Where's My Wine

About the Challenge:

The job is to process the pictures of the farmlands and detect the location of grapes in the farm for a camera-based system.

Data Set Available:

We are provided with 300 images in jpeg format directly from the camera sensor. For every individual image in the dataset, there is an associated label in the label directory in docx format containing the four coordinates of the bounding box around grape images. All the images provided are the images of the grapes in the farm.

Components: (Kindly do not change structure of directory 'final_detection':

Grape_training_notebook_colab – (Notebook)

This notebook was used for training of the object detection model on google colab. The notebook contains the script to convert the docx format of the labels to Pascal VOC XML which is then used as an input for training. Rest of the codes show how the training was done.

Grape_Detection – (Notebook)

This notebook is made to use for detecting grape object in an image by passing the path of single image, or by-passing path of folder having multiple images. The annotated output images will be shown on the notebook and will also get save in

Json – (Folder)

Having json file, that maps the objects names in your image dataset and the detection anchors.

Model – (Folder)

Having our custom model for grape detection

ImageAI_OCR – (Folder)

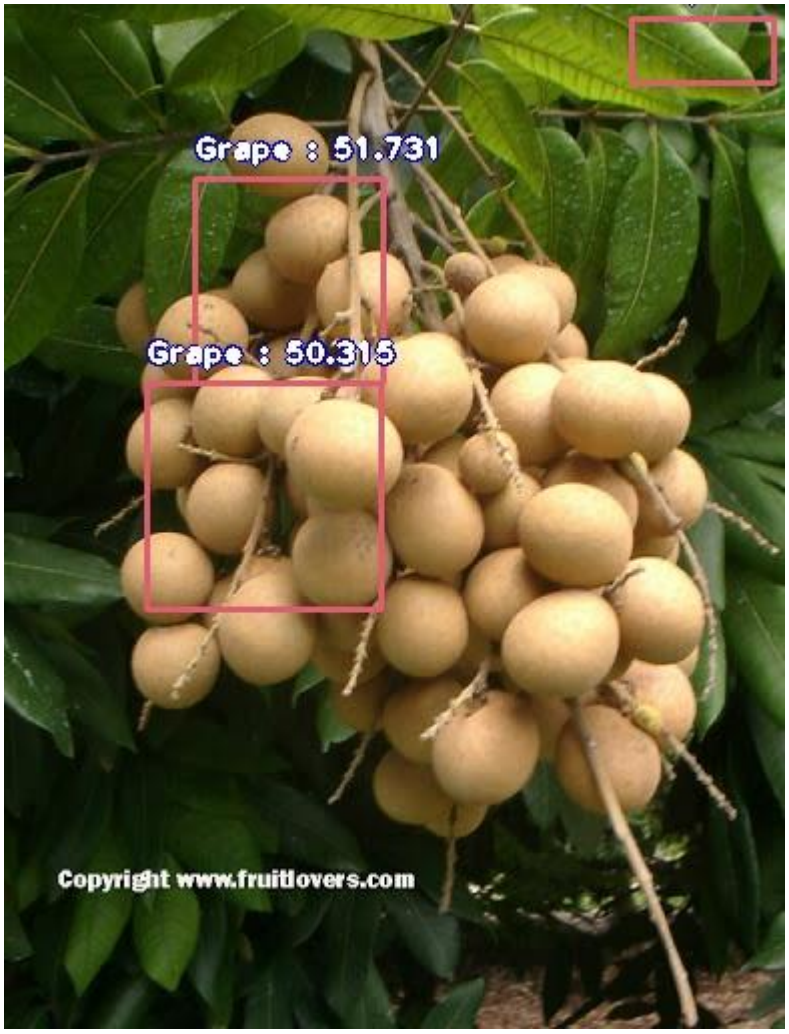
Our output Images are saved in this folder.

Libraries used:

- os
- io
- xml.etree.cElementTree
- PIL
- Imageai
- Matplotlib
- pathlib

Suggestions to Client:

1. As the data provided here is only of grapes and not of any other fruit, so model have no reference to distinguish what are grapes and what are not. This type of data may lead to model predicting grape like looking fruit or object as grape and create bias. Example:



In these photos (output from our model) we can see that the model is classifying and detecting grape like looking flowers and fruit as grape which should not be the case in ideal condition. Or we can increase 'minimum_percentage_probability' parameter, cause Lowering the value shows more objects while increasing the value ensures objects with the highest accuracy are detected so we must do trade-off accordingly.

2. The training data should have images from more angle's like closeup or top view because almost all the images are taken from safe distance and if the grapes are to be detected from close range or much far range or from strange angle, it may cause a problem.
   Example

Assumptions:

- The 300 images provided are sufficient for the task and no other images of different fruit is required or from different angles are required. More important is to detect the grape, if other fruit is also classified as grape, it is not a problem.
- The images will always be taken from safe distance.

Approach:

As we need a custom object detection model, so we decided to use Image AI python library to train our custom object detection model using YOLO v3 architecture. The training process generates a JSON file that maps the objects names in your image dataset and the detection anchors, as well as creates lots of models which can be compared afterwards.

We used Yolov3 because it is as accurate as SSD but much faster than it. It is comparatively slower then Yolov2 due to its complex structure, but It has much stronger performance then Yolov2. Yolov3 is a lot better cause feature extractor it uses is called Darknet-53 which is much deeper and better feature extractor. It is also better at detecting smaller objects then other models as different layer is responsible to detect different size object.

As the data is given in docx, first it is converted to Pascal VOC type XML using a script mentioned in the training notebook. Then the data is arranged in the directory in a particular structure which is required for Imageai library for object detection task and then model is trained over our custom data.

Data Pre-processing:

- Docs file format of labels are converted to Pascal VOC XML type which is compatible with imageai object detection.
- Data is structured in directory according to imageai compatibility.
- Data is split into train and validation with 250 for training and 50 for validation and then model is trained.

Model Training:

Parameters:

- Iou_threshold = This is used to set the desired Intersection over Union for the mAP evaluation. Intersection over Union is a ratio between the intersection and the union of the predicted boxes, and the ground truth boxes. IoU Threshold is used to calculate True Positives and False Positives in case of object detection.
- object_threshold = Confidence in classification threshold.
- Mean Average Precision is, literally, the average of all the average precisions (APs) of our classes in the dataset. As both classification and localisation are involved, so normal criteria can not be used, so we use mAP

After training the model, it is saved and then is used for object detection in Grape_Detection notebook.

mAP of the model used: 0.7447

Description of Grape Detection Notebook:

The notebook is kept in final_detection directory. This directory is having other folders too like

- model: having the pre trained custom model.
- json: Having the Json required by model for object detection.
- ImageAI_OCR: Where the output images are saved.
- Inputimage: Was used to keep input images (optional, another folder path can also be given).

The Script is made such that when path of folder having the input images is provided, the script reads all the images inside the folder, imports the model and json required and perform object detection on the image. The bounding box coordinates are printed on the notebook and the output image is saved in ImageAI_OCR folder and is shown on the notebook.

If path of single Image is provided, it can also be handled by the script. It will import the image, perform object detection, and then will save the output image in ImageAI_OCR folder and will show it in notebook.

References:

https://imageai.readthedocs.io/en/latest/customdetection/index.html

https://towardsdatascience.com/survey-d4f168791e57

https://towardsdatascience.com/what-is-map-understanding-the-statistic-of-choice-for-comparing-object-detection-models-1ea4f67a9dbd

https://stackoverflow.com/

https://towardsdatascience.com/yolo-v3-object-detection-53fb7d3bfe6b