A Report on Image classification and Object detection of Banana and cassava images

Gloria Ainebyona
Student No: 2100702283
Registration No: 2021/HD05/2283U

Abstract

Image classification is a pattern recognition technique in computer vision which uses contextual information to classify images.
Object detection is a technique in computer vision and image processing that deals with detecting instances of semantic objects of a certain class in images and videos.
Image classification and object detection utilise machine learning to perform in various applications like face detection, remote sensing of crop gardens among others.
In this project, I performed cropping of cassava and banana plants from images that were taken from a garden with a lot of clutter. I then used SIFT, ORB and color histograms to extract features from the cropped images. The extracted features were then used for image classification using traditional machine learning algorithm of K-Nearest Neighbor (KNN).
I also carried out an object detection task where I first annotated images which I then used on a pre-trained You Only Look Once (YOLOv5) model to detect images .
Comparing the performance, the YOLOv5 model had overall average Mean Average Precision (mAP) of 0.983, Precision of 0.948, Recall of 0.959 while the KNN model had accuracy of 0.93 with the fair precision, recall and F1 score as shown in the classification report in figure 7.
The YOLOv5 model proved to be better than KNN because it was very fast in terms of computation and did not require a lot of resources. Its pre-training on the coco dataset with about 80 classes and about 200,000 features contributes a lot to its efficiency.

Keywords: Image classification, Object detection,Machine learning, Mean average precision, Computer vision,Features.

1. Introduction

Machine learning and Computer Vision are applications of artificial intelligence which are tremendously being used in various applications of Image classification, object recognition and detection, image analysis, action recognition among others.
The main aim of any image classification based models is to assign labels to captured images and consequently these labels can be used to arrange images in a semantic order. These are then eventually arranged in various domains of digital image processing and computer vision such as image retrieval, object recognition, object detection, image annotation, scene analysis and video analysis.

Plant classification is one of the most experimented applications of image classification and object detection. This is because of the various challenges in agriculture which affect farmers and generally the agricultural production that contributes to economy's GDP. These challenges include; pests, weeds, bad weather which when not managed in time lead to reduction in crop yields.
This has therefore intrigued many researchers into coming up with solutions that help to combat these challenges.
Machine learning and deep learning algorithms are the state of the art solutions for most of challenges as evidenced in the contributions made in sectors like health in terms of medical diagnosis. I agriculture, these algorithms support in the extraction of features from the crop parts where leaves are the main focus for identification of novel features using the shape, texture, color and veins [1]. Remote sensing technology has been also been boosted with machine learning algorithms to assist in crop growth monitoring, soil moisture estimation among others.This technology uses less human effort and provides timely information for analysis of crop health [3].

2. Related literature

Plant classification from a garden is quite cumbersome due to a lot of challenges like; deformed plant parts like deformed leaves due to pests. Some plants have parts that have similar features with other plants [4]. While some plants get covered by other plants which brings about occlusion.

Textual content of the images can potentially provide

better results than the other features.
S. Barburiceanu, R. Terebes, and S. Meza [5] proposed a texture feature extraction technique through use of feature vectors and occurrence matrix based methods.
The features that are mostly used in the classification of the plants are leaves as these provide more details for a specific plant.
P. P. Kaur and S. Singh [6] propose an automated system for calculating leaf features like length of the leaf, width, height among others and they use KNN and support vector machine (SVM) on model training.
P. Siva Prasad and A. Senthilrajan [1] proposed a deep convolutional neural network to detect leaf images. They the used an Adaboost algorithm to collect classifiers from softmax.
M. D. Fauzi et al. [2] used a DenseNet201 to classify weed and potato plants. The main aim of this model was to help in detection of the potato plants from weeds during pesticide spraying. Since most plants get affected by the pesticides as the farmer is not able to determine which plants are potatoes and which ones are weeds.
Several feature extraction techniques are used to extract features for image classification due to advance in image processing provided by various pre-processing techniques to make images suitable for feature extraction. This involves finding discriminating features which serve as the basis for classification using several machine learning techniques like KNN, Naive Bayes, Support Vector Machines (SVM) among others.
J. A. Villaruz [8] explores use of a pre-trained ALexNet to extract features and then later applied a multi-class SVM to classify images of the of the seedlings of the three most important berry trees belonging to the Philippine indigenous plants.
In response to the difficulty in differentiation of various plants from each other, various object detection techniques have been developed to assist in detecting the different plants and classifying them in their respective categories.In addition object detection supports in reducing the adverse effects of diseases that in turn could affect the crop yields by classifying the health plants from unhealthy ones.
Various machine learning algorithms like YOLOv4 algorithm [10] have been utilised in combating some of these challenges through early detection. J.Wang et al [9] proposed three plant leaf disease detection methods called squeeze-and-excitation SSD (Se-SSD), deep block SSD (DB-SSD) and deep block attention SSD (DBA-SSD) for disease identification and disease degree classification of various plant leaves. From their experiments,DBA-SSD performed better than the rest of the methods. This was then further improved through utilisation of the convolutional layers that were previously trained on the Image Net dataset by the VGG model.

3. Methodology

This section entails the procedures that I followed in performing Image Classification and Object detection tasks.

3.1. Image classification

An image can be defined as a two-dimensional function, f (x,y) where x and y are spatial coordinates, and f is the amplitude at any pair of coordinates (x,y).
In this task, I was to build a machine learning model using either KNN or Naive Bayes, which could classify whether a certain image was a cassava or banana plant.
For the successful implementation, I undertook steps like image cropping, image processing, feature detection, feature extraction and so on.

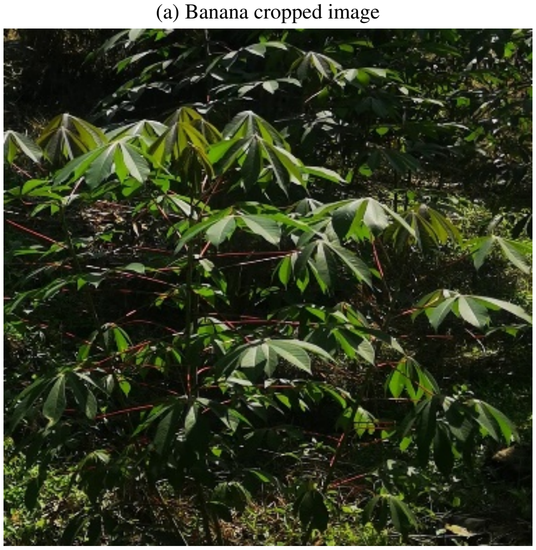
3.1.1 Image processing

During this step, I performed image cropping, where I used the Microsoft office Picture manager to manually crop out banana and cassava plants from the provided images. The cropped out images were 765 cassava images and 444 banana images.
There was a class imbalance because some images that were provided did not contain any banana plant except had only cassava plants.
The cropped out images were then resized to a size of 500*500 for uniformity.

3.1.2 Feature extraction

During this step, I performed feature detection first by use of key points on the cropped images. With the key points, unique features are detected for every image.
I then experimented on the use of colour histograms, SIFT and ORB for the extraction of features to be used for classification.
I also experimented with the key points using SIFT and ORB in the images.
The colour histograms had variations and so were not consistent thus could not rely on them for purpose of classification.
I also used the BF matcher object to find matched features of the images and then computed the sum of the distances of both cassava and banana matches.
I converted the resized images to gray scale and then applied SIFT and ORB for feature extraction.
For image classification, ORB features were preferred since ORB is well known for being rotation invariant and resistant to noise. ORB also executes faster than SIFT.
The ORB features were then applied to the KNN algorithm for classification. The results are visualised in Figure 7 where 0 denotes cassava and 1 denotes banana.

(a) Banana cropped image



(b) Cassava cropped image.

Figure 1. Cropped banana and cassava images

```
# Plot histogram for banana sample
sample = random.sample(ban_imgs_resized, k = 1)[0]
# sample = ban_imgs[34]
print(sample.shape)
plot_hist(sample)
```

```
(200, 200, 3)
```

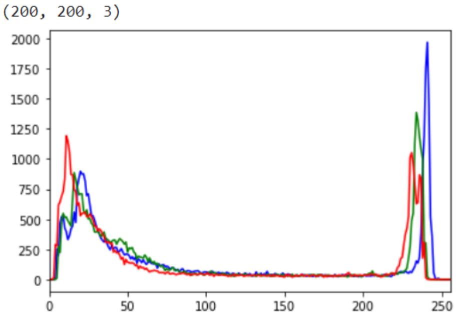


Figure 2. Color histogram for a banana plant

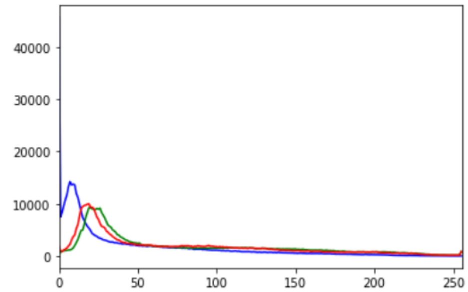
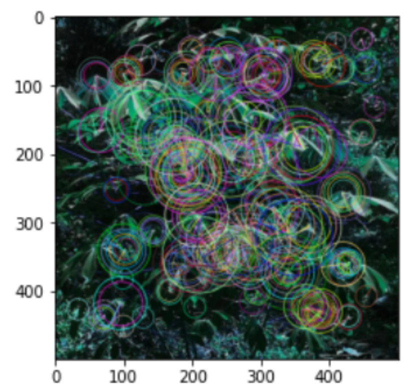
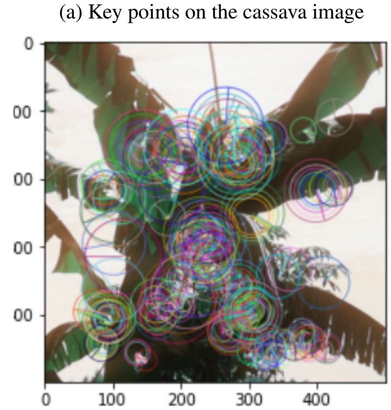


Figure 3. Color histogram for a Cassava plant



(a) Key points on the cassava image



(b) Key points on the banana image

Figure 4. Extraction of key points on images

```
ban_images_orb

[array([ 92, 184, 191, ..., 121,  80,  42], dtype=uint8),
 array([201, 180, 188, ..., 243,  70,  33], dtype=uint8),
 array([169, 160, 189, ..., 243,   6,  51], dtype=uint8),
 array([108, 169, 161, ..., 125, 255, 199], dtype=uint8),
 array([205, 118, 250, ..., 223, 218, 245], dtype=uint8),
 array([255, 120, 238, ..., 243, 171,  17], dtype=uint8),
 array([248, 109,  19, ..., 179,  70, 114], dtype=uint8),
 array([ 84,  92, 221, ..., 191,  94, 245], dtype=uint8),
 array([ 38, 165,  33, ..., 204, 181, 249], dtype=uint8),
 array([ 77, 117,  17, ..., 243, 134, 120], dtype=uint8),
 array([100, 120, 211, ..., 243,  70, 114], dtype=uint8),
 array([111, 113, 242, ..., 178, 226,  57], dtype=uint8),
 array([221, 188, 222, ..., 243, 134, 120], dtype=uint8),
 array([197,  92, 215, ..., 121,  80,  42], dtype=uint8),
 array([116,  99, 248, ..., 125, 255, 199], dtype=uint8),
 array([244, 253, 185, ..., 243, 198, 112], dtype=uint8),
 array([ 81,  10, 158, ..., 243, 103, 161], dtype=uint8),
 array([ 95,  28,  23, ..., 251,  80,  34], dtype=uint8),
 array([ 95,  28,  23, ..., 250, 198, 115], dtype=uint8),
 array([237, 253,  57, ..., 121,  80,  42], dtype=uint8),
 array([ 24, 217, 188, ..., 243, 134, 114], dtype=uint8),
 array([248, 250, 234, ..., 243, 198, 114], dtype=uint8),
 array([ 15, 189, 229, ..., 243, 103, 161], dtype=uint8),
```

(a) Banana ORB features

```
cass_images_orb

[array([248,  34, 250, ..., 241, 136,  49], dtype=uint8),
 array([ 84, 236, 156, ..., 128, 131,  81], dtype=uint8),
 array([224,  80,  89, ..., 180,  11,  93], dtype=uint8),
 array([221, 121, 247, ..., 160,   7, 185], dtype=uint8),
 array([213,  72, 130, ..., 128, 131,  81], dtype=uint8),
 array([ 82, 177, 191, ...,  96, 238, 131], dtype=uint8),
 array([ 61, 181,  29, ..., 129, 203,  35], dtype=uint8),
 array([163, 172, 135, ..., 224,   7, 185], dtype=uint8),
 array([ 41,  92, 113, ..., 129, 219,  35], dtype=uint8),
 array([ 69,  17,  21, ...,  68, 143,  95], dtype=uint8),
 array([  7,  63, 107, ...,  64, 238, 131], dtype=uint8),
 array([233,  16, 213, ..., 211,  17, 120], dtype=uint8),
 array([248, 161,  87, ...,  96, 238, 131], dtype=uint8),
 array([209,  75,  10, ..., 102,  43,  23], dtype=uint8),
 array([ 96,  32, 149, ..., 160,   7, 185], dtype=uint8),
 array([255,  62, 238, ...,  76,  84, 127], dtype=uint8),
 array([224, 160,  57, ...,  59,  86, 115], dtype=uint8),
 array([ 69,  17,  29, ...,  96, 238, 131], dtype=uint8),
 array([119, 126, 129, ..., 133,  14, 117], dtype=uint8),
 array([152, 232,  66, ..., 243,  75,   4], dtype=uint8),
 array([120, 245,  61, ..., 251, 244,  32], dtype=uint8),
```

(b) Cassava ORB features.

Figure 5. Extracted features

```
cass_images_sift

[array([1., 0., 1., ..., 0., 0., 0.], dtype=float32),
 array([0., 0., 0., ..., 0., 0., 0.], dtype=float32),
 array([107.,  56.,  35., ...,   0.,   0.,   0.], dtype=float32),
 array([2., 0., 1., ..., 0., 3., 1.], dtype=float32),
 array([0., 0., 0., ..., 6., 1., 5.], dtype=float32),
 array([0., 0., 0., ..., 0., 0., 0.], dtype=float32),
 array([ 0.,  0.,  0., ..., 44.,  2.,  5.], dtype=float32),
 array([12., 13., 13., ...,  4.,  2.,  6.], dtype=float32),
 array([11., 16.,  2., ..., 69.,  4.,  9.], dtype=float32),
 array([121.,  57.,  16., ...,   0.,   0.,   0.], dtype=float32),
 array([12.,  4.,  7., ...,  0.,  0.,  0.], dtype=float32),
 array([ 0.,  1., 10., ...,  0.,  0.,  0.], dtype=float32),
 array([60.,  2.,  3., ...,  0.,  0.,  0.], dtype=float32),
 array([128.,  92.,  22., ...,   0.,   0.,  19.], dtype=float32),
 array([36., 66., 43., ..., 34.,  5.,  2.], dtype=float32),
 array([6., 2., 2., ..., 0., 0., 0.], dtype=float32),
 array([26., 25., 21., ...,  0.,  0.,  0.], dtype=float32),
```

Figure 6. Cassava SIFT features

```
print(classification_report(y_orb_test, preds))

              precision    recall  f1-score   support

           0       1.00      0.89      0.94       199
           1       0.83      1.00      0.91       102

    accuracy                           0.93       301
   macro avg       0.91      0.95      0.93       301
weighted avg       0.94      0.93      0.93       301
```

Figure 7. KNN Classification Report on ORB features

3.2. Object detection

Object detection is a computer vision field, which involves detection of instances of an object from the image. In this task, I carried data annotation using the LabelImg tool.

I annotated banana and cassava plants from the images while using the bounding boxes, which help to determine the target plant by extracting the x and y coordinates for the image's borders.

I split the data into training, testing and validation with the use of the roboflow tool. I then used transfer learning of YOLOv5 to train the model. I considered this because it was previously trained on a famous COCO dataset of 80 classes with about 200,000 features. With this transfer learning, it helps to attain better accuracy since the model uses the best saved weights to learn new patterns in the dataset.

Figure 8. Detected annotated images

4. Experimental results

In this subsection, I visualise the results obtained from the model training, how the model predicts on the unknown images and how it was able to detect the objects after training for 150 epochs.

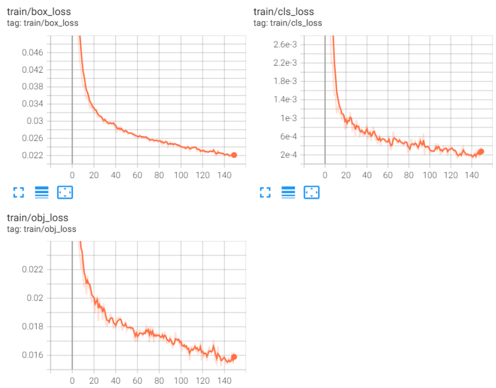


Figure 9. Visualisation of the training loss

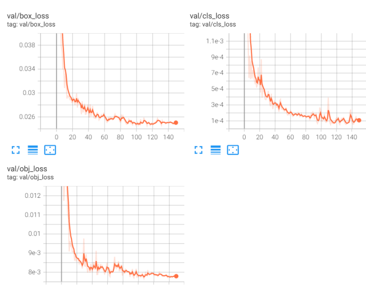


Figure 10. Visualisation of validation loss

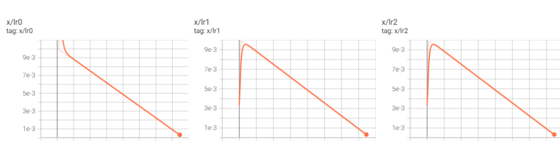


Figure 11. The learning rate curve visualisation

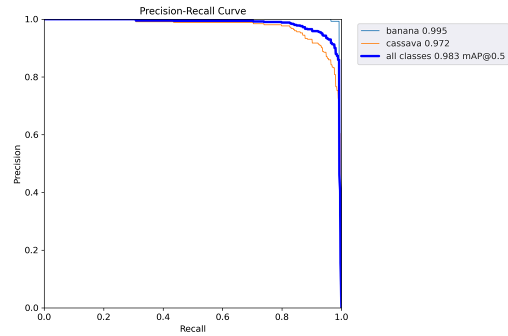


Figure 12. Precision and Recall curve

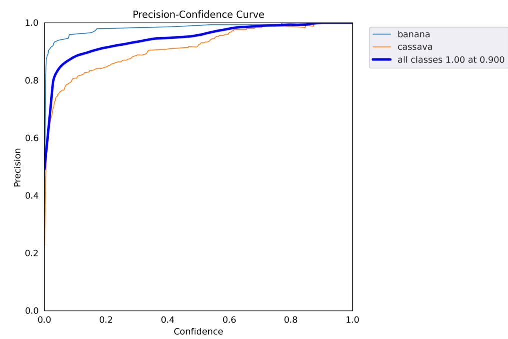


Figure 13. Precision confidence curve

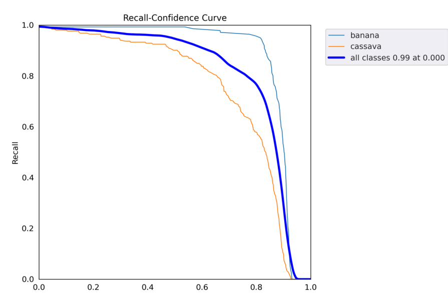


Figure 14. Recall confidence curve

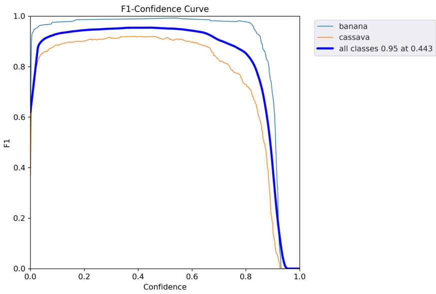


Figure 15. F1 confidence curve

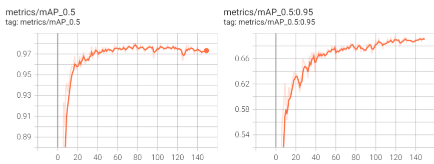


Figure 16. Mean Average Precision curve metrics

In conclusion, this report shows the potential of machine learning and computer vision in image classification and object detection tasks. The use of KNN and YOLOv5 showed great success on these tasks as seen from the results.
Both models provided optimal results, although YOLOv5 performed better as seen from the results and was computationally efficient. This therefore shows how Object detection can be a great potential for combating many challenges faced in agriculture sector.

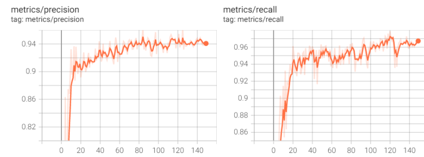


Figure 17. Precision and Recall curve metrics

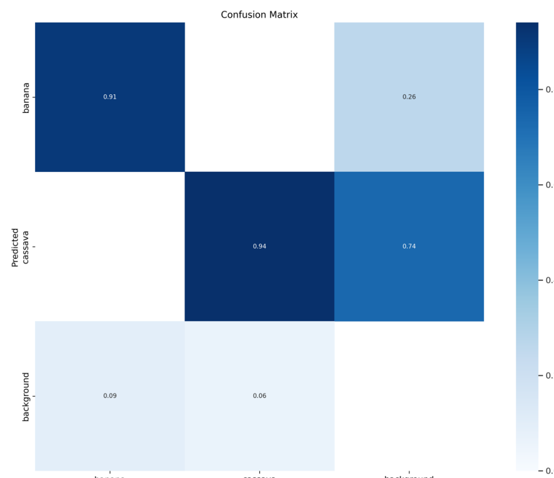


Figure 18. Confusion Matrix

```
150 epochs completed in 0.774 hours.
Optimizer stripped from runs/train/exp/weights/last.pt, 14.3MB
Optimizer stripped from runs/train/exp/weights/best.pt, 14.3MB

Validating runs/train/exp/weights/best.pt...
Fusing layers...
Model summary: 157 layers, 7015519 parameters, 0 gradients, 15.8 GFLOPs
                 Class     Images  Instances          P          R      mAP50   mAP50-95: 100% 25/25 [00:02<00:00,  9.44it/s]
                   all        195        393      0.924      0.956      0.978      0.694
                banana        195        140      0.967      0.993      0.995      0.789
               cassava        195        253      0.882       0.92      0.961      0.598
Results saved to runs/train/exp
```

Figure 19. Results from the model weights

Figure 20. Predictions from the unknown images

References

[1]. P. Siva Prasad and A. Senthilrajan, “An Adaptive Boosting Approach for Plant Classification through Deep CNN”, International Journal of Research Publication and Reviews, Vol.2, No. 3, (2021), pg 1-4.

[2]. M. D. Fauzi et al., “A Hybrid DenseNet201-SVM for Robust Weed and Potato Plant Classification”, Jurnal Ilmiah Teknik Elektro Komputer dan Informatika (JITEKI), Jurnal Ilmiah Teknik Elektro Komputer dan Informatika (JITEKI), Vol. 8, No. 2, June 2022, pp. 298-306.

[3].P. P. Kumar et al, ”Application of Remote Sensing in Agriculture”, Indian Journal of Natural Sciences, Vol. 12, no. 68, Dec. 2021. [4]. S. Sachar and A. Kumar, “Survey of feature extraction and classification techniques to identify plant through leaves,” Expert Systems with Applications, vol. 167, p. 114181, Apr. 2021

[5]. S. Barburiceanu, R. Terebes, and S. Meza, “3D Texture Feature Extraction and Classification Using GLCM and LBP-Based Descriptors,” Applied Sciences, vol. 11, no. 5, p. 2332, Mar. 2021.

[6]. P. P. Kaur and S. Singh, “Classification of Herbal Plant and Comparative Analysis of SVM and KNN Classifier Models on the Leaf Features Using Machine Learning,” Soft Computing for Intelligent Systems, pp. 227–239, 2021.

[7]. R. I. Borman, R. Napianto, N. Nugroho, D. Pasha, Y. Rahmanto, and Y. E. Pratama Yudoutomo, “Implementation of PCA and KNN Algorithms in the Classification of Indonesian Medicinal Plants,” IEEE Xplore, Oct. 01, 2021.

[8]. J. A. Villaruz, ”Deep Convolutional Neural Network Feature Extraction for Berry Trees Classification”, Journal of Advances in Information Technology, Vol. 12, No. 3, Aug. 2021.

[9]. J.Wang et al, ”DBA-SSD: A Novel End-to-End Object Detection Algorithm Applied to Plant Disease Detection”, MDPI, Vol.12, no. 11, Nov.2021.

[10]. A. M. Roy, R. Bose J. Bhaduri, ”A fast accurate fine-grain object detection model based on YOLOv4 deep neural network”,
Neural Computing and Applications, Vol.34,3895–3921, Jan. 2022.