NJICE –National Journal on Information and Communication Engineering
ISSN: 2231-2099Volume 11 Issue 1
Jan-March 2021 Pages 14-30

FEATURE EXTRACTION AND IMAGE CLASSIFICATION FOR MAIZE LEAF DISEASE IDENTIFICATION USING MACHINE LEARNING

Dr. KANNAGI
Professor
Dr. S. K. MOULEESWARAN
Associate Professor
Dr.N. KALIMUTHU
Associate Professor
Dr. EDURU NAGARJUNA
Associate Professor
Department of Civil Engineering
Indira Gandhi Institute of Engineering and Technology
Nellikuzhi P.O, Kothamangalam, Ernakulam (Dist) Pincode 686691

Abstract: The paper discusses methods for extracting features and classification algorithms utilized for categorizing images of maize leaf diseases. It involves extracting features from images of maize diseases and employing machine learning classifiers to identify potential diseases based on the detected features. The images used encompass common rust, leaf spot, northern leaf blight, and healthy maize leaves. An assessment was conducted to determine the effectiveness of different feature extraction methods in conjunction with image classification algorithms. The evaluation revealed that Histogram of Oriented Gradients (HOG) outperformed KAZE and Oriented FAST and rotated BRIEF (ORB) in terms of compatibility with classifiers. Among the classifiers, the random forest classifier exhibited the highest performance in image classification, as evidenced by four metrics: accuracy, precision, recall, and F1-score. According to the experimental results, the random forest classifier achieved an accuracy of 0.74, precision of 0.77, recall of 0.77, and F1-score of 0.75.

Keywords: *maize leaf disease, feature extraction, image classification, machine learning, random forest classifier*

I. INTRODUCTION

The aim of this study is to detect maize diseases by extracting features and categorizing maize disease images based on the extracted features using machine learning algorithms. Farmers often struggle to identify diseases affecting their crops visually, leading to significant financial losses. Utilizing captured crop images for disease detection through image classification employing

NJICE –National Journal on Information and Communication Engineering
ISSN: 2231-2099Volume 11 Issue 1
Jan-March 2021 Pages 14-30

machine learning algorithms offers a solution to this challenge. Once a disease is identified, farmers can procure the appropriate treatment for their crops.

In this research, features are extracted from images using the ORB, HOG, and KAZE methods. These extracted features are then input into machine learning image classification algorithms to identify specific maize diseases affecting the crops. A comparison of the three methods revealed that the HOG feature extraction method outperformed others with image classification algorithms, prompting its selection for further investigation.

The HOG feature descriptor captures key points from images while discarding non-essential information, thereby reducing dimensionality. These key points are unique to each image, enabling clear differentiation between images. The feature descriptor converts images into vectors, which serve as input values for classification algorithms. Prior to descriptor calculation, images are resized to a standard aspect ratio, typically 64 × 128, facilitating feature extraction. Gradient images are computed by calculating vertical and horizontal gradients, retaining shape and edge information crucial for feature extraction. Unlike other descriptors, HOG extracts both magnitude and direction of edges, distinguishing it as the Histogram of Oriented Gradient.

Gradient calculations involve determining the change in pixel values in both x and y directions, generating matrices for each small image patch. Total Gradient Magnitude (T.G.M) is computed to obtain the magnitude of all elements in an image, while pixel direction is determined using mathematical equations. Histograms are then computed for each pixel using magnitude and direction information, with HOG features serving as input for image classification algorithms.

The subsequent sections of this paper are structured as follows: Section II presents related work on feature extraction methods and image classification algorithms. Section III details the feature extraction process, hyperparameters used with classification algorithms, and cross-validation techniques employed to mitigate overfitting. Section IV discusses experimental results and identifies the best classifier. Finally, Section V concludes and outlines future research directions.

II. RELATED WORKS

The employment of feature extraction methods in this study, serving as inputs to machine learning algorithms for maize disease image identification, is widespread. Each machine learning algorithm utilizes distinct metrics to gauge the classification of disease images. Various methods within computer vision are employed for identifying crop infections, with feature extraction from images being a prominent technique.

Pujai et al. conducted experiments utilizing Artificial Neural Networks (ANN) and Support Vector Machine (SVM) for maize disease image classification. The classifiers were trained on image features extracted via a feature extraction method, revealing SVM's superior performance over ANN. SVM achieved an accuracy of 0.9217, surpassing ANN's 0.874 accuracy.

NJICE –National Journal on Information and Communication Engineering
ISSN: 2231-2099Volume 11 Issue 1
Jan-March 2021 Pages 14-30

Yaktundimath et al. categorized three cereal plant types and employed machine learning algorithms to classify their leaf diseases. SVM and ANN were utilized to classify various maize leaf diseases based on fungal symptoms. The experiment involved acquiring and preprocessing 750 JPG images of normal and fungal-affected leaves, followed by feature extraction using the Color Co-occurrence matrix algorithm. SVM yielded a classification accuracy of 83.83%, outperforming ANN's 77.75%.

Zhang et al. focused on classifying five types of maize crop diseases using machine learning algorithms. They collected 20 images per disease category for training and testing purposes. K-Nearest Neighbors (KNN) algorithm was utilized for feature classification, achieving over 80% accuracy. The study suggested future research involving larger training datasets and key point extraction for improved classification.

Xiaoyang et al. conducted research in Chinese farm areas, classifying four types of maize leaf diseases. The images were captured under sunlight conditions, converted to BMP format, and segmented using thresholding. Classification was performed using the GA-SVM algorithm, yielding precision rates between 88.72% and 92.59%. Further studies are proposed to explore alternative algorithms and assess precision, recall, and F1-score metrics for comprehensive evaluation of maize leaf disease classification.

III. METHODOLOGY

Dataset Description The research utilized a publicly available augmented maize disease dataset sourced from Kaggle, which included images for training and testing. The dataset featured images of common rust, leaf spot, northern leaf blight diseases, and healthy maize leaves. Specifically, the training dataset comprised 7308 images—1634 for leaf spot, 1907 for common rust, 1908 for northern leaf blight, and 1859 for healthy leaves. The testing dataset contained 1826 images, distributed as 407 for leaf spot, 477 for common rust, 477 for northern leaf blight, and 465 for healthy leaves. Due to constraints in time and resources, the initial focus was on a subset of 300 training images from each disease category, totaling 1200 images, as feature generation from each image was time-intensive and resource-heavy. For testing, 30 images from each disease category were used out of the total 1826.

Numerical Feature Extraction

Feature extraction was conducted using ORB, KAZE, and HOG methods. The extracted features, represented as integers, were then inputted into various machine learning classifiers. From each image, 4096 key points were extracted, serving as crucial attributes for the classifiers. These key points are significant as they uniquely identify features within an image that remain detectable despite variations in appearance. The extracted features were evaluated through classification algorithms to assess which method enhanced machine learning performance the most. The

algorithms' effectiveness was measured by their accuracy, revealing that the HOG method outperformed the others in terms of enhancing classifier performance.

Dataset Description

The study made use of a maize disease dataset available on Kaggle, which was augmented and included both training and testing images. This dataset incorporated images depicting various conditions such as common rust, leaf spot, northern leaf blight, and healthy maize leaves. The training set included a total of 7308 images, with 1634 images for leaf spot, 1907 for common rust, 1908 for northern leaf blight, and 1859 images of healthy leaves. The testing set consisted of 1826 images, with 407 for leaf spot, 477 for common rust, 477 for northern leaf blight, and 465 for healthy leaves. Due to limited time and resources, the research initially concentrated on a selected subset of 300 images per disease category, resulting in a total of 1200 images for the preliminary training phase. The testing phase utilized 30 images from each category.

Numerical Feature Extraction

Feature extraction processes were implemented using ORB, KAZE, and HOG techniques, where the features were numerically represented and fed into various machine learning classifiers. Each image provided 4096 key points, which were pivotal for the classifiers. These key points are essential as they represent unique features of an image that are identifiable even when the image undergoes changes in appearance. The effectiveness of these feature extraction methods was tested using classification algorithms to determine which method best enhances the performance of machine learning algorithms. Among the tested methods, the HOG technique proved superior in improving the accuracy of classification algorithms.

HOG Feature Extraction Approach Image Preprocessing

For the preprocessing stage in the HOG feature extraction process, an image with original dimensions of 180 × 280 would typically be resized to a ratio of 1:2, commonly adjusted to 64 × 128. This resizing is a crucial preprocessing step as it simplifies the subsequent breakdown of the image into smaller blocks of 8 by 8 and 16 by 16 pixels, facilitating the feature generation process. The choice of a 1:2 pixel ratio aids in streamlining the computation of feature extraction. After resizing, the gradients in both the x and y directions of each pixel are calculated to ascertain the orientation and magnitude of changes across the image, using a smaller, representative pixel matrix window from the adjusted image as an example for calculations.

NJICE –National Journal on Information and Communication Engineering
ISSN: 2231-2099Volume 11 Issue 1
Jan-March 2021 Pages 14-30

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

Source: 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

In the pixel matrix, the pixel with a value of 85 is highlighted in red, which we will use to illustrate how the gradient of a pixel is computed. To calculate the change in the x-direction for the pixel with value 85, we subtract the value of the pixel to its left from the value of the pixel to its right. Similarly, the change in the y-direction for this pixel is determined by subtracting the pixel value directly below it from the pixel value directly above it. The gradients in the x and y directions for pixel value 85 are calculated as follows: Gx = 89 - 78 = 11 Gy = 68 - 56 = 8 These calculations are performed for all pixel values within the matrix, resulting in a new matrix containing these gradient values, which are crucial for computing the orientation and magnitude of each pixel.

Orientation and Magnitude

The orientation and magnitude for each pixel are calculated using the gradient values from the newly formed matrix, applying the Pythagorean theorem. This method helps to determine the direction and strength of each gradient within the image.

NJICE –National Journal on Information and Communication Engineering
ISSN: 2231-2099Volume 11 Issue 1
Jan-March 2021 Pages 14-30

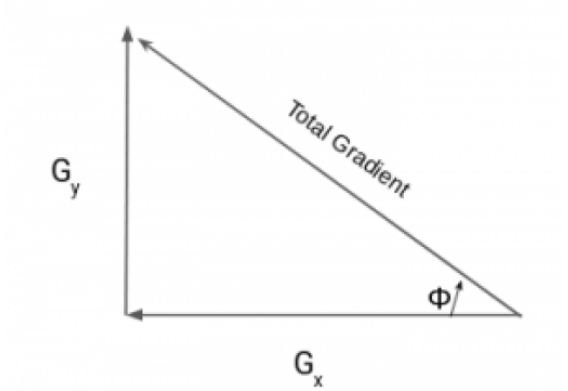


Figure 1. Orientation and Magnitude

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;

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

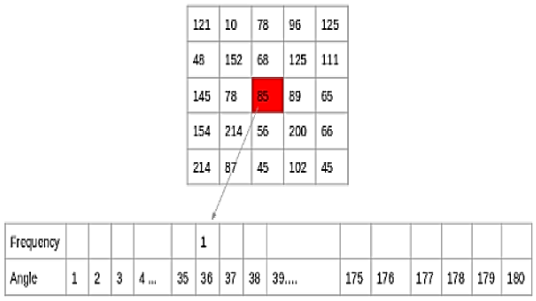
$\Theta = \arctan(G_x / G_y)$

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

$= 36$

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.



Source: Applied Machine Learning - Beginner to Professional. (n.d.). Retrieved from

NJICE –National Journal on Information and Communication Engineering
ISSN: 2231-2099Volume 11 Issue 1
Jan-March 2021 Pages 14-30

https://courses.analyticsvidhya.com/courses/applied-machine-learning-beginner-to-professional/?utm_source=blog&utm_medium=understand-math-HOG-feature-descriptor

As indicated by the pixel matrix, the direction value of pixel 85 is 36. In the frequency table, the occurrence of the value 36 is noted once. This procedure is repeated for each pixel value in the image. The y and x-axis values are derived from the frequency distribution table.

Histogram of Gradients in an 8 × 8 Image Patch The Histogram of Oriented Gradients (HOG) technique segments the image into smaller sections or patches of 8 × 8 cells, calculating the features for these patches. Typically, a 9 × 1 matrix is generated for each cell after creating the histogram from an image divided into these 8 × 8 cells. These histograms are then normalized after the HOG features are extracted from the 8 × 8 cells.

Normalize Gradients

In the context of 8 × 8 cells, normalization of gradients is essential because some parts of the image may appear brighter than others. This normalization involves using 16 × 16 blocks to reduce variations in lighting. By combining four 8 × 8 cells into one, we form a 16 × 16 block. Each 8 × 8 cell contributes a 9 × 1 matrix to the histogram, resulting in a combined 36 × 1 matrix. The normalization process involves summing the squares of the matrix values, calculating the square root of this sum, and then dividing each matrix value by this result. For a vector F defined as: F = [x1, x2, x3, ..., x36] The root of the sum of squares is calculated as: $Y = \sqrt{(x1^2 + x2^2 + x3^2 + ... + x36^2)}$ The values of vector F are then divided by Y, resulting in a normalized 36 × 1 matrix.

Complete Image Features

The comprehensive image features are obtained by amalgamating the features of the 16 × 16 blocks. For an image size of 64 × 128, the calculation shows that 7 × 15 blocks of 16 × 16 are required. Since each 16 × 16 block contains 36 × 1 feature values, the total feature count for a 64 × 128 image is 7 × 15 × 36 × 1, equating to 3780 HOG features.

Hyperparameter Tuning

Optimal hyperparameters for each classification algorithm were identified using a grid search method, which exhaustively explores a predefined set of hyperparameters. This tuning improves the performance of machine learning algorithms but is computationally intensive and time-consuming. Before implementing this method in a Jupyter notebook, a specific grid of parameters was defined to guide the search.

Cross-Validation

Cross-validation is utilized to ascertain that the classifier performs well with data it has not previously encountered. This method provides assurances that the classification algorithm

effectively predicts new data and also helps in determining if the algorithm is underfitting or overfitting. Typically, using only part of the training data for validation can lead to underfitting, as it might omit significant patterns and trends from the reduced training dataset. K-Fold cross-validation addresses this by allocating parts of the data for validation and the rest for training, alternating the roles in subsequent iterations. This approach significantly reduces variance and bias, enhancing the reliability of the model's effectiveness assessment. Each data point is included in the training set k-1 times and appears once in the testing set, ensuring comprehensive data utilization and accurate error estimation.

The steps followed in the research methodology can be outlined as follows:

1. **Randomize the Dataset:** Begin by randomly shuffling the dataset to prevent any bias that might arise due to the initial ordering of the data.
2. **Partition the Dataset into K Groups:** Divide the entire dataset into K distinct groups. This process ensures that each subset is representative of the whole dataset, facilitating a robust cross-validation process.
3. **Iterative Testing and Training:** For each distinct group:
    - Use the k-th subset as the test or validation set.
    - Combine the remaining K-1 subsets to form the training set.
    - Train the model on the K-1 subsets and evaluate it using the k-th subset.
    - Discard the model after evaluation to ensure that each iteration uses a fresh model, thereby avoiding any leakage of data between iterations.
    - Record the evaluation score for each trial.
4. **Calculate Overall Model Effectiveness:** Average the evaluation scores from all K trials to estimate the total effectiveness of the model. This average provides a measure of how well the model is expected to perform on unseen data, taking into account variations across different test sets.

Experimental Results and Discussion

Feature Extraction Analysis Figure 2 demonstrates the process of feature extraction using the ORB method on an image affected by common rust disease. This visual representation highlights how the ORB algorithm identifies and captures critical features from the image, which are essential for the successful classification of the disease type. This step is crucial as it directly influences the accuracy and performance of the subsequent classification models.

NJICE –National Journal on Information and Communication Engineering
ISSN: 2231-2099Volume 11 Issue 1
Jan-March 2021 Pages 14-30

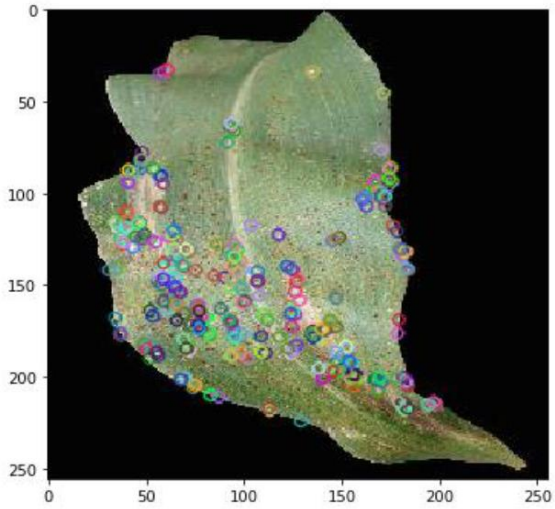


Figure 2. Detecting key points from common rust disease image using ORB method

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

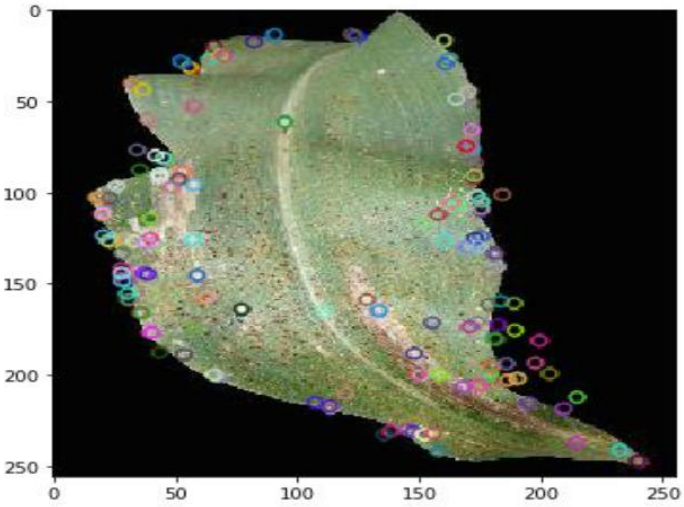


Figure 3. Detecting key points from common rust disease image using KAZE method

The dataset consisted of images categorized into common rust, leaf spot, northern leaf blight disease, and healthy leaves. Due to the intensive computational demands of feature extraction, the research team limited the number of images to 300 per disease type, extracting 4096 features from each image. The selected images were then processed using three different feature extraction methods—KAZE, ORB, and HOG—to prepare for machine learning training.

NJICE –National Journal on Information and Communication Engineering
ISSN: 2231-2099Volume 11 Issue 1
Jan-March 2021 Pages 14-30

Machine Learning Algorithms

A variety of machine learning algorithms were employed to evaluate the effectiveness of each feature extraction method. These included:

- Random Forest
- Logistic Regression
- AdaBoost
- Bagging
- Gradient Boosting
- Bernoulli Naive Bayes (NB)
- Gaussian NB
- K-Nearest Neighbors (KNN)
- Neural Network
- Linear Support Vector Classifier (SVC)
- Support Vector Machine (SVM)

Each algorithm was trained on the features derived from the aforementioned extraction methods. The main objective of the training was to link each set of features to their corresponding disease type, thereby enabling the algorithms to learn from the data provided.

Evaluation of Feature Extraction Methods

Initially, to gain preliminary insights, features were extracted from just 32 images per disease type using each of the three methods. The performance of each classifier was then assessed based on accuracy, aiming to determine which feature extraction method yielded the best results. The findings are summarized in Table 1, which presents the accuracy rates achieved by each algorithm when utilizing features extracted via KAZE, ORB, and HOG methods.

This comparative analysis helps identify which feature extraction technique is most effective when paired with various machine learning classifiers, guiding further detailed studies and optimizations in the feature extraction phase of the research.

NJICE –National Journal on Information and Communication Engineering
ISSN: 2231-2099Volume 11 Issue 1
Jan-March 2021 Pages 14-30

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

From the data presented in Table 1, it's evident that the ORB feature extraction method significantly underperformed when compared to the other methods. The HOG method stood out, yielding the highest average performance with an accuracy of 0.618, followed closely by KAZE with an accuracy of 0.608, and ORB trailing significantly at 0.354. Consequently, the decision was made to primarily utilize the HOG method for feature extraction in further experiments.

Hyperparameter Tuning

Hyperparameter tuning is a critical process in optimizing machine learning algorithms. Hyperparameters, unlike model parameters, are set before the learning process begins and directly influence the behavior of the training algorithm. Effective tuning of these parameters can dramatically enhance the performance of a model.

For classifiers that were compatible with hyperparameter tuning, the following hyperparameters were adjusted to optimize the models using HOG features:

1. **Random Forest:** Number of trees, depth of the tree, and the minimum number of samples required to split a node.
2. **Logistic Regression:** Regularization strength and type of solver.
3. **Support Vector Machine (SVM):** Kernel type (e.g., linear, polynomial, RBF), C (penalty of the error term), and gamma (kernel coefficient).
4. **Gradient Boosting:** Number of boosting stages, learning rate, and max depth of the individual regression estimators.
5. **AdaBoost:** Number of estimators and learning rate.

NJICE –National Journal on Information and Communication Engineering
ISSN: 2231-2099Volume 11 Issue 1
Jan-March 2021 Pages 14-30

6. **Neural Networks:** Number of layers, number of neurons in each layer, activation function, and learning rate.

Each of these hyperparameters plays a unique role in shaping the learning process. For example, in Random Forest, increasing the number of trees can lead to better model performance but may also increase computational cost. In SVMs, the choice of kernel and the values of C and gamma significantly affect the decision boundary's flexibility and hence the classifier's ability to generalize.

The process of hyperparameter tuning typically involves trials of different combinations of parameters to find the most effective settings. Common strategies for this include grid search, where a predefined grid of hyperparameters is exhaustively searched, and random search, which randomly samples parameter combinations from a defined range.

Ultimately, the selected hyperparameters are those that yield the best performance on a validation set or through cross-validation, ensuring that the model is neither underfitting nor overfitting. The goal is to achieve a well-tuned model that can accurately generalize from the training data to unseen data, significantly enhancing the robustness and reliability of predictive analytics.

RandomForestClassifier(bootstrap=False, ccp_alpha=0.0, class_weight=None, criterion='entropy', max_depth=None, max_features=100, 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=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='ovo', 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 is an essential tool for assessing the performance of classification algorithms. It provides key metrics that reflect the quality of predictions made by the algorithms.

NJICE –National Journal on Information and Communication Engineering
ISSN: 2231-2099Volume 11 Issue 1
Jan-March 2021 Pages 14-30

These metrics usually include precision, recall, f1-score, and support for each class, allowing you to understand not just overall effectiveness but also how well each class is being predicted.

For this study, the classification report was generated for a subset of the algorithms due to the computational time required to produce these reports. The testing involved a balanced dataset with 30 images from each disease category within the testing set.

Figures 5, 6, 7, and 8 in your report presumably illustrate the results of this testing phase for three different classifiers. These figures would typically display the classification metrics for each classifier to highlight their individual performance on the test data, often presented in a tabular format where rows represent the true classes and columns contain the performance metrics for those classes.

To assess the classifiers' ability to distinguish between the different categories of disease, and to identify healthy leaves, the metrics in the classification report are crucial. Higher precision indicates that the classifier is good at not labeling negative samples as positive, while higher recall shows that the classifier is good at finding all positive samples. The f1-score provides a balance between precision and recall, giving a single score that accounts for both false positives and false negatives. The support indicates the number of true instances for each label, which helps to understand the distribution of the dataset.

	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

NJICE –National Journal on Information and Communication Engineering
ISSN: 2231-2099Volume 11 Issue 1
Jan-March 2021 Pages 14-30

	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

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

The classification reports depicted in Figures 5 through 8 provide a clear view of how each classifier performed when tasked with categorizing 30 images per image category, as indicated in the 'support' column of each report.

NJICE –National Journal on Information and Communication Engineering
ISSN: 2231-2099Volume 11 Issue 1
Jan-March 2021 Pages 14-30

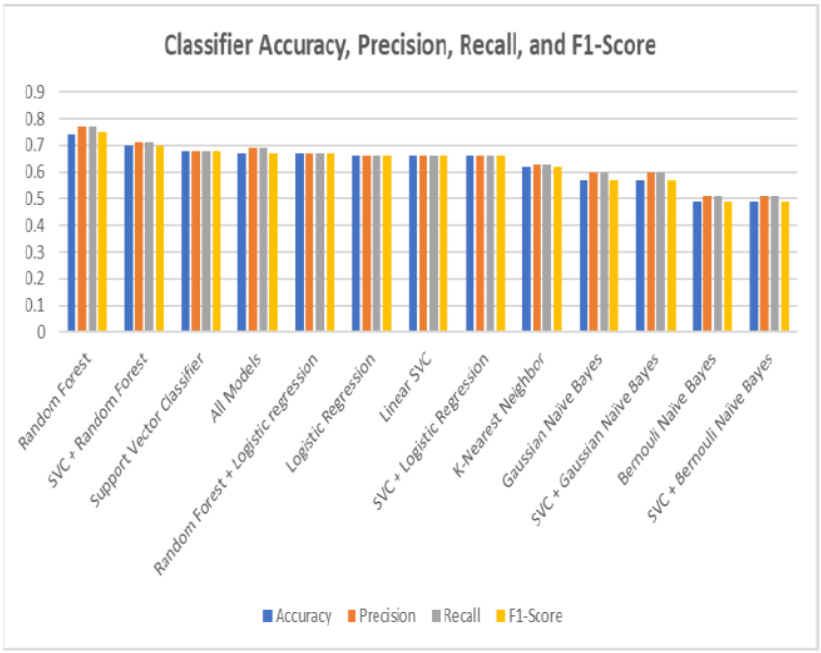


Figure 9. Overall Classification Metrics

Overall Classification Metrics Upon analyzing the classification report for each model, we were able to gauge the overall effectiveness of the models. This comparative analysis revealed the models' capabilities in accurately classifying images from the test dataset. Specifically, the outcomes presented in Figure 9 highlighted that the Random Forest model was the most proficient with the test data, whereas the Bernoulli Naive Bayes and a hybrid model incorporating both the Support Vector Classifier and Bernoulli Naive Bayes were the least effective.

IV. CONCLUSION AND FUTURE WORK

The article presents a technological approach to aid farmers in diagnosing maize diseases by employing machine learning algorithms. The research aimed to ascertain the most effective feature extraction technique for enhancing the performance of machine learning classifiers. The findings indicated that the Histogram of Oriented Gradients (HOG) method surpassed other methods such as KAZE and ORB in effectiveness, leading to its selection for ongoing research. Furthermore, an evaluation of various machine learning algorithms revealed that the Random Forest algorithm outperformed its counterparts, delivering an accuracy of 0.74, precision of 0.77, recall of 0.77, and an F1-score of 0.75. The classification reports corroborated the superior performance of the Random Forest classifier. In summary, the study recommends employing the HOG method for feature extraction in conjunction with the Random Forest algorithm to achieve more accurate identification of maize diseases, as this combination yielded the most favorable results.

NJICE –National Journal on Information and Communication Engineering
ISSN: 2231-2099Volume 11 Issue 1
Jan-March 2021 Pages 14-30

V. REFERENCES

1. Amato, G., & Falchi, F. (2017). 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). 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). KAZE-BOF-based large vehicles detection at night. In *2016 International Conference on Communication Problem-Solving* (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). 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). 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). SVM binary classifier ensembles for image classification. In *Proceedings of the tenth international conference on Information and knowledge management* (pp. 395-402).
9. Kim, J. I., 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. Song, K., Liu, Z., et al. (2011). 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). 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). 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. Jagadeesh, P. D., Rajesh, Y., et al. (2013). Classification of fungal disease symptoms affected on cereals using color texture features. *International Journal of Signal Processing, Image Processing and Pattern Recognition, 6*, 321-330.
16. Jagadeesh, P., Yakkundimath, R., et al. (2016). SVM and ANN based classification of plant diseases using feature reduction technique. *International Journal of Interactive Multimedia and Artificial Intelligence, 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). 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). 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). An improved random forest classifier for image classification. In *2012 IEEE International Conference on Information and Automation* (pp. 795-800). IEEE.
23. Zhiyong, Z., Xiaoyang, H., et al. (2015). Image recognition of maize leaf disease based on GA-SVM. *Chemical Engineering Transactions, 46*, 199-204.