BANANA AND CASSAVA CROP CLASSIFICATION AND DETECTION

Brenda Adokorach
Department of Computer Science
Makerere University
Kampala, Uganda
2021/HD05/2282U
2100702282
brendaadokorach@gmail.com

Abstract

In the plant leaf classification, leaf representation with traditionally handcrafted features is difficult to reveal its complex shape and texture. In this paper, we proposed a novel plant leaf classification method based on Yolov5 convolutional neural network (CNN) due to its powerful capability of feature extraction and classification and a K-Nearest Neighbor(KNN) machine learning algorithm to classify whether the plant leaf is for a banana or cassava. In our method, we propose two methods for image classification and object detection. The image classification is implemented using machine learning model K-Nearest Neighbour while Yolov5 for object detection. The models were trained and tested. The accuracy for K-NN is 99.4% while a mean average precision of 95% was obtained for Yolov5.

1. Introduction

Crop classification using a leaf is one of the method used by very people to identify or classify what species or type of crop it belongs to. The shape, color or texture of the leaf is one the features normally used. This traditional approach is inaccurate as some of the crops present similar features .For example, Banana leaves are large, wide, elongated, and little bit rounded. The surface of the Banana leaves are waxy, flexible, and glossy, and range in color from lime, olive green, to dark green while the cassava leaves are palm-like. Hence, leaf is one of the main reference points for leaf classification.

Generally, the extraction of leaf features, such as traditional shape , texture and color feature [1][2], is important for leaf classification. Based on the extracted feature, machine learning or deep learning model is used to classify [3][4].

Image-based methods for classification is considered as the best method as Artificial intelligence(AI) is gaining grounds in this field. A user can take a picture of a plant in the field with the build-in camera of a mobile device and analyze it with an installed recognition application to identify the plant to which the leaf belongs. By using these applications, also non-professionals can take part in this process. An image classification process is divided into the steps in Figure 1. Image acquisition: The purpose of this step is to obtain the image of a whole plant or its organs so that analysis towards classification can be performed. The aim of image preprocessing is enhancing image data so that undesired distortions are suppressed and

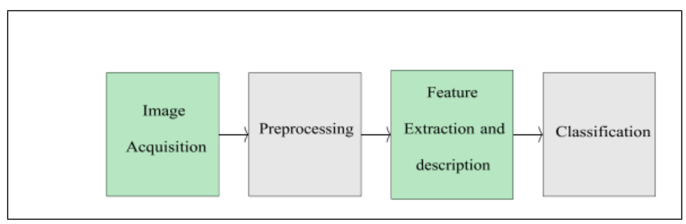

Figure 1. Image Processing steps

features that are relevant for further processing are emphasized. The preprocessing generates a modified image as output, for the next step, the feature extraction. Preprocessing includes operations like image denoising, image content enhancement, and segmentation. These can be applied in parallel or individually, and they may be performed several times until the quality of the image is satisfactory. Feature extraction and description: Feature extraction refers to taking measurements, geometric or segmented, meaningful regions in the image. Features are described by a set of numbers that characterize some property of the plant. Classification: In this step, all extracted features are concatenated into a feature vector, which is then being classified.

The rest of the paper is organized as follows: Section 2 discusses related works using traditional machine learning and deep learning, Section 3 presents our proposed approach, Section 4 shows the result and discussion, followed by the conclusion in Section 5.

2. Literature Review

In most recent literature, many authors have used Artificial intelligence in classifying whether a leaf has disease or not. Several deep learning models like CNN, VGG16, VGG19, Inception, AlexNet, GoogleNet, EfficientNet B4, Yolo and machine learning has been used by many authors.

According to authors in [5], they proposed a CNN model for distinguish 32 plant spices. The Flavia leaf database used in this work contains 1,600 images of 32 species. Each kind of leaf contains 50 images. Each leaf image is a white background RGB image and the size of the image is 1600×1200 pixels. The ten-layer CNN achieved an accuracy of 87.92%. Traditional method SVM was also used but results were poor.

A few-shot learning method based on the Siamese network framework is proposed in [6] to solve a leaf classification problem with a small sample size. First, the features of two different images are extracted by a parallel two-way convolutional neural network with weight sharing. Then, the network used a loss function to learn the metric space, in which similar leaf samples are close to each other and different leaf samples are far away from each other. In addition, a spatial structure optimizer (SSO) method was used to improve the accuracy of leaf classification. Finally, a k-nearest neighbor (kNN) classifier is used to classify leaves in the learned metric space. The open Flavia dataset, Swedish dataset and Leafsnap dataset were used to evaluate the performance of the method. The classification accuracies are 95.32%, 91.37% and 91.75% for the Flavia, Swedish and Leafsnap datasets, respectively.

The authors in [7] used leaf dataset with 30 different plant species and classification is performed using Multilayer Perceptron, Naive Bayes and Support Vector classifiers. Time taken to build and classify the leaves was used to measure and compare performance of classifiers. The dataset used contains 15 features. The models achieved the accuracies of 79.7%, 53.2% and 74.1% for Multilayer, SVM and Naïve Bayes respectively.

A study in [8] proposed two methods for the problem of plant species identification from leaf patterns. Firstly, they used a traditional recognition shallow architecture with extracted features histogram of oriented gradients (HOG) vector, and then used the features to classify by SVM algorithm. Secondly, a deep convolutional neural network (CNN) for recognition purpose. Experiments were done on leaves data set in the Flavia leaf data set and the Swedish leaf data set.

Three features were combined to classify plant species from a leaf dataset in [9]. The three features are shape, texture and margin features. The features are combined using a probabilistic framework. The texture and margin features use histogram accumulation, while a normalised description of contour is used for the shape.

The researchers in [10] applied naïve Bayesian method in plant classification based on leaf shape and textures as input features in order to produce model classifier that is accurate and efficient. The dataset with 30 different plant species was used to construct and train the classifier.

A Vehicle Classification Application on Video Using Yolov5 Architecture was developed by authors in [12]. A sample of 750 images and 11 vehicle types were used in the dataset. Image markup was performed in the web tool COCO Annotator. The training and validation data was split using the 80:20 ratios. Training was run for 100 epochs. The performance was measured using Mean Average Precision(Map), recall and precision.

In [13], a deep learning convolution neural network was used to classify food-11 dataset and they got an accuracy of 92.86%. Transfer learning was done with inception v3 model pre-trained with ImageNet.

A pre-trained image classification models for classifying weed species and also for locating and identifying weed species was developed in [14]. The image classification models were trained on two commonly used deep learning frameworks i.e., Keras and PyTorch. An annotated dataset comprising of RGB images of four, early season weeds, found in corn and soybean production system in the Midwest US, namely, cocklebur (Xanthium strumarium), foxtail (Setaria viridis), redroot pigweed (Amaranthus retroflexus), and giant ragweed (Ambrosia trifida) was used in this study. VGG16, ResNet50, and InceptionV3 pre-trained models were used for image classification. The object detection model, based on the You Only Look Once (YOLOv3) library, was trained to locate and identify different weed species within an image. The performance of image classification models was assessed using testing accuracy and F1-score metrics. Average precision (AP) and mean average precision (mAP) were used to assess the performance of the object detection model.

In this paper, we propose two methods that uses machine learning Naïve Bayes classifier and a deep learning Yolov5 pre-trained model to classify and detect banana and cassava leaves.

3. Methodology

In this paper, both machine learning and deep learning model was used to detect and classify crops using their leaves. Image processing was done before the dataset was used to train and test both models. For machine learning KNN classifier was used. Transfer learning was used by a pre-trained Yolov5 model. The model was then trained our crop leaf dataset provided. For KNN, the images were cropped in order to separate the cassava from banana leaves.

3.1. Dataset

A dataset consisting of 583 annotated images were used to train the Yolov5 neural network. The data was divided into test, train and validation using 80:20 ratios. The annotation was done from an open source tool Roboflow and later the dataset is imported into the development environment google colab using an API call. In image classification, the images were first cropped in order to separate cassava leaves from banana leaves. The dataset formed contained 422 images with 211 images for each class.

3.2. Image Classification

Since in machine learning, feature extraction is not automatic as compared to deep learning, region properties were used to extract features from the leaves and image classification using KNN. The features extracted are iqr,75th Percentile, inertia_tensor11, std_intensity, mean_intensity,25th Percentile, minor_axis_length, solidity, eccentricity. Ee also evaluated the performance of the KNN classifier using a combination of these features.pixel values, color histogram and Oriented Fast and Rotated Brief (ORB) and features.The images were first converted to gray scale before feature extraction except for color histogram where the processing is done on colored images.

Below are the steps followed.

- Crop the images in order to get banana and cassava plants
- Examine the cropped the images to identify features that can distinguish them
- Develop a feature extraction algorithm
- Extract the features and store them in a panda data frame and assign label to each feature
- Split the dataset into train and test using the ratio of 80:20
- Develop the KNN classifier for binary classification
- Train the model
- Test the model using the test dataset.
- Evaluate model using the following metrics: accuracy, precision, recall and f1-score, confusion matrix.

3.3. Object Detection

Yolov5 deep learning neural network was used for object detection in this paper. The model consists of 157 layers and 7015519 parameters. The code was implemented using

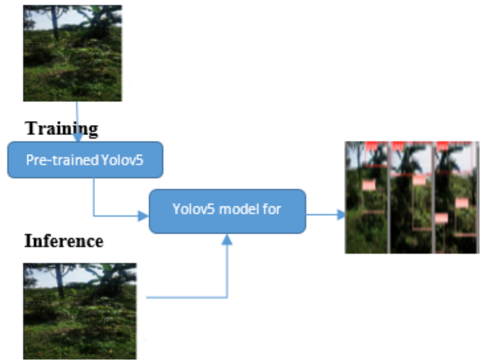

Figure 2. Yolov5 model flow

google colab. The methodology for objection using Yolov5 is shown in fig 2.

The neural network was trained on the Google Colaboratory cloud platform. The entire dataset had 583 images. The training sample was 80% of the total sample, which was 483 images. The images were resized to 415x415 and training was performed for 40 epochs. The training lasted for 0.061 hours. Validation was done with a sample of 116 images. The script train.py script was ran to train the neural network. When training a neural network, the script divides the dataset into a hierarchy of folders. These folders are: images and labels. The images folder should store the images from the dataset, and the labels folder should contain text files with annotations for each image. In each of these folders, the train, val and test folders are created, where train and validation which the files are placed. The markup text file contains information about one object per line. The information contained are the object's class number, coordinates of the object's center and its width, and height. The coordinates of the center, width and height are normalized from 0 to 1. An example of a markup file is shown in Fig. 3.

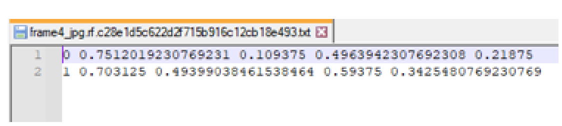

Figure 3. Example of a markup file

Yolov5 Architecture

YOLOv5 uses convolutional layers, which makes it a fully convolutional neural network. The architecture of YOLOv5 is divided into three main parts: feature extractor, detector and classifier. When a new image enters the network, it passes through the feature extractor to extract feature maps of different scales. The feature maps are fed into the detector to obtain information about the location of the bounding boxes and the class of the found object is determined by the classifier. This process is described in Fig. 4.

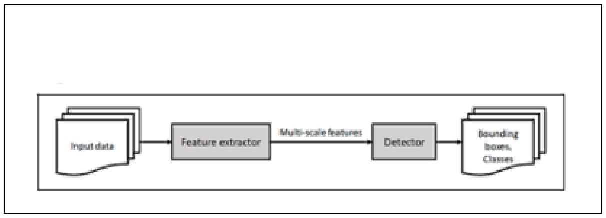

Figure 4. Yolo Architecture

The YOLOv5 architecture uses CSPDarknet, as the feature extractor and PANet. as the detector. A detailed diagram of the YOLOv5 network architecture is shown in Fig. 5. There are 4 subspecies of YOLOv5 topologies: YOLOv5s,YOLOv5m, YOLOv5l and YOLOv5x. They are distinguished from each other by the number of convolutional layers and subsamples in the CSP layer blocks of the feature extractor. In this regard, these models differ in the size of the occupied disk space and the quality of recognition.

3.4. Evaluation Metrics

Results from machine learning and deep learning technique used in this study were assessed using different pairs of evaluation metrics. Classification models were evaluated using testing accuracy, confusion metrics and F1-score metrics. Accuracy is one of the most widely used metrics for evaluating the performance of deep learning models.

For Object detection, precision, recall, mean average precision(mAP) and also confusion metrics were used.
Confusion matrix shows the actual and predicted labels from a classification problem
Precision quantifies the number of positive class predictions that actually belong to the positive class.
Recall quantifies the number of positive class predictions made out of all positive examples in the dataset

Precision = TP/(TP + FP)

Recall = TP / (TP + FN)

Accuracy= TP/Total

F1-score = 2*Precision*Recall/(Precision + Recall)

Where Total = (TP + TN+ FP + FN)

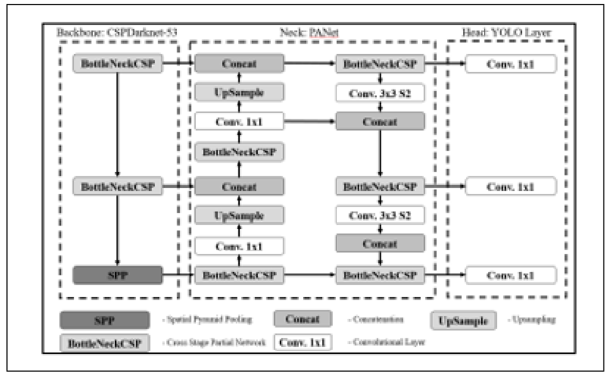

Figure 5. Yolov5 detailed Architecture

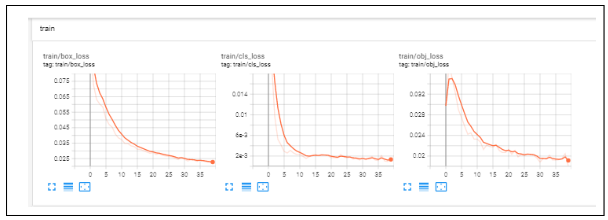

Figure 6. Yolov5 Training loss

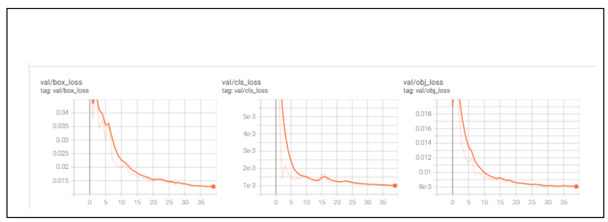

Figure 7. Yolov5 Validation loss

TP (True Positive) and TN (True Negative) are the number of truly predicted samples of positive and negative class, respectively, FP (False Positive) means the number of negative class samples that are predicted positive, and FN (False Negative) means the number of positive class samples that are predicted negative.

4. Results and Discussions

4.1. Object Detection

After training, loss functions were measured for the training and validation samples of the dataset. The loss functions were measured to determine the class, object, and bounding boxes. Graphs of the loss functions are shown in Fig. 6 and Fig. 7

From the precision recall curve, the model obtained a high area under the curve which represents both high recall and high precision. This implies our Yolov5 model has a low false positive rate and a low false negative rate. The

Figure 8. Yolov5 Validation Samples test result for object detection using Yolov5

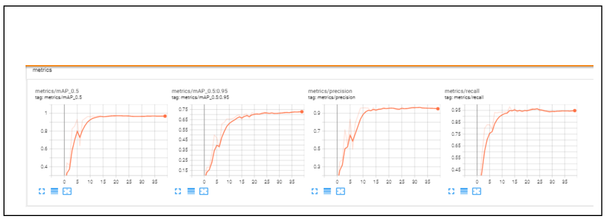

Figure 9. Precision, Recall and Mean Average Precision

mean average precision of 95% was obtained. The model was tested with a set of images, banana and cassava leaves were correctly detected using the trained Yolov5 model as shown in fig. 8.

The graph showing the precision, recall and mean average precision is shown in fig 9.

4.2. Image Classification

Two methods were used for image classification. Method 1 uses a combination of three features (image pixels, color histogram and org) while method 2 uses features extracted using region properties.

Results obtained from image classification model KNN showed classification method1 features performs better that using The accuracies are 97.% and 99.4% for the for method 1 and 2 respectively. The classification report showing performance in terms of accuracy, precision and recall and F1-score are shown in fig 10 and 11. A plot of training scores against number of neighbors showed that high accuracies are obtained for values between 2 and 5. The accuarcy decreases with an increase in the number of neighbors.

From the confusion matrix, its shows that 2 banana leaves were misclassified as cassava and all cassava leaves were correctly classified in method 1 as compared with method 2 where 44 banana and 87 cassava were misclassified as cassava and banana respectively. The misclassification of 87 cassava is due to class imbalance.

	precision	recall	f1-score	support
0	0.99	0.97	0.98	13355
1	0.89	0.96	0.92	3535
accuracy			0.97	16890
macro avg	0.94	0.97	0.95	16890
weighted avg	0.97	0.97	0.97	16890

Figure 10. Classification Report for method 2

	precision	recall	f1-score	support
0	0.95	1.00	0.97	38
1	1.00	0.96	0.98	47
accuracy			0.98	85
macro avg	0.97	0.98	0.98	85
weighted avg	0.98	0.98	0.98	85

Figure 11. Classification Report for method 1

5. Conclusion

In this paper, we proposed and implemented two models for image classification and object detection. the models uses Machine learning and deep learning. KNN was used for image classification and from the results obtained, an accuracy of 99.4% showed that the model was able to classify whether a plant for banana or cassava. In object detection, A Yolov5 model was implemented and mAP of 0.95 was obtained. The model was tested and both banana and cassava plants were correctly detected.

References

[1] H. Kebapci, B. Yanikoglu and G. Unal, ”Plant Image Retrieval Using Color, Shape and Texture Features”, The Computer Journal, vol. 54, no. 9, pp. 1475-1490, 2010. Available: 10.1093/comjnl/bxq037 [Accessed 30 September 2022].

[2] Kadir, Abdul, et al. ”Leaf Classification Using Shape, Color, and Texture Features.” International Journal of Computer Trends Technology 1.3(2011):225-230.

[3] Hong Fu, Zheru Chi, Dagan Feng and Jiatao Song, ”Machine learning techniques for ontology-based leaf classification”, ICARCV 2004 8th Control, Automation, Robotics and Vision Conference, 2004.. Available: 10.1109/icarcv.2004.1468909 [Accessed 30 September 2022].

[4] Q. Man, C. Zheng, X. Wang and F. Lin, ”Recognition of Plant Leaves Using Support Vector Machine”, Communications in Computer and Information Science, pp. 192-199, 2008. Available: 10.1007/978-3-540-85930-726 [Accessed 30 September 2022].

[5] J. Liu, S. Yang, Y. Cheng and Z. Song, ”Plant Leaf Classification Based on Deep Learning,” 2018 Chinese Automation Congress (CAC), 2018, pp. 3165-3169, doi: 10.1109/CAC.2018.8623427.

[6] B. Wang and D. Wang, ”Plant Leaves Classification: A Few-Shot Learning Method Based on Siamese Network,” in IEEE Access, vol. 7, pp. 151754-151763, 2019, doi: 10.1109/ACCESS.2019.2947510.

[7] A. Music and S. Gagula-Palalic, ”Classification of Leaf Type Using Multilayer Perceptron, Naive Bayes and Support Vector Machine Classifiers”, Southeast Europe Journal of Soft Computing, vol. 5, no. 2, 2016. Available: 10.21533/scjournal.v5i2.119 [Accessed 30 September 2022].

[8] Quoc Bao, T., Tan Kiet, N., Quoc Dinh, T. and Hiep, H., 2019. Plant species identification from leaf patterns using histogram of oriented gradients feature space and convolution neural networks. Journal of Information and Telecommunication, 4(2), pp.140-150.

[9] C. Mallah, J. Cope and J. Orwell, ”Plant Leaf Classification using Probabilistic Integration of Shape, Texture and Margin Features”, Computer Graphics and Imaging / 798: Signal Processing, Pattern Recognition and Applications, 2013. Available: 10.2316/p.2013.798-098 [Accessed 30 September 2022].

[10] F. R. F. Padao and E. A. Maravillas, ”Using Naïve Bayesian method for plant leaf classification based on shape and texture features,” 2015 International Conference on Humanoid, Nanotechnology, Information Technology,Communication and Control, Environment and Management (HNICEM), 2015, pp. 1-5, doi: 10.1109/HNICEM.2015.7393179.

[11] S. V. Militante, B. D. Gerardo and N. V. Dionisio, ”Plant Leaf Detection and Disease Recognition using Deep Learning,” 2019 IEEE Eurasia Conference on IOT, Communication and Engineering (ECICE), 2019, pp. 579-582, doi: 10.1109/ECICE47484.2019.8942686

[12] D. Snegireva and G. Kataev, ”Vehicle Classification Application on Video Using Yolov5 Architecture,” 2021 International Russian Automation Conference (RusAutoCon), 2021, pp. 1008-1013, doi: 10.1109/RusAutoCon52004.2021.9537439

[13] M. T. Islam, B. M. N. Karim Siddique, S. Rahman and T. Jabid, ”Image Recognition with Deep Learning,” 2018 International Conference on Intelligent Informatics and Biomedical Sciences (ICIIBMS), 2018, pp. 106110, doi: 10.1109/ICIIBMS.2018.8550021.

[14] A. Ahmad, D. Saraswat, V. Aggarwal, A. Etienne and B. Hancock, ”Performance of deep learning models for classifying and detecting common weeds in corn and soybean production systems”, Computers and Electronics in Agriculture, vol. 184, p. 106081, 2021. Available: 10.1016/j.compag.2021.106081 [Accessed 6 October 2022].