp. 7-12, doi: 10.1109/IC3INA.2018.8629534.

[60] P. Ahmadi, "A simple and efficient method for segmentation and classification of aerial images," 2013 6th International Congress on Image and Signal Processing (CISP), 2013, pp. 566-570, doi: 10.1109/CISP.2013.6744061.

[61] M. J. A. Jansen, H. J. Kuijf and J. P. W. Pluim, "Automatic classification of focal liver lesions based on clinical DCE-MR and T2-weighted images: A feasibility study," 2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018), 2018, pp. 245-248, doi: 10.1109/ISBI.2018.8363565.

[62] P. Ahmadi, "A simple and efficient method for segmentation and classification of aerial images," 2013 6th International Congress on Image and Signal Processing (CISP), 2013, pp. 566-570, doi: 10.1109/CISP.2013.6744061.

[63] K. R. Aravind, P. Raja, K. V. Mukesh, R. Aniirudh, R. Ashiwin and C. Szczepanski, "Disease classification in maize crop using bag of features and multiclass

support vector machine," 2018 2nd International Conference on Inventive Systems and Control (ICISC), 2018, pp. 1191-1196, doi: 10.1109/ICISC.2018.8398993.

[64] PD Jagadeesh, Y Rajesh, et al. Classification of Fungal Disease Symptoms affected on Cereals using Color Texture Features. International Journal of Signal Processing, Image Processing and Pattern Recognition 2018; 6: 321-330.

[65] S. Luo, K. Sarabandi, L. Tong and L. Pierce, "A SAR Image Classification Algorithm Based on Multi-Feature Polarimetric Parameters Using FOA and LS-SVM," in IEEE Access, vol. 7, pp. 175259-175276, 2019, doi: 10.1109/ACCESS.2019.2957547.

[66] Pujari, D., Yakkundimath, R., & Byadgi, A. S. (2016). SVM and ANN Based Classification of Plant Diseases Using Feature Reduction Technique. International Journal of Interactive Multimedia and Artificial Intelligence 2018; 3: 6-14.

[67] R. Singh and R. Gupta, "Improvement of Classification Accuracy Using Image Fusion Techniques," 2016 International Conference on Computational Intelligence and Applications (ICCIA), 2016, pp. 36-40, doi: 10.1109/ICCIA.2016.21.

[68] Y. Yamamoto, S. Tsuruta, S. Kobashi, Y. Sakurai and R. Knauf, "An Efficient Classification Method for Knee MR Image Segmentation," 2016 12th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS), 2016, pp. 36-41, doi: 10.1109/SITIS.2016.15.

[69] P. Ahmadi, "A simple and efficient method for segmentation and classification of aerial images," 2013 6th International Congress on Image and Signal Processing (CISP), 2013, pp. 566-570, doi: 10.1109/CISP.2013.6744061.

[70] R. Singh and R. Gupta, "Improvement of Classification Accuracy Using Image Fusion Techniques," 2016 International Conference on Computational Intelligence and Applications (ICCIA), 2016, pp. 36-40, doi: 10.1109/ICCIA.2016.21.

[71] S. A. Ahmed, S. Dey and K. K. Sarma, "Image texture classification using Artificial Neural Network (ANN)," 2011 2nd National Conference on Emerging Trends and Applications in Computer Science, 2011, pp. 1-4, doi: 10.1109/NCETACS.2011.5751383.

[72] Ramteke, R. J., & Monali, Y. K. (2012). Automatic medical image classification and abnormality detection using k-nearest neighbour. *International Journal of Advanced Computer Research*, *2*(4), 190-196.

[73] S. Kai, L. Zhikun, S. Hang and G. Chunhong, "A Research of Maize Disease Image Recognition of Corn Based on BP Networks," 2011 Third International Conference on Measuring Technology and Mechatronics Automation, 2011, pp. 246-249, doi: 10.1109/ICMTMA.2011.66.

[74] X. Zhang, K. Jiang, Y. Zheng, J. An, Y. Hu and L. Jiao, "Spatially constrained Bag-of-Visual-Words for hyperspectral image classification," 2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), 2016, pp. 501-504, doi: 10.1109/IGARSS.2016.7729124.

[75] S. Manthira Moorthi, I. Misra, R. Kaur, N. P. Darji and R. Ramakrishnan, "Kernel based learning approach for satellite image classification using support vector machine," 2011 IEEE Recent Advances in Intelligent Computational Systems, 2011, pp. 107-110, doi: 10.1109/RAICS.2011.6069282.

[76] KIM¹, J. I. N. H. O., Kim, B. S., & Savarese, S. (2012). Comparing image classification methods: K-nearest-neighbor and support-vector-machines. In *Proceedings of the 6th WSEAS international conference on Computer Engineering and Applications, and Proceedings of the 2012 American conference on Applied Mathematics* (Vol. 1001, pp. 48109-2122).

[77] Tang, Y. (2013). Deep learning using linear support vector machines. arXiv preprint arXiv:1306.0239.

[78] S. Yadav, R. Rathod, S. R. Pawar, V. S. Pawar and S. More, "Application of Deep Convulational Neural Network in Medical Image Classification," 2021 International Conference on Emerging Smart Computing and Informatics (ESCI), 2021, pp. 120-129, doi: 10.1109/ESCI50559.2021.9396854.

[79] Y. Yu, E. Favour and P. Mazumder, "Convolutional Neural Network Design for Breast Cancer Medical Image Classification," 2020 IEEE 20th International Conference on Communication Technology (ICCT), 2020, pp. 1325-1332, doi: 10.1109/ICCT50939.2020.9295909.

[80] S. Zhong, C. -I. Chang and Y. Zhang, "Iterative Support Vector Machine for Hyperspectral Image Classification," 2018 25th IEEE International Conference on Image Processing (ICIP), 2018, pp. 3309-3312, doi: 10.1109/ICIP.2018.8451145.

[81] T. Lu, "Improving Image Matching Accuracy with Keypoint Features and Feature Fusion," 2021 IEEE 3rd International Conference on Frontiers Technology of Information and Computer (ICFTIC), 2021, pp. 513-520, doi: 10.1109/ICFTIC54370.2021.9647145.

[82] S. Anuja Prasad and L. Mary, "A Comparative Study of Different Features for Vehicle Classification," 2019 International Conference on Computational Intelligence in Data Science (ICCIDS), 2019, pp. 1-5, doi: 10.1109/ICCIDS.2019.8862136.

[83] X. Zhang, Q. Song, Y. Zheng, B. Hou and S. Gou, "Classification of imbalanced hyperspectral imagery data using support vector sampling," 2014 IEEE Geoscience and Remote Sensing Symposium, 2014, pp. 2870-2873, doi: 10.1109/IGARSS.2014.6947075.

[84] Y. Ou, Z. Cai, J. Lu, J. Dong and Y. Ling, "Evaluation of Image Feature Detection and Matching Algorithms," 2020 5th International Conference on Computer and Communication Systems (ICCCS), 2020, pp. 220-224, doi: 10.1109/ICCCS49078.2020.9118480.

[85] T. Treebupachatsakul and S. Poomrittigul, "Bacteria Classification using Image Processing and Deep learning," 2019 34th International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC), 2019, pp. 1-3, doi: 10.1109/ITC-CSCC.2019.8793320.

[86] Y. -S. Chen, J. -C. Chien and J. -D. Lee, "KAZE-BOF-based large vehicles detection at night," 2016 International Conference On Communication Problem-Solving (ICCP), 2016, pp. 1-2, doi: 10.1109/ICCPS.2016.7751099.

[87] P. Raikar and S. M. Joshi, "Efficiency Comparison of Supervised and Unsupervised Classifier on Content Based Classification using Shape, Color, Texture," 2020 International Conference for Emerging Technology (INCET), 2020, pp. 1-7, doi: 10.1109/INCET49848.2020.9154016.

[88] Y. Yamamoto, S. Tsuruta, S. Kobashi, Y. Sakurai and R. Knauf, "An Efficient Classification Method for Knee MR Image Segmentation," 2016 12th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS), 2016, pp. 36-41, doi: 10.1109/SITIS.2016.15.

[89] P. Sidike, C. Chen, V. Asari, Y. Xu and W. Li, "Classification of hyperspectral image using multiscale spatial texture features," 2016 8th Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS), 2016, pp. 1-4, doi: 10.1109/WHISPERS.2016.8071767.

[90] B. Demir and S. Ertürk, "Improving SVM classification accuracy using a hierarchical approach for hyperspectral images," 2009 16th IEEE International Conference on Image Processing (ICIP), 2009, pp. 2849-2852, doi: 10.1109/ICIP.2009.5414491.

APPENDICES

Appendix 1. Code for Importing Libraries

These codes import the libraries for computer vision and image classification models that were used during the experiment.

```
#for array manipulations

import numpy as np

#for image processing

import cv2

#for displaying images

import matplotlib.pyplot as plt

#to display images in this notebook, not in a separate window

%matplotlib inline

#to access system resources such as directories

import os

import time

import pandas as pd

import seaborn as sns

import random

from sklearn.preprocessing import StandardScaler

from sklearn.model_selection import train_test_split

from sklearn.preprocessing import LabelEncoder

from sklearn.ensemble import RandomForestClassifier

from sklearn.linear_model import LogisticRegression

from sklearn.svm import SVC, LinearSVC

from sklearn.neighbors import KNeighborsClassifier
```

```
from sklearn.neural_network import MLPClassifier
from sklearn.naive_bayes import BernoulliNB
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score, auc, classification_report, confusion_matrix, f1_score
from sklearn.metrics import precision_score, recall_score
from sklearn.feature_selection import RFECV
from sklearn.model_selection import train_test_split
from sklearn.model_selection import KFold
from sklearn.model_selection import cross_val_score, cross_val_predict
from sklearn.model_selection import GridSearchCV, RandomizedSearchCV
from sklearn.utils import shuffle
from sklearn.ensemble import VotingClassifier
import warnings
warnings.filterwarnings('ignore')
```

Appendix 2. Code for Setting up Directories

```
#Set this to point to the project root; all paths will be relative to this one
project_dir = 'C:/Users/VINCENT/Documents/maize-disease-detection'
def set_up_directories(project_dir=project_dir):
    """Sets up the paths to important directories

    Parameters
    ----------
    project_dir : string; default is the current working directory
        The path to the project root i.e 'VINCENT/maize-disease-detection'

    returns
    -------
    base_dir : string
        The project directory path
    data_folder : string
        The data subfolder path
    maize_data_folder :
        The path to the subdirectory containing the maize images

    example usage
    -------------
    base_dir, data_folder, maize_data_folder = set_up_directories()
    """
```

```
#set our base directory. This should point to the location of the plant-diseases folder

base_dir = project_dir

#set the path to our data folder

data_folder = os.path.join(base_dir, 'data')

#set the path to the maize folder and list the various categories available

maize_data_folder = os.path.join(data_folder, 'maize')


return base_dir, data_folder, maize_data_folder
```

Apendix 3.Code for Loading Images from the Respective Folders

```
def get_images(disease, image_count=200, offset=0):
    """Loads a specified number of images for a given maize disease

    parameters
    ----------
    disease: string
        A string that could be common_rust, healthy, leaf_spot, nothern_leaf_blight
    image_count : int
        Number of images to return
    returns
    -------
    disease_images: list
        A list of images for the selected disease
    offset : int
        Where to begin
    """

    #this list will contain the  images returned
    disease_images = []
    #path to the images
    disease_images_path = os.path.join(maize_data_folder, disease)
    count = 0
    image_paths = os.listdir(disease_images_path)
    for image_path in image_paths[offset:]:
        if count == image_count:
            break
        image_path = os.path.join(disease_images_path, image_path)
        image = cv2.imread(image_path, cv2.IMREAD_COLOR)
        image = cv2.cvtColor(image,cv2.COLOR_BGR2RGB)
```

```
        disease_images.append(image)
        count += 1
    return disease_images
```

Appendix 4. Code for Extracting Train Features from the Images

```
def extract_features_hog(image, feature_size=4096):
    """Extracts hog features for the image


    parameters
    ----------
    image : numpy array
        The image whose features are to be extracted
    feature_size : int
        The number of features to generate
    returns
    -------
        hog_features : numpy array


    raises
    ------
    cv2.error
    """
    hog = cv2.HOGDescriptor()
    try:
        features = hog.compute(image)
        required_features = features[:feature_size].ravel()
    except AttributeError as e:
        raise AttributeError('Unable to generate features for the given image')
    else:
        return required_features


def extract_train_features(algorithm='hog', dataset_size=450):
    """Extracts features for the given number of images in the dataset
```

```
Uses the specified algorithm
Generates specified training samples for each maize disease
parameters
----------
algorithm : string
---------
    The algorithm to use; could be 'kaze', 'orb' or 'hog'
dataset_size : int, optional
    Number of images to load for each category
returns
-------
features : numpy array
    The features used to train the models
labels : numpy array
    The feature labels
"""

features = []
labels = []
disease_names = ['common_rust', 'healthy', 'leaf_spot', 'nothern_leaf_blight']
for disease_name in disease_names:
    images = get_images(disease_name, image_count=dataset_size)
    for image in images:
        try:
            if algorithm == 'kaze':
                image_features = extract_features_kaze(image)
                features.append(image_features)
            elif algorithm == 'orb':
                image_features = extract_features_orb(image)
                features.append(image_features)
```

```
            else:
                image_features = extract_features_hog(image)
                features.append(image_features)
            labels.append(disease_name)
        except AttributeError as e:
            continue
features = np.array(features)
labels = np.array(labels)
features = StandardScaler().fit_transform(features)
labels = LabelEncoder().fit_transform(labels)
features, labels = shuffle(features, labels, random_state=34)

return  features, labels
```

Appendix 5. Code for Extracting Test Features from the Images

```
def extract_test_features(algorithm='hog', dataset_size=100):
    """Extracts features for the given number of images in the dataset

    Uses kaze algorithm
    parameters
    ----------
    algorithm : string
        The algorithm to use; could be 'kaze', 'orb' or 'hog'
    dataset_size : int
        Number of images to load for each category
    returns
    -------
    features : numpy array
        The features used to train the models
    labels : numpy array
        The feature labels
    """

    test_features = []
    test_labels = []
    disease_names = ['common_rust', 'healthy', 'leaf_spot', 'nothern_leaf_blight']
    for disease_name in disease_names:
        images = get_images(disease_name, image_count=dataset_size, offset=685)
        for image in images:
            try:
                if algorithm == 'kaze':
                    image_features = extract_features_kaze(image)
                    test_features.append(image_features)
                elif algorithm == 'orb':
```

```
            image_features = extract_features_orb(image)
            test_features.append(image_features)
        else:
            image_features = extract_features_hog(image)
            test_features.append(image_features)
        test_labels.append(disease_name)
    except AttributeError as e:
        continue
test_features = np.array(test_features)
test_labels = np.array(test_labels)
test_features = StandardScaler().fit_transform(test_features)
test_labels = LabelEncoder().fit_transform(test_labels)
test_features, test_labels = shuffle(test_features, test_labels, random_state=34)

return  test_features, test_labels
```

Appendix 6. Code for Confusion Matrix

```
def create_confusion_matrix_labelled(predictions, labels, diseases=['common_rust',
'healthy', 'leaf_spot', 'nothern_leaf_blight']):
    """generates the confusion matrix


    parameters
    ----------
    returns
    -------
    confusion_matrix : numpy array
        The confusion matrix
    """
    cm = confusion_matrix(predictions, labels)


    plt.Figure(Figsize=(12,10))
    sns.heatmap(cm, annot=True, cbar=True)
    tick_marks = np.arange(len(diseases))
    plt.xticks(tick_marks, diseases, rotation=45)
    plt.yticks(tick_marks, diseases, rotation=45)
    plt.title('Confusion Matrix Visualization')
    plt.ylabel('True Label')
    plt.xlabel('Predicted Label')
    plt.show()
```

Appendix 7. Code for Training the Developed Model

```
def train_model(model, train_features, train_labels):
    """Trains the given model

    Uses cross-validation
    parameters
    ----------
    model
    train_features
    """

    start_time = time.time()
    train_scores = scores = cross_val_score(model, train_features, train_labels,
scoring='accuracy', cv=10)
    stop_time = time.time()
    train_time = round((stop_time - start_time), 3)
    train_score = round(np.mean(train_scores), 3)
    return train_score, train_time
```

Appendix 8. Code for Testing the Developed Model

```
def test_model(model, test_features, test_labels):
    """Trains the given model

    Uses cross-validation
    parameters
    ----------
    model
    train_features
    """

    test_predictions = cross_val_predict(model, test_features, test_labels, cv=10)
    test_score = round(accuracy_score(test_labels, test_predictions), 3)
    prescision = round(precision_score(test_labels, test_predictions, average='micro'),
3)
    recall = round(recall_score(test_labels, test_predictions, average='micro'), 3)
    f_1 = round(f1_score(test_labels, test_predictions, average='micro'), 3)
    return test_score
```

Appendix 9. Code for Generating Classification Report

```
def model_classification_report(model, test_features, test_labels,
diseases=['common_rust', 'healthy', 'leaf_spot',
                                        'nothern_leaf_blight']):
    predictions = model.predict(test_features)
    encoder = LabelEncoder()
    encoded_labels= encoder.fit_transform(diseases)
    decoded_predictions = encoder.inverse_transform(predictions)
    decoded_test_labels = encoder.inverse_transform(test_labels)
    print(classification_report(decoded_test_labels, decoded_predictions,
            labels=['common_rust', 'healthy', 'leaf_spot', 'nothern_leaf_blight']))
```

Appendix 10. Code for Model Performance

```
def model_perfomance(model, name, train_set=(hog_train_features,
hog_train_labels),
                  test_set=(hog_test_features, hog_test_labels)):
    """Model perfomance on the hog dataset

    parameters
    ----------
    model : scikit-learn Classifier
        Model to be tested
    returns
    -------
    model_perfomance : pandas DataFrame
    """

    #Assert dataset shape and type

    import warnings
    warnings.filterwarnings('ignore')

    #Training accuracy and Time
    print(f'**********{name}**********')
    print('\n')
    train_score, train_time = train_model(model, train_set[0], train_set[1])

    #Test Accuracy
    model.fit( train_set[0], train_set[1])
    test_score = test_model(model, test_set[0], test_set[1])

    #The model perfomance
```

```
    df = pd.DataFrame({'Model': [name], 'Train Accuracy': [train_score], 'Test Accuracy': [test_score], 'Train Time':[train_time]})
    print(df)
    print('\n')


    #The classification report
    model_classification_report(model, test_set[0], test_set[1])
    print('\n')
    #The confusion matrix
    predictions = model.predict(test_set[0])
    #create_confusion_matrix(predictions, test_set[1])
    create_confusion_matrix_labelled(predictions, test_set[1])
    print('\n')
    print('*********The Key**********')
    print('\t0: Common Rust\n\t1: Healthy\n\t2: Leaf Spot\n\t3: Nothern Leaf Blight')
```

Appendix 11. Approval of Research Proposal and Supervisors

MURANG'A UNIVERSITY OF TECHNOLOGY
DIRECTORATE OF POSTGRADUATE STUDIES

P.O. BOX 75 - 10200, MURANG'A Email: bps@mut.ac.ke

Ref: MUT/ARP/PGS/20/2020/VOL.I **Date:** 3rd February 2022

Dear Vincent Mbandu (SC401/5430/2020),

RE: APPROVAL OF RESEARCH PROPOSAL AND SUPERVISORS.

I am pleased to inform you that the Directorate of Postgraduate Studies on 17th January 2022 considered and approved your Masters research proposal entitled ***"A Model for Enhanced Image Classification Accuracy in Maize Leaf Diseases"*** and appointed the following as supervisors:

1. **Dr. Geoffrey Mariga -Murang'a University of Technology**
2. **Dr. John Ndia - Murang'a University of Technology**

You may now proceed with your data collection subject to obtaining research permit from NACOSTI, if required. You should also begin consulting your supervisors and submit through them quarterly progress reports to the Director Postgraduate Studies through your CoD and School Dean. Progress Reports can be accessed in the University Website.

It is the policy and regulations of the University that you observe deadlines. The guidelines on Postgraduate supervision can be accessed in the post graduate Handbook.

Your responsibilities as a student will include, among others;

i. Maintain regular consultation with your supervisor(s), at least once a month
ii. Submit quarterly reports on time, through your supervisors, CoD, Dean and to the Director of Postgraduate Studies;
iii. Ensure quality work all through;
iv. Present your research findings at 2- 3 seminars/conferences prior to thesis examination.
v. Publish one article from your research findings in a refereed journal prior to thesis examination

For any further clarification, please contact the undersigned.

Yours Sincerely,

DIRECTORATE OF POSTGRADUATE STUDIES
MURANG'A UNIVERSITY OF TECHNOLOGY
03 FEB 2022
P. O. Box 75 - 10200, MURANG'A
Tel: 0771463515, Email: info@mut.ac.ke

PROF. GEOFFREY MUCHIRI, PhD
DIRECTOR, POSTGRADUATE STUDIES

Cc Registrar (ASA)
Dean (SCIT)

MUT IS ISO 9001:2015 CERTIFIED

Appendix 12. NACOSTI Research License

REPUBLIC OF KENYA

NATIONAL COMMISSION FOR SCIENCE,TECHNOLOGY & INNOVATION

Ref No: **859626**

Date of Issue: **16/February/2022**

RESEARCH LICENSE

This is to Certify that Mr.. Vincent Mbandu Ochango of Murang'a University of Technology, has been licensed to conduct research in Kiambu on the topic: An Enhanced Image Classification Accuracy Model for Maize Leaf Diseases for the period ending : 16/February/2023.

License No: **NACOSTI/P/22/15663**

859626

Applicant Identification Number

Director General

NATIONAL COMMISSION FOR SCIENCE,TECHNOLOGY & INNOVATION

Verification QR Code

NOTE: This is a computer generated License. To verify the authenticity of this document, Scan the QR Code using QR scanner application.

THE SCIENCE, TECHNOLOGY AND INNOVATION ACT, 2013

The Grant of Research Licenses is Guided by the Science, Technology and Innovation (Research Licensing) Regulations, 2014

CONDITIONS

1. The License is valid for the proposed research, location and specified period
2. The License any rights thereunder are non-transferable
3. The Licensee shall inform the relevant County Director of Education, County Commissioner and County Governor before commencement of the research
4. Excavation, filming and collection of specimens are subject to further necessary clearence from relevant Government Agencies
5. The License does not give authority to tranfer research materials
6. NACOSTI may monitor and evaluate the licensed research project
7. The Licensee shall submit one hard copy and upload a soft copy of their final report (thesis) within one year of completion of the research
8. NACOSTI reserves the right to modify the conditions of the License including cancellation without prior notice

National Commission for Science, Technology and Innovation
off Waiyaki Way, Upper Kabete,
P. O. Box 30623, 00100 Nairobi, KENYA
Land line: 020 4007000, 020 2241349, 020 3310571, 020 8001077
Mobile: 0713 788 787 / 0735 404 245
E-mail: dg@nacosti.go.ke / registry@nacosti.go.ke
Website: www.nacosti.go.ke

Appendix 13. Publications

The following research papers were published from this thesis.

[1] V. M. Ochango G. M. Wambugu, J. G. Ndia, "Comparative Analysis of Machine Learning Algorithms Accuracy for Maize Leaf Disease Identification," *International Journal of Formal Sciences: Current and Future Research Trends,* Amman, Jordan., 2022.

[2] V. M. Ochango G. M. Wambugu, J. G. Ndia, "Feature Extraction using Histogram of Oriented Gradients for Image Classification in Maize Leaf Diseases," *International Journal of Computer and Information Technology*, Sitapur Road, Lucknow India, 2022.