Kenya Yield Study: Proof of concept for measuring yields – an artificial intelligence approach

Prepared for TechnoServe / Nespresso

SEPTEMBER 2018

laterite
DATA | RESEARCH | ADVISORY

laterite
DATA | RESEARCH | ADVISORY

Contents

Abbreviations

AI	Artificial Intelligence
ANN	Artificial Neural Network
COCO	Common Objects in Context
IOU	Intersection Over Union
RCNN	Regional Convolutional Neural Network

List of Figures and Tables

1. Aims of the Report

The Kenyan Coffee Yield Study[1] for the TechnoServe Farmer Training Program 2018 Cohort aims to 1) assess the quality of farmer recall data for land area, number of trees, and deliveries (yield) and 2) develop and test alternative methods to more accurately measure these variables. 600 farmers participated in the study, which uses data from the 2018 cohort baseline survey (carried out in January), and plot-level and recall data from a survey conducted in June 2018. The aim of this report is to 1) demonstrate proof of concept for using Artificial Intelligence (AI) to help collect more accurate yield data, and 2) to propose fieldwork slated at the end of September 2018 to test the proposed method. Specifically, the AI method focuses on using plot sampling techniques and an object detection algorithm to count the number of cherries on coffee branches. After the September data collection, we will develop a model to extrapolate yield data from subsamples of trees and tree branches on each plot, wherein the object detection algorithm will count the number of cherries on each branch. Since the images for the Proof of Concept Report were taken during the early harvest (June 2018), we did not use them to try to estimate yield for this report, but have used them to show that a computer can count cherries from an image.

The report will be split in three parts. First, we will discuss the methodology and results of the initial object detection tests. Then, we propose a field study for using this AI technology to help estimate coffee yields. Lastly, we will discuss extensions to the current methodology.

2. Methodology

The goal of the initial object detection evaluation is to capture images of branches with ripe cherries and assess whether an AI program can count the number of cherries in the image.

2.1 Image Capturing

We visited three farms in Kirinyaga and Nyeri counties and used two people to take photos. One person placed a royal blue poster board behind the coffee branch, and the other took a photo of the branch. The choice of colour in poster board was decided based on its contrast with the colours naturally occurring in coffee branches and cherries. The photos were taken with an iPhone6 (8 megapixels, $f/2.2$ aperture, and autofocus). Using this photo-capturing method, we took 291 photos that included 8,133 cherries (both red and green). The initial photo size is 2,448 x 3,264 pixels (or 3,264 X 2,448 pixels depending on the orientation). Image sizes were reduced by a factor of 6 to 544 x 408 to reduce file sizes and training time.

2.2 Labelling

[1] The Kenya Yield Study is being undertaken by TechnoServe, Laterite, Wageningen Universty and Research, and Nespresso.

All 291 photos were labelled using a Python-based program, LabelImg. We drew bounding boxes around each cherry in the photo, and labelled them as cherries. This process is time consuming, but once a large enough library of labelled images is created, the process does not necessarily have to be repeated when new photos are taken. Nevertheless, more labelled images should improve performance.

2.3 Training and Test Sets

After labelling, the images were converted to TFRecord files, the datatype used in Google's object detection API, TensorFlow (see below). At this stage, we also performed the training-test split, assigning roughly 75% of images to the training dataset (the images to which the model is fit) and 25% to the testing dataset (the images for which the model tries to detect cherries). A diagram is shown in Figure 1 below. Having training and test datasets is at the heart of classification problems in data science and can be likened to fitting a forecasting model (training) and having out-of-sample forecasts (testing).

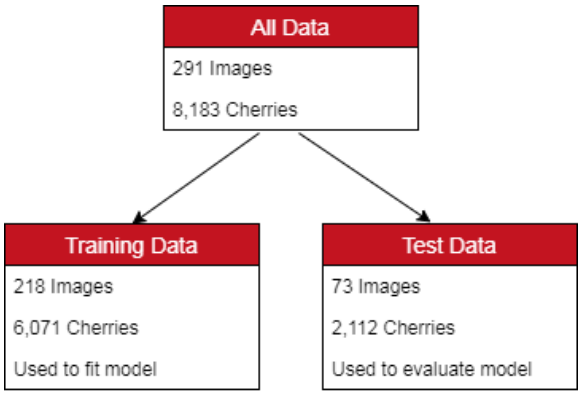

Figure 1

2.4 Model and Training

The object detection model was trained in TensorFlow using a Faster Regional Convolutional Neural Network (Faster - RCNN) (Ren, et. al., 2015). Artificial neural networks (ANNs) are typically used for image classification or object detection purposes, and various pre-trained models are provided by Google. We chose to use a model previously trained on the Microsoft Common Objects in Context (COCO) dataset (a collection of 200,000+ images and 1.5 million objects from 91 categories) (Lin, et al., 2014). It is important to use a model trained on irrelevant objects so that the algorithm can learn what is *not* a coffee cherry in addition to what *is* a cherry. Using this trained model as a starting point, we retrained the model using our tagged training images. Because training an ANN is computationally expensive, the model was trained on a cloud server using FloydHub. The results were then downloaded from the server and analysed.

2.5 Testing

After training, we then tested the model, i.e. tried to detect cherries in images the model had never seen before. With the 73 testing images, the program drew bounding box estimates around each estimated cherry and calculated the probability of each estimate belonging to its estimated class (cherries). We then analysed the performance of the model on an image-by-image basis and more rigorously, across the entire testing set. Details on the evaluation metrics and results are reported in the next section.

3. Results

Overall, the model performs quite well, indicating that this is a viable method for counting the number of coffee cherries on a branch. This section demonstrates results by showing a subset of images displaying the algorithm's estimates and by displaying more complete evaluation metrics inspired by the COCO object detection challenge (Lin, et al., 2014).

3.1 Image Analysis

Before looking at more rigorous evaluation metrics, seeing the images and the estimated bounding boxes gives a visual understanding of what the algorithm is estimating and can help in understanding the strengths and weaknesses of the model. Below we present several images of model estimates. The boxes are the estimated bounding boxes, and the corresponding percentages are the probabilities that the bounding box contains a cherry. From the images, we can see that the model performs well with only several false positives (incorrectly identified cherries) and false negatives (undetected cherries).

Figures 2 – 5: Predictions with Bounding Boxes

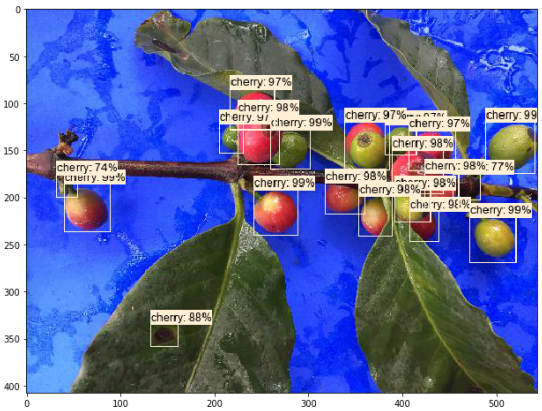

Figure 2

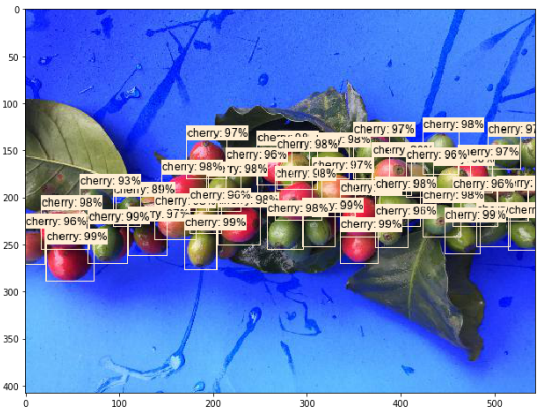

Figure 3

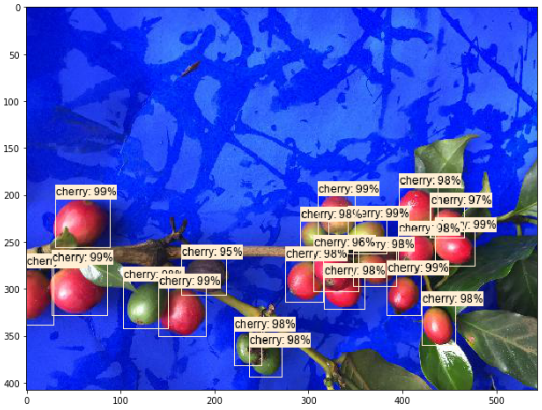

Figure 4

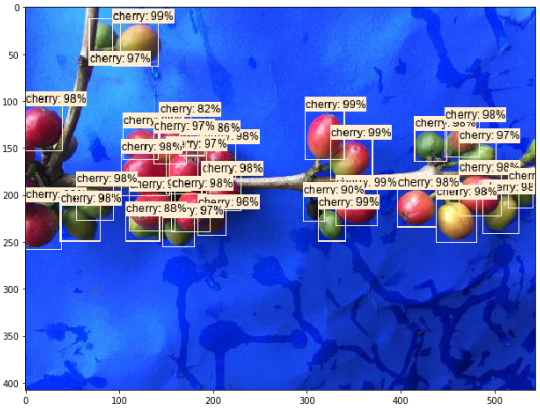

Figure 5

3.2 Evaluation Metrics

The evaluation metrics used here are similar to the evaluation metrics used in the COCO object detection challenge (Lin, et al., 2014). Namely, we will use precision and recall to determine how well our model performs. These measures are based off of the intersection over union (IOU) calculation, which is a measure of how much the estimated bounding box overlaps with the ground-truth box. More specifically, IOU is the area of intersection of the bounding box and ground-truth box divided by the sum of the total area of the two boxes (see Figure 6). As a result, each IOU is a number between 0 and 1, with 1 indicating perfect overlap, and 0 indicating zero overlap (Long, et. al., 2015).

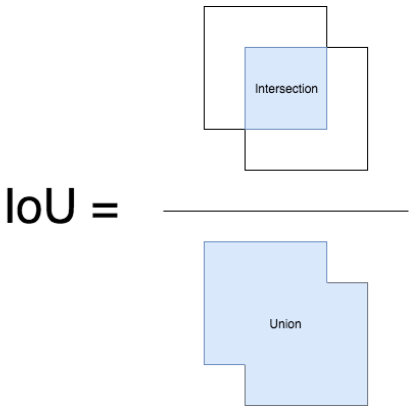

Figure 6

We then calculate the number of estimates that are true positives (correctly detected cherries), false positives (detections that are not cherries), and false negatives (undetected cherries)[2]. The IOU of a cherry detection determines whether the cherry is detected correctly or not. If the IOU is greater than a chosen threshold (typically 0.5), then we say that the detection is a true positive; if IOU is less than the threshold, then it is a False Negative; the difference between total detections and true positives is the number of false positives. From these three values, we can calculate precision and recall:

$$precision = \frac{true\ positives}{true\ positves + false\ positives}$$

$$recall = \frac{trues\ positives}{true\ positives + false\ negatives}$$

Intuitively, precision is the fraction of detections that are actually cherries, and recall is the fraction of cherries that get detected. Average precision and average recall are simply averages of the precision and recall for all images tested. In both calculations, we vary the IOU threshold from 0.05 to 0.95 and take the mean of precision and recall, giving us mean average precision and mean average recall, respectively.

3.3 Evaluation Results

Using the above metrics, we calculate mean average precision and mean average recall for the entire dataset as well as average precision and average recall using a 0.5 IOU threshold and a stricter, 0.75 IOU threshold. We also calculate mean average precision and mean average recall for terciles of image size and number of detections[3]. Based on the following results, we conclude that using a 0.5 IOU threshold is the optimal choice for application because of its superior performance and intuitive implementation. The 0.5 threshold is common as well; it was used in model evaluation for the PASCAL Visual Object Classes Challenge (Everingham et. al., 2015).

The average IOU is 0.70 with a median of 0.82. When we use the 0.5 IOU threshold, the overall precision is 0.96 and the average recall is 0.84. This means the model detects 84% of all cherries, and when the model does detect a cherry, the detection is correct 96% of the time. Naturally, the mean average and stricter, 0.75 threshold yield poorer results, but still indicate that the model performs well under tougher evaluation criteria. See Table 1 for complete reporting.

Table **1**

[2] Please note that true negatives are not calculated because they are everything that is not a cherry and not detected, which is an impossible statistic to calculate. For this reason, accuracy (another typical Machine Learning evaluation metric) is not used.

[3] The COCO object detection challenges uses absolute area definitions and 1, 10, and 100 as cut-offs for number of detections instead of terciles. We chose to use terciles because of the relative homogeneity of our single detection object. The maximum number of detections was 57, and most cherries have similar sizes in area, whereas the COCO dataset includes quite diverse detection categories and sizes.

Overall Precision and Recall		
	Precision	Recall
Mean Average	0.88	0.77
0.75 Threshold	0.78	0.68
0.5 Threshold	0.96	0.84

Next, we check for heterogeneity in results based on detection size and number of detections per image. For detection size, we group the detections into terciles based on the area of the detection boxes. For the number of detections per image, images are grouped into terciles based on the number of detections in each image.

Table 2 suggests that the model performs much better with large cherries when compared to small cherries. Precision in Tercile 3 is 0.20 points above precision in the first tercile and recall is 0.88 for Tercile 3, compared to 0.51 in Tercile 1.

Table 2

Precision and Recall by Image Size		
	Precision	Recall
Tercile 1 (small)	0.68	0.51
Tercile 2 (medium)	0.76	0.70
Tercile 3 (large)	0.88	0.88

Finally, Table 3 shows that precision is not affected by the number of detections in a given image, but recall is slightly lower when there are many cherries in an image. This phenomenon is likely linked to Table 2 above because images with more cherries tend to have smaller cherries as well (in the image). Training on more images after the next data collection may help alleviate this issue.

Table 3

Precision and Recall by Number of Detections		
	Precision	Recall
Tercile 1 (few)	0.89	0.80
Tercile 2 (some)	0.88	0.81
Tercile 3 (many)	0.88	0.75

4. Proposed Data Collection

We now want to test whether the number of cherries counted in an image is a good predictor of the number of cherries on a real branch, and more importantly, whether we can extrapolate yield from counting the number of cherries on a subsample of trees and on a subsample of branches. To do so, we must start data collection before the main harvest (late September 2018). This section will outline the proposed data collection process.

4.1 Goals

The goal of data collection is to get population estimates of the number of cherries on tree branches in a farm so that we can extrapolate yield during analysis. We want to obtain estimates on how many cherries a tree can produce by having enumerators take random subsamples of trees and subsequent random subsamples of branches on each sampled tree. The enumerator will then count the number of cherries on each sampled branch and take a photo of the branch using the methodology described above. Using this cherry-count data, we will try to estimate yield.

4.2 Sampling

Using Slovin's formula[4] , we have determined that 100 farms will be sampled to give us a 90% probability of a representative sample of farms. Within each farm, we want to get a representative sample of the number of branches on each tree. With an error tolerance of 0.2 (80% probability of a representative sample), we can randomly select 22 trees on the average farm[5]. Since we are not concerned about getting cherry counts for a representative branch for each tree, but rather a representative branch at the farm-level (because we are not trying to find yield per tree, but rather yield per farm), we can determine a sample size using the farm-level, not tree-level population of branches. During field-testing (see below) we will get a more accurate prior for the number of branches per tree, but for now we can assume each tree has 30 branches[6]. Using Slovin's formula with the population being the number of branches on the farm, we can sample 1 branch per sampled tree for an error tolerance of 20% or 2 branches per tree for an error tolerance of 15%. We prefer the lower error tolerance, and we will test the feasibility of a larger sample during testing.

---

[4] Slovin's formula: $n = \frac{N}{1+e^2}$, where N is the population size, n is the sample size, and e is the margin of error.

[5] Although the median value of trees per farm is 167, using the median value does not substantially alter the sample size needed.

[6] The number of branches needed is robust to changing branches per tree in the interval of 10 to 50. The curve of Slovin's formula flattens as the population size grows. This is why, for example, national polls can sample relatively few people.

The enumerators will use a random walk sampling method to sample trees. From prior tree count data, we will calculate the sample size of each farm, and the enumerator will start at the first tree on the plot, walk through the field, and sample every j^{th} tree, where $j = \frac{N}{n}$. At each tree the enumerator will then count the number of branches, and SurveyCTO will tell them which branch(es) to sample.

In summary, we will sample 100 farms from the 600 participants in the cohort study. On the average farm, we will sample 22 trees and 2 branches per tree. Enumerators will implement a random walk sampling method.

4.3 Challenges

Ensuring that the photos are of adequate quality and that the enumerators are able to take pictures alone will be the primary challenges of data collection. We propose testing methods in the field beforehand and providing one day of intensive on-farm training for enumerators before data collection begins. Finally, data uploading will be costly in terms of data used and in terms of server space. We will develop methods to alleviate these problems.

4.4 Timelines

The primary harvest begins on the week of September 24th, and we should start data collection on this date so that our samples are not affected by the ongoing harvest. During the week of September 17th, we will conduct on-farm testing, make appointment calls, develop brief training materials, and develop a survey instrument. On Monday, September 24th we will have intensive on farm training for the enumerators and they will all pilot one plot. Data collection will begin on the 25th. In June, enumerators were able to complete four farms per day. We expect this data collection process to be similar, perhaps even a little slower. At this rate, we can spend 5 days in the field and 1-2 days for mop up if we use 5 enumerators. Data collection should be over by Friday, October 6th at the latest. We can get a better sense of timings after the on-farm testing. To obtain the benchmark yield data, cooperative data will be collected again in early 2019 after the harvest.

5. Extensions

Several improvements can be made to the object detection algorithm. First, during the tagging phase, we can use more labels (e.g. 'red cherry', 'green cherry', and 'diseased cherry'). Such labels will provide more information on yields, as well as plant health. With more pictures coming from the proposed fieldwork, we should be able to have more detailed classifications of cherries without loss of model performance.

Secondly, we can experiment with measuring cherry size using a reference point. If we capture the reference point in an image, then we can determine the physical size of each cherry. With this information, more detailed estimates of yield can be made.

Finally, we can make improvements to the Faster RCNN model. Although model performance is quite strong, we are still using a generic model. Tweaks to the model could induce modest improvements in performance.

After data collection, we will tag images, re-train the model, and analyse results. In January, we will get the 2018 harvest data from cooperatives. At this point, we can match our model predictions to the benchmark cooperative data. A report detailing results will be completed once the cooperative data is collected. The analysis will focus on comparing model estimates to cooperative data.

6. References

Everingham, M., Eslami, S. A., Van Gool, L., Williams, C. K., Winn, J., & Zisserman, A. (2015). The pascal visual object classes challenge: A retrospective. International journal of computer vision, 111(1), 98-136.

Lin, T. Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., ... & Zitnick, C. L. (2014, September). Microsoft coco: Common objects in context. In European conference on computer vision (pp. 740-755). Springer, Cham.

Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 3431-3440).

Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. In Advances in neural information processing systems (pp. 91-99).