Feature Extraction using Histogram of Oriented Gradients for Image Classification in Maize Leaf Diseases

Vincent Mbandu Ochango
School of Computing and Information Technology
Murang'a University of Technology
Murang'a, Kenya
Email:ochangovincent [AT] gmail.com

John Gichuki Ndia
School of Computing and Information Technology
Murang'a University of Technology
Murang'a, Kenya
Email:jndia [AT] mut.ac.ke

Geoffrey Mariga Wambugu
School of Computing and Information Technology
Murang'a University of Technology
Murang'a, Kenya
Email:gmariga [AT] mut.ac.ke

Abstract—**The paper presents feature extraction methods and classification algorithms used to classify maize leaf disease images. From maize disease images, features are extracted and passed to the machine learning classification algorithm to identify the possible disease based on the features detected using the feature extraction method. The maize disease images used include images of common rust, leaf spot, and northern leaf blight and healthy images. An evaluation was done for the feature extraction method to see which feature extraction method performs best with image classification algorithms. Based on the evaluation, the outcomes revealed Histogram of Oriented Gradients performed best with classifiers compared to KAZE and Oriented FAST and rotated BRIEF. The random forest classifier emerged the best in terms of image classification, based on four performance metrics which are accuracy, precision, recall, and F1-score. The experimental outcome indicated that the random forest had 0.74 accuracy, 0.77 precision, 0.77 recall, and 0.75 F1-score.**

Keywords- Feature extraction, ORB, HOG, KAZE, Image classification, machine learning, and classifier

I. INTRODUCTION

The purpose of this paper is to identify maize disease through feature extraction and classify the maize disease images from the features extracted using machine learning algorithms. The farmers are usually unable to detect diseases on their crops by just looking at them. This leads to damages that cost farmers a lot of money. Using captured images of crops to tell whether or not they are disease through image classification using a machine learning algorithm and if they are disease, the machine learning algorithm to tell the particular disease affecting the plant is the solution to this problem. The farmer can then purchase the right medicine for their plants. From this research paper, the features from the images are extracted using ORB, HOG, and KAZE method, and once the features are extracted, they are passed to the machine learning image classification algorithm which can tell the particular maize disease affecting the crops. A comparison of three methods was done and the HOG feature extraction method performed better with image classification algorithms hence the researcher decided to work out with HOG as a feature extraction method. HOG feature descriptor extracts key points from images and throws away information that is not useful and this is what is considered dimensionality reduction. These key points are the ones that differentiate an image from the other images since they are unique for every image and clearly distinguish an image from the other images. The feature descriptor converts an image to a vector which is an array and this feature vector is an input value to the classification algorithms [4], [5], [6], [7]. Before HOG calculates the descriptor it resizes the image window to an aspect ratio of 1:2 and most probably 64 × 128 and this process is known as image preprocessing. The main reason for resizing the image to 64 × 128 size is that when extracting features the image needs to be divided into a patch of 8 × 8 and 16 ×16. The histogram of Gradient is calculated by first calculating the vertical and horizontal gradient which is achieved by applying filters to an image. A lot of unnecessary information such as colored background is usually removed by gradient image and only the shape and the edges of the image remain and this is exactly what a feature descriptor does to an image. Other feature descriptors usually recognize if an element in an image is an edge or not in case of edge features but HOG goes further and extracts the magnitude and direction of the edges thus being able to provide the edge direction and that is why it is called Histogram of Oriented Gradient [1], [2], [3]. Calculating gradient means calculating the direction of x and y pixel values for the image. A patch is usually taken from an image and a gradient is calculated for the patch taken. The pixel matrix is usually generated for each small patch taken from an image. For every pixel value in the matrix, we then calculate the change in x and y direction which is denoted by G_x and G_y respectively. And the process usually gives us the new matrices, one storing G_x and the other one storing $G_{y.}$ The step that follows is to find now the direction and magnitude of all elements in an image. And the process is done by calculating the Total Gradient Magnitude (T.G.M). And the following equation helps us in calculating the total gradient magnitude;

T.G.M= $\sqrt{}[(\ (G_y)^2 + G_x)^2 +]$

The following mathematical equation shows how the direction of the pixel is calculated;

Θ=arctan(G_x / G_y)

Finally, the histogram is calculated for each pixel using the magnitude and the direction of each pixel. The HOG features are the ones that act as the input value for the image classification algorithm. The rest of this paper is structured as follows; Section II provides related work on feature extraction methods and image classification algorithms. Section III explains how the feature extraction was done, the hyperparameters used with classification algorithms, and how the cross-validation was done to reduce the overfitting of the classification algorithms. Section IV provides an explanation of the experimental results obtained and the best classifier. Section V the conclusion and future work.

II. RELATED WORKS

The use of feature extraction methods in this paper as the input to the machine learning algorithms to identify maize disease images is widely used. The metrics used to measure the classification of disease images differ for each machine learning algorithm. By using computer vision different methods are utilized in the identification of crop infections. Extracting features from images is one of the techniques that is used to detect diseases from the plant.

Pujai et al. [8], [9], [10], [16] classified maize disease images by carrying out an experiment that uses ANN and Support Vector Machine. The two classifiers were trained based on image features extracted using a feature extraction method. The results demonstrated that SVM performed best with the image features extracted compared to ANN. The SVM classifier had an accuracy of 0.9217 and 0.874 for the ANN classifier.

Yaktundimath et al. [11], [12], [16] identified three kinds of cereal plants and classified them using machine learning algorithms. Specifically, they used two classifiers to classify jowar, maize, and wheat leaf diseases by using the fungal symptoms associated with each leaf disease. Normal, smut, powdery mildew, leaf spot, and leaf blight maize leaf disease were collected by the authors and were used in the experiment. The authors followed certain steps to identify and categorize fungal disease symptoms; acquired normal and fungal affected 750 JPG formatted images. After that, the images are preprocessed and then the image segmentation is done. The leaf disease images are used to extract features using the Color Co-occurrence matrix algorithm and the features extracted act as the input value to the machine learning algorithms and for program interface MATLAB tool was used. The classification accuracy for SVM and ANN machine learning algorithms used was 83.83% and 77.75% respectively. To identify and classify cereals' fungal disease the authors found out that the SVM algorithm is the best to use since it is more accurate than the Artificial Neural network [22]. Other feature extraction methods and machine learning algorithms for classifying leaf disease images were recommended by the researcher as future work that needs to be done.

Zhang et al. [14], [15], [23] used machine learning algorithms to categorize 5 types of maize crop diseases. They collected 20 images for each category of maize leaf disease which aided in the experiment and all the images collected were used both for training and testing purposes. However, the authors did not mention the five-leaf diseases used in the experiment. The images collected were scaled and normalized in terms of orientation and histogram equilibrium was used to convert them to 32 by 32 pixels with each image having a white background due to each pixel having a 255 gray level. The experiment was conducted by first collecting the images of five different types of leaf diseases by using digital cameras and then the images were segmented. Features were then extracted from images and passed onto the KNN algorithm which classified the features according to the respective leaf diseases hence producing class labels for each image feature. The experiment for image classification was done 50 times and the results showed the classification accuracy was above 80%. The researcher finally proposed that future work should be done by increasing the training data set and extracting key points from an image since the key points clearly distinguish an image from one another [16], [17].

Xiaoyang et al. [18], [19], [20], [23] did research in china farm area that classifies four types of maize leaf diseases and the researchers followed these steps; Under sunlight conditions, the researchers used digital cameras to collect JPG types of maize disease images and to obtain the information from the images they are converted to BMP format and later used a thresholding value to segment the images. The standard deviation and mean are calculated after the images are converted from RGB to HIS and the researchers finally classified the images using the GA-SVM algorithm. The maize leaf diseases were classified also using support vector machine and RBF kernel function. The machine learning algorithm which classified the maize leaf disease was measured in terms of precision and the GA-SVM algorithm had a precision of between 88.72% and 92.59% for each image for maize leaf disease and the support vector machine had a precision of between 69.63% and 90.09%. Many experiments have been done on image classification using support vector machine algorithm and even accuracy and precision have been measured, further research needs to be done with other machine learning algorithm so as to be able to verify if really support vector machine is best to be used in image classification. Also, a further experiment needs to be done in order to assess the machine learning algorithms in terms of recall and f1-score. This has propelled the current study to increase the training dataset and use more algorithms in order to come to a conclusion which algorithm is best when it comes to maize leaf disease classification. And the current study also wants to explore if there are other better algorithms than support vector machine when it comes to maize leaf disease classification.

III. METHODOLOGY

Dataset Description

The research used a maize disease data augmented dataset which is public and was obtained from the Kaggle website and the data set contained training and testing images. The dataset consisted of common rust, leaf spot, northern leaf blight disease images, and healthy images. The whole training dataset consisted of 7308 images,1634 images for leaf spot, 1907 images for common rust,1908 images for northern leaf blight, and 1859 for healthy leaf images [21]. The whole testing dataset consisted of 1826 images, 407 images for leaf spot, 477 images for common rust, 477 images for northern leaf blight, and 465 for healthy leaf images. And because of time and limited resources, the researcher decided first to work with 300 images from the training data set which will be used for training for each category of disease resulting in 1200 images in total. This is because the process of generating features from each image takes a lot of time and consumes a lot of computer resources hence the researcher decided first to work with a total of 1200 images. The testing data set had a total of 1826 images and we used 30 images from each category of disease.

Numerical Feature Extraction

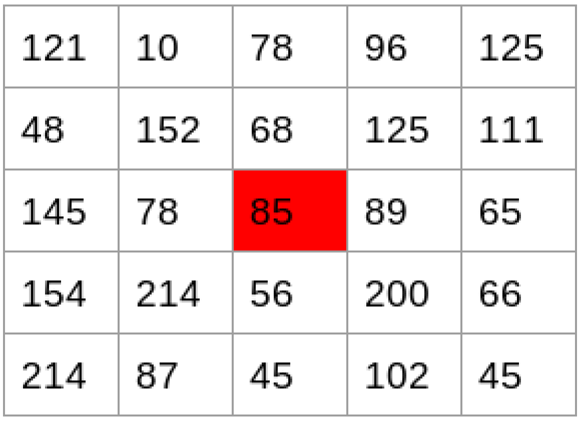
Feature extraction from the image was done using ORB, KAZE, and HOG feature extraction methods. The features extracted were in terms of integers and these integers were passed to the machine learning classifier algorithms. From each image, 4096 key points were extracted and acted as input values to the machine learning classifiers. These attributes extracted were key points to each image since the key point is a feature that is unique to an image and can be detected despite the change in the image. The features extracted by each method were passed to the classification algorithms to find out which feature extraction method performs better with machine learning algorithms. The machine learning algorithms were measured in terms of accuracy to determine which feature extraction method works best with them. Based on the results and analysis of the classification algorithms it was found out that HOG performs better with the classification algorithms.

HOG Feature Extraction Approach

Image Preprocessing

Assume we have an image of 180 × 280 the first step that is done with the HOG feature descriptor is to resize the image into a ratio of 1:2 and most probably the image is resized to 64 × 128 and this process is known as image preprocessing. Image preprocessing is important since the image will be broken further into 8 by 8 and 16 by 16 pixel window to be able to generate the features from the image. An image size of pixel ratio 1:2 makes the calculation of feature extraction easier and faster.

The change in x and y direction of every pixel is 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 an assumed example of 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

Applied Machine Learning - Beginner to Professional. (n.d.). Retrieved from https://courses.analyticsvidhya.com/courses/applied-machine-learning-beginner-to-professional/?utm_source=blog&utm_medium=understand-math-HOG-feature-descriptor

As shown in the pixel matrix the value of pixel 85 is highlighted in red and that is the one that we will work with to be able to demonstrate how the gradient of a pixel is calculated. The change in the x-direction of the pixel 85 is 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 the pixel value 85. The same thing happens for change in y-direction for the pixel value 85 which is 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 is done for all the pixel values in the matrix and a new matrix is obtained with these new values which help us calculate the direction and the magnitude.

Orientation and Magnitude

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

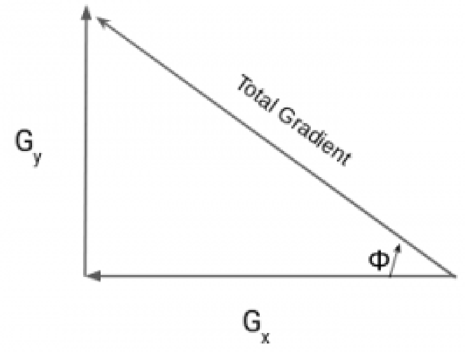


Figure 1. Orientation and Magnitude

Fig 1 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 is 8 and 11 respectively.

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

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

= 13.6

The pixel direction is calculated as;

$\operatorname{Tan}(\Theta) = (G_x / G_y)$

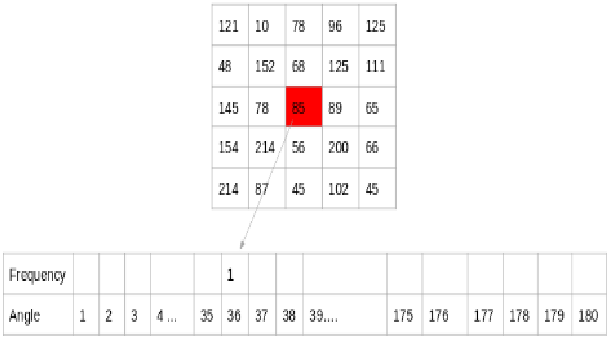
$\Theta = \arctan(G_x / G_y)$

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

= 36

Generating Histogram

After having the direction and magnitude of each image element value now the magnitude and direction is 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	

Applied Machine Learning - Beginner to Professional. (n.d.). Retrieved from https://courses.analyticsvidhya.com/courses/applied-machine-learning-beginner-to-professional/?utm_source=blog&utm_medium=understand-math-HOG-feature-descriptor

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

Histogram of Gradients in 8 × 8 Image Patch

HOG divides the image into smaller parts or an image patch of 8 × 8 cells and calculates the features for this every 8 × 8 cells which represent the histogram for the whole image. A matrix of 9 × 1 is usually obtained for each cell after generating the histogram from an image divided into 8 × 8 cells. The histogram id finally normalized after extracting HOG features from 8 × 8 cells.

Normalize Gradients

Gradients are usually normalized since in the 8 × 8 cells some portion of the image usually appears bright than the other portion hence the gradients are normalized by taking 16 × 16 blocks which help in reducing the variation in light. To create 16 × 16 blocks we combine the four 8 × 8 cells into one and remember all eight by eight cell has a matrix of 9 × 1 for a histogram so when we combine four 9 × 1 matrix we end up with a single 36 × 1 matrix. We sum the square of each value in the matrix and find the square root and then the result is 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

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

And this will be 36 × 1 matrix normalized vector size.

Complete Image Features

The final image features are done by combining the features for a 16 × 16 block features. And this you need to know how many 16 × 16 blocks can be gotten from an image size of 64× 128 and the answer is you need 7 × 15 blocks of 16 × 16 block to make up an image of 64 × 128. Since a block of 16 × 16 has 36 × 1 features then the entire 64 × 128 image has 7 × 15 × 36 × 1 features which are equal to 3780 HOG features.

Hyperparameter Tuning

This was done to be able to work with a set of optimal hyperparameters for each classification algorithm. To work with classification algorithms you need to set parameters for each algorithm before the process of learning begins. The penalty in logistic regression and loss in stochastic gradient descent is some of the examples of hyperparameters for the classification algorithm. The tuning strategy used for our case for optimizing hyperparameters for each classification algorithm was grid search. Hyperparameter optimization according to research usually improves the performance of the machine learning algorithm. The grid search method used for hyperparameter tuning works by exhaustively searching through a specified set of hyperparameters. The optimal combination of parameters supplied is guaranteed by using a grid search and one of the major disadvantages of grid search is that it is computationally expensive and time-consuming. During the implementation of hyperparameter tuning is that before we ran the grid search method in the Jupiter notebook, we first defined our grid of parameters to search over.

Cross-Validation

The validation is done to test your classifier if it performs well on the data that it has never seen before introducing your classifier to the training data set. Cross-validation is done to get an assurance that your classification algorithm works better and predicts correctly in case it is given data that it has never seen before. The method also helps you to know if the classification algorithm is either underfitting or overfitting the data.

Using part of the training data for validating your model usually results in an underfitting problem since there is never enough data for training your model. This, in turn, increases error induced by bias and we risk losing important trends in training data set and patterns which results after reducing the training data. Hence, we require K-Fold cross-validation which leaves the part of the data for validation and the other for training. The method puts together a k-1 subset to be used for training and k subset to be used for testing or validation purposes. The total effectiveness of the model is obtained by averaging error estimation for all k trials. With this method, every data gets to be in training set k-1 times and gets to be in the testing set once. This reduces variance significantly since most of the data is used in the validation set and most of the data is used in the training set which reduces bias significantly. The effectiveness of this method is seen since it interchanges the test set or validation set with the training set.

Steps followed;

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.

IV. EXPERIMENTAL RESULTS AND DISCUSSION

Feature Extraction

Fig 2 illustrates feature generation using ORB method for common rust disease image

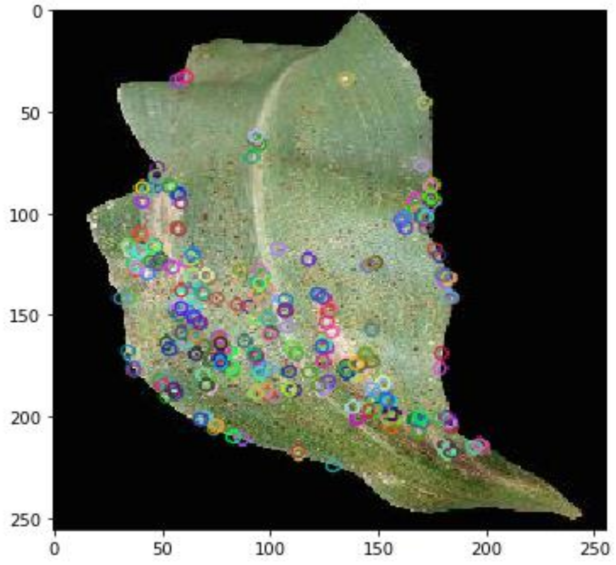


Figure 2. Detecting key points from common rust disease image using ORB method

Feature generation using the KAZE method for common rust disease image is shown in Fig 3.

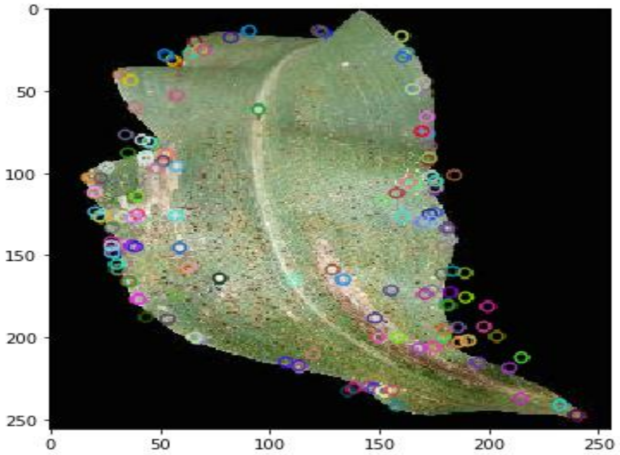


Figure 3. Detecting key points from common rust disease image using KAZE method

The dataset had images for common rust, leaf spot, northern leaf blight disease, and images for healthy leaves. The features extracted from each image were 4096 features and we narrowed it down to 300 images of each disease since extracting features for the whole images in the dataset takes a lot of time. After the features were extracted using the three feature extraction methods, the machine learning algorithms were trained based on the features extracted. This training was done basically to associate each feature with their respective disease type and hence making the algorithms to learn from the training data.

Machine Learning Algorithms

Machine learning algorithms used for a comparison of feature extraction methods were Random Forest, Logistic Regression, AdaBoost, Bagging, Gradient Boosting, Bernoulli NB, Gaussian NB, K-Nearest Neighbors, Neural Network, Linear SVC, and Support Vector Machine. First of all the researcher looked at the accuracy level of each classifier based on each feature extraction method and this was to be able to identify which feature extraction method performs better. Features generated using KAZE, ORB, and HOG methods and how the classifier performed in terms of accuracy with each of the feature extraction methods. Since feature extraction takes a lot of time we decided first to extract features from 32 images for each disease type to get an insight on which feature extraction method works best with the classifiers. Table 1 shows the accuracy values for various algorithms based on features extracted using KAZE, ORB, and HOG, respectively.

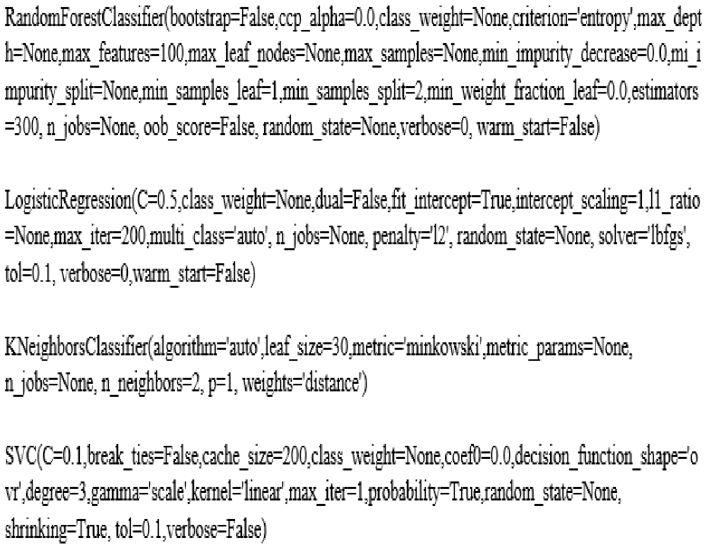
Table 1: Classifier Accuracy using features generated by KAZE, ORB, and HOG methods

Model	**KAZE**	**ORB**	**HOG**
Gaussian NB	0.718	0.342	0.564
Random Forest	0.641	0.395	0.692
Gradient Boosting	0.641	0.342	0.641
Logistic Regression	0.615	0.316	0.769
Bernoulli NB	0.615	0.395	0.538
K-Nearest Neighbors	0.615	0.263	0.538
Neural Network	0.615	0.342	0.744
Linear SVC	0.615	0.421	0.718
Bagging	0.590	0.526	0.513
AdaBoost	0.513	0.263	0.410
Support Vector Machine	0.513	0.289	0.667
Average Accuracy:	0.608	0.354	0.618

As seen in Table 1, the base models perform so badly with the ORB feature extraction method. HOG produces the best performance as shown by the average performance of 0.618, followed by KAZE at 0.608, and ORB at 0.354. Based on the HOG performance, it was decided to work with the HOG method for feature extraction.

Hyperparameter Tuning

The researcher looked at the optimal hyperparameters to work with the classifiers. The hyperparameter is a parameter that is set for each machine learning algorithm before it starts 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;

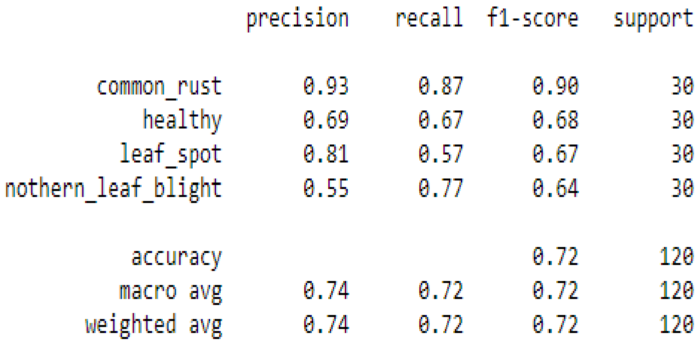


```
RandomForestClassifier(bootstrap=False,ccp_alpha=0.0,class_weight=None,criterion='entropy',max_dept
h=None,max_features=100,max_leaf_nodes=None,max_samples=None,min_impurity_decrease=0.0,mi_i
mpurity_split=None,min_samples_leaf=1,min_samples_split=2,min_weight_fraction_leaf=0.0,estimators
=300, n_jobs=None, oob_score=False, random_state=None,verbose=0, warm_start=False)

LogisticRegression(C=0.5,class_weight=None,dual=False,fit_intercept=True,intercept_scaling=1,l1_ratio
=None,max_iter=200,multi_class='auto', n_jobs=None, penalty='l2', random_state=None, solver='lbfgs',
tol=0.1, verbose=0,warm_start=False)

KNeighborsClassifier(algorithm='auto',leaf_size=30,metric='minkowski',metric_params=None,
n_jobs=None, n_neighbors=2, p=1, weights='distance')

SVC(C=0.1,break_ties=False,cache_size=200,class_weight=None,coef0=0.0,decision_function_shape='o
vr',degree=3,gamma='scale',kernel='linear',max_iter=1,probability=True,random_state=None,
shrinking=True, tol=0.1,verbose=False)
```

Figure 4. Hyperparameter Tuning

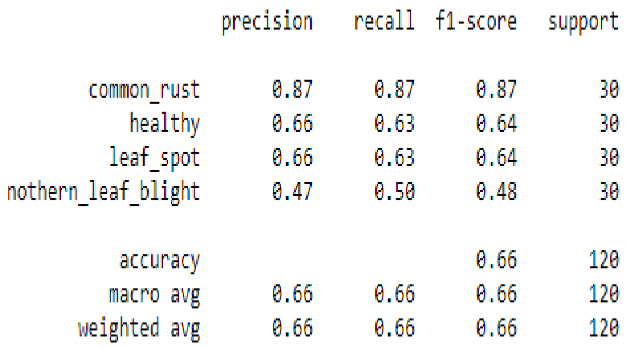
Classification Report

The classification report enabled us to know if our classification algorithm can classify the images well and be able to measure the quality of prediction the algorithms used. And this report was done for a few algorithms as shown in the report since it takes time to execute the classification report. The testing of three classifiers as shown in Fig 5, Fig 6, Fig 7, and Fig 8 was done using 30 images for each category of the disease from the testing data set and the results are shown in Fig 5, Fig 6, Fig 7, and Fig 8.



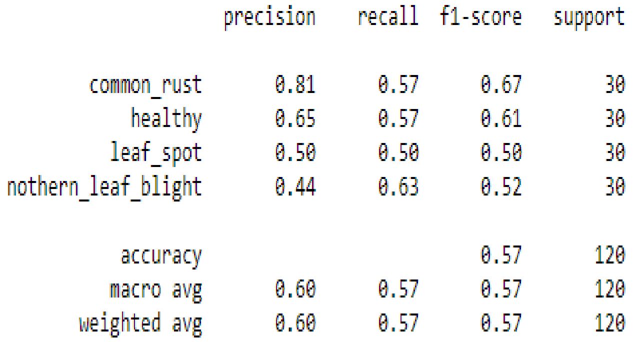
```
                     precision    recall  f1-score   support

         common_rust       0.93      0.87      0.90        30
             healthy       0.69      0.67      0.68        30
           leaf_spot       0.81      0.57      0.67        30
nothern_leaf_blight       0.55      0.77      0.64        30

            accuracy                           0.72       120
           macro avg       0.74      0.72      0.72       120
        weighted avg       0.74      0.72      0.72       120
```

Figure 5. Random Forest Classification Report



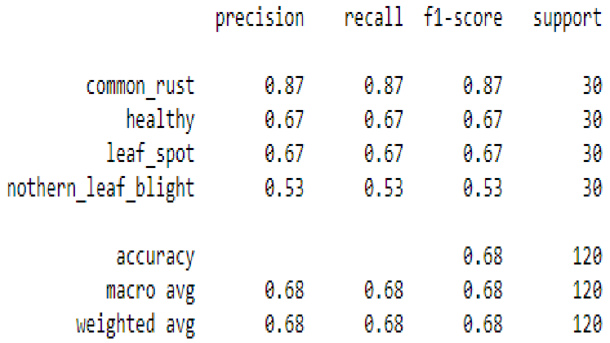
```
                   precision    recall  f1-score   support

       common_rust       0.87      0.87      0.87        30
           healthy       0.66      0.63      0.64        30
         leaf_spot       0.66      0.63      0.64        30
nothern_leaf_blight       0.47      0.50      0.48        30

          accuracy                           0.66       120
         macro avg       0.66      0.66      0.66       120
      weighted avg       0.66      0.66      0.66       120
```

Figure 6. Logistic Regression Classification Report



```
                   precision    recall  f1-score   support

       common_rust       0.81      0.57      0.67        30
           healthy       0.65      0.57      0.61        30
         leaf_spot       0.50      0.50      0.50        30
nothern_leaf_blight       0.44      0.63      0.52        30

          accuracy                           0.57       120
         macro avg       0.60      0.57      0.57       120
      weighted avg       0.60      0.57      0.57       120
```

Figure **7. Gaussian Naïve Bayes Classification Report**



```
                   precision    recall  f1-score   support

       common_rust       0.87      0.87      0.87        30
           healthy       0.67      0.67      0.67        30
         leaf_spot       0.67      0.67      0.67        30
nothern_leaf_blight       0.53      0.53      0.53        30

          accuracy                           0.68       120
         macro avg       0.68      0.68      0.68       120
      weighted avg       0.68      0.68      0.68       120
```

Figure 8. Support Vector Classifier Classification Report

As you can see from the classification report in Fig 5, Fig 6, Fig 7, and Fig 8 that all the classifiers were classifying the 30 images of each image category as shown in the support column of the classification report.

Overall Classification Metrics

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 following is the results gotten for each model;

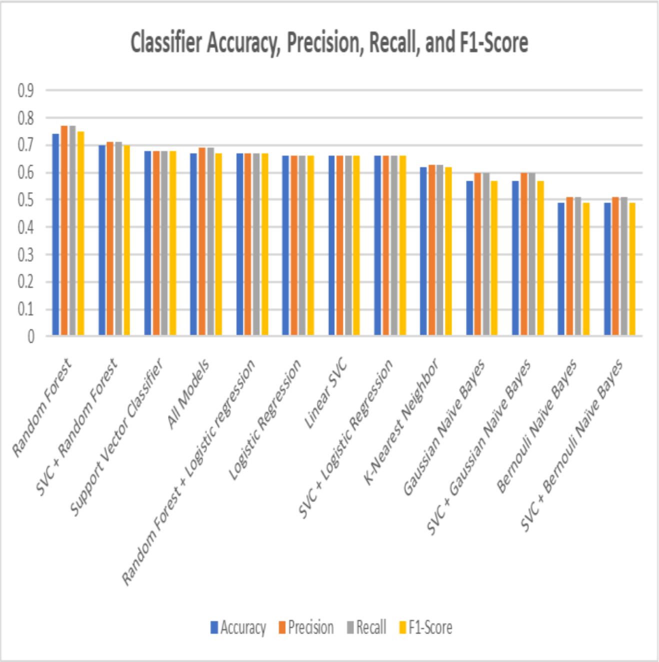


Figure 9. Overall Classification Metrics

The results in Fig 9 above show the best performing model(Random Forest) with the test data set while Bernoulli Naïve Bayes and a combination of Support Vector Classifier and Bernoulli Naïve Bayes performed badly.

Discussion

The experimental results shown above were mainly used to investigate different machine learning algorithms and provide a way forward on the classification algorithm to be used in identifying maize leaf diseases. Figure 2 and Figure 3 are used to demonstrate how feature extraction methods are used to extract key points from an image. These key points are the ones that act as an input to the machine learning algorithms. Since after the features are extracted they are passed to the classification algorithms. Table 1 shows how three feature extraction methods perform with different classification algorithms. As can be seen from table 1 Gaussian NB performs best with KAZE, Bagging performs best with ORB and Logistic Regression performs best with HOG. The average accuracy for each feature extraction method on how it performed with different classification algorithms was calculated and the results showed that HOG performed well. It performs well than the other feature extraction methods because;

- It mainly concentrates on the shape of the image and be able to offer the direction of the edges.

- The direction and magnitude of the edges are calculated after dividing the image into smaller parts called cells.
- The Histogram is developed from the smaller parts of the images by using the magnitude and direction of the pixel values.
- It is more accurate because the gradients are usually normalized since in the 8 × 8 cells some portion of the image usually appears bright than the other portion hence the gradients are normalized by taking 16 × 16 blocks which help in reducing the variation in light.

Due to HOG performance, it was concluded to work with it the entire experimental work. After choosing the HOG the dataset was increased from 32 to 300 for each maize disease category hence subjecting the classification algorithms to a more increased training dataset which will enable the algorithms to make predictions with minimal errors. Figure 5,6,7, and 8 shows the classification reports, and these reports are used to tell whether the algorithms are making good or bad predictions based on the testing dataset they are subjected to. Each algorithm was subjected to a test data set of 30 images from each category of maize diseases and a classification report was obtained after the predictions. And from Figure 9 random forest emerged the best in terms of classifying the maize disease images. The main reason why the random forest produced good results 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. Further investigation needs to be done by increasing the training dataset and the testing data set and comparing the results and making a new conclusion based on the increased dataset.

V. CONCLUSION AND FUTURE WORK

This paper provides a solution to farmers for them to be able to identify maize disease through machine learning algorithms. The research has tried to find out which feature extraction method can perform better with machine learning classification algorithms. And based on the results we find out that the HOG feature extraction method performs best with the classification algorithms compared to KAZE and ORB hence enabling the researcher to work with the HOG method. There was also a comparison of machine learning algorithms and the random forest algorithm emerged the best. The results indicated that the random forest had 0.74 accuracy, 0.77 precision, 0.77 recall, and 0.75 F1-score. From the classification report, the random forest classifier also was seen to be the best and in conclusion, based on my result the researcher proposed working with the HOG feature extraction method and the random forest algorithm when it comes to maize disease identification since the classifier produces better results with HOG features.

REFERENCES

[1] Amato, G., & Falchi, F. (2017, September). kNN based image classification relying on local feature similarity. In *Proceedings of the Third International Conference on SImilarity Search and APplications* (pp. 101-108).

[2] Bosch, A., Zisserman, A., & Munoz, X. (2018, October). Image classification using random forests and ferns. In *2017 IEEE 11th international conference on computer vision* (pp. 1-8). Ieee.

[3] Chen, P. Y., Huang, C. C., Lien, C. Y., & Tsai, Y. H. (2013). An efficient hardware implementation of HOG feature extraction for human detection. *IEEE Transactions on Intelligent Transportation Systems*, *15*(2), 656-662.

[4] Chen, Y. S., Chien, J. C., & Lee, J. D. (2016, September). KAZE-BOF-based large vehicles detection at night. In *2016 International Conference On Communication Problem-Solving (ICCP)* (pp. 1-2). IEEE.

[5] 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.

[6] Gan, G., & Cheng, J. (2018, December). Pedestrian detection based on HOG-LBP feature. In *2014 Seventh International Conference on Computational Intelligence and Security* (pp. 1184-1187). IEEE.

[7] Geismann, P., & Schneider, G. (2018, June). A two-staged approach to vision-based pedestrian recognition using Haar and HOG features. In *2011 IEEE Intelligent Vehicles Symposium* (pp. 554-559). IEEE.

[8] Goh, K. S., Chang, E., & Cheng, K. T. (2011, October). SVM binary classifier ensembles for image classification. In *Proceedings of the tenth international conference on Information and knowledge management* (pp. 395-402).

[9] 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).

[10] Kobayashi, T. (2015). 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).

[11]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.

[12] Li, W., Qian, Y., Loomes, M., & Gao, X. (2015, March). The application of KAZE features to the classification echocardiogram videos. In *International Workshop on Multimodal Retrieval in the Medical Domain* (pp. 61-72). Springer, Cham.

[13] Li, Y., & Cheng, B. (2019, August). An improved k-nearest neighbor algorithm and its application to high resolution remote sensing image classification. In *2009 17th International Conference on Geoinformatics* (pp. 1-4). Ieee.

[14] 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.

[15] 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 2013; 6: 321-330.

[16] P Jagadeesh, R Yakkundimath, et al. SVM and ANN Based Classification of Plant Diseases Using Feature Reduction Technique. International Journal of Interactive Multimedia and Artificial Intelligence 2016; 3: 6-14.

[17] Ramteke, R. J., & Monali, Y. K. (2018). Automatic medical image classification and abnormality detection using k-nearest neighbour. *International Journal of Advanced Computer Research*, *2*(4), 190-196.

[18] Sanchez-Morillo, D., González, J., García-Rojo, M., & Ortega, J. (2018, April). Classification of breast cancer histopathological images using KAZE features. In *International Conference on Bioinformatics and Biomedical Engineering* (pp. 276-286). Springer, Cham.

[19] Spyrou, E., Le Borgne, H., Mailis, T., Cooke, E., Avrithis, Y., & O'Connor, N. (2015, September). Fusing MPEG-7 visual descriptors for image classification. In *International Conference on Artificial Neural Networks* (pp. 847-852). Springer, Berlin, Heidelberg.

[20] Umbaugh, S. E., Wei, Y. S., & Zuke, M. (2017). Feature extraction in image analysis. A program for facilitating data reduction in medical image classification. *IEEE engineering in medicine and biology magazine*, *16*(4), 62-73.

[21] Xia, J., Ghamisi, P., Yokoya, N., & Iwasaki, A. (2018). Random forest ensembles and extended multiextinction profiles for hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, *56*(1), 202-216.

[22] Xu, B., Ye, Y., & Nie, L. (2012, June). An improved random forest classifier for image classification. In *2012 IEEE International Conference on Information and Automation* (pp. 795-800). IEEE.

[23] Z Zhiyong, H Xiaoyang, et al. Image recognition of maize leaf disease based on GA-SVM. Chemical Engineering Transactions 2015; 46:199-204.