Kenya Yield Study: Artificial intelligence feasibility report – imaging study report

Prepared for TechnoServe / Nespresso

AUGUST 2019

laterite
DATA | RESEARCH | ADVISORY

From data to policy

www.laterite.com

Acknowledgements

This report was prepared by Laterite Ltd. Laterite would like to thank Nespresso for their funding and continued support of the Kenya Yield Study.

Laterite would also like to thank TechnoServe for their time, insights, and assistance in the Kenya Yield Study. We appreciate the support from TechnoServe management, especially Paul Stewart and Carole Hemmings. In particular, Laterite would like to thank Carole Hemmings for her agronomic insights during initial field visits and throughout the study. We would also like to thank Arthur Nganga, Beth Wangui, and Alice Ngugi for their support of the research.

Laterite also thanks all participating farmer cooperatives: Rwama FCS, Kibugu FCS, Tekangu FCS, Rugi FCS, Mugaga FCS, New Gikaro FCS, Mwirua FCS, and Barichu FCS. The managers of these cooperatives have been helpful in facilitating this study. Particularly, we would like to thank these cooperatives for sharing their delivery data, making this research possible.

We acknowledge the support of members of the research team at Laterite who have supported this project. Francisco Garcia Jordan, Ganesh Rao, and Dimitri Stoelinga have offered their continuous assistance and insights. Laterite also thanks the data collection team led by Anne Wagaturi with the support of Nicholas Cohen.

The research team for the Kenya Yield Study is comprised of Michael Keenan[1], Emily Fabrace, and Sachin Gathani.

The report was delivered by Michael Keenan.

[1] Wageningen University and Research and International Center for Tropical Agriculture

Table of Contents

List of Figures

List of Tables

List of Abbreviations

AI	Artificial Intelligence
FCS	Farmer Cooperative Society
KNN	K Nearest Neighbors
ML	Machine Learning
NW	Nadaraya-Watson
RCNN	Regional Convolutional Neural Network

Executive Summary

Laterite, in partnership with TechnoServe and Nespresso, carried out a study on coffee yield data in Central Kenya. The study aims to 1) assess the quality of yield data and 2) to develop an Artificial-Intelligence (AI) assisted model to predict/measure coffee yields.

The imaging component of the study assesses the viability of using pictures of coffee branches to count coffee cherries with AI. The AI algorithm was initially tested based on its ability to accurately count cherries from pictures and was checked against human counts *from pictures*. The algorithm was able to detect 84% of cherries from the pictures and was correct in classifying cherries 96% of the time.

The AI methodology is now being tested against enumerator counts of cherries from tree branches (rather than pictures). The initial AI methodology, using Nueral Network object detection, detects the number of cherries visible in a photograph but some cherries on the tree may not be visible in the picture. Therefore, a second stage of AI modelling, using Machine Learning (ML), is required to estimate the total number of cherries on the tree (visible and not visible).

Predicted counts using the Neural Network methodology combined with ML regression, do NOT statistically differ from enumerator counts. The results suggest that an AI-based algorithm can be as good at counting cherries as a human being.

There are several downsides to the AI-based approach, making implementation difficult in the future. Taking photos takes longer than counting cherries on average; time-requirements for this methodology are otherwise quite high; there is no guarantee that the model will work in other contexts without further human calibration.

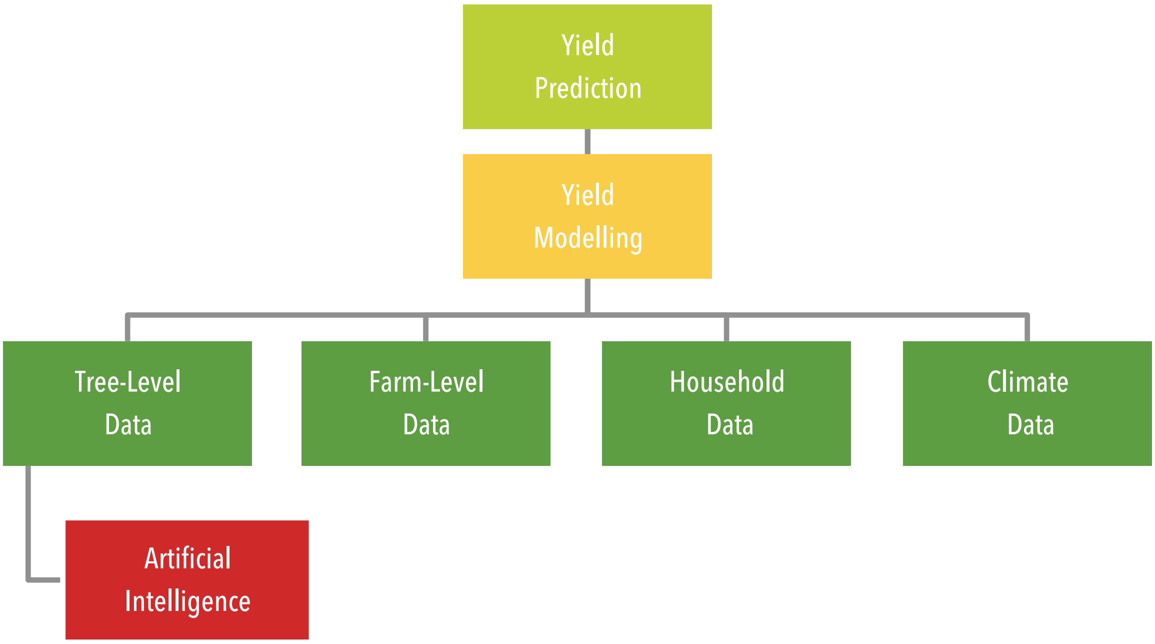
Introduction

This report intends to 1) provide an overview of the imaging component of the 2018 Kenya Yield Study and 2) demonstrate the feasibility of Artificial Intelligence (AI) to count coffee cherries from images. This analysis is part of a larger study on measurement of coffee yields in Central Kenya and provides the basis for the use of images in the modelling of coffee yields. Overall, the report shows that image-based AI algorithms are as effective at counting cherries as enumerators.

Obtaining cheap, objective and accurate smallholder coffee yield data is a challenge facing researchers and program implementers. Poor data quality negatively affects the analysis of coffee-related programs, and can even lead to misleading results. An earlier report on farmer recall error in the Kenya Yield Study demonstrates that farmers cannot accurately report their yields. One approach to obtaining objective and more accurate data is through yield modelling, whereby a model predicts coffee yields. The cooperative structure in Central Kenya allows us to build such a model as cooperatives keep detailed records on farmer deliveries of coffee cherry.

We intend to build a coffee yield model using household data, farm attributes (e.g. number of trees and area), satellite-based weather data, and tree-level data (e.g. number of branches and number of cherries). See Figure 1 for a conceptual framework of the model. The focus of this report is on using AI to estimate tree-level data, contributing to one component of the overall model. Manually counting cherries on each sampled tree can be 1) time-consuming and 2) biased due to enumerator effects. Simply taking pictures of the branches and counting the cherries with AI algorithms may provide a solution to increase speed and reduce enumerator bias.[2]

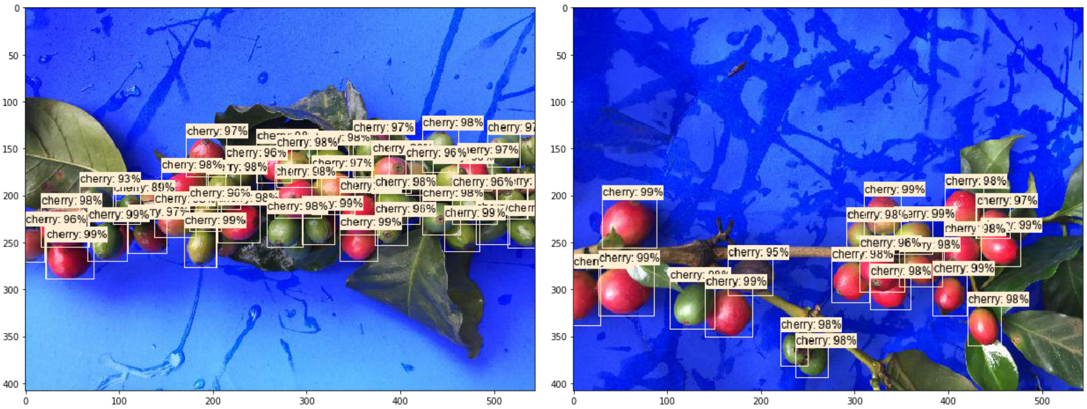
Figure 1: Conceptual Framework of Yield Modelling



[2] Preliminary evidence shows that manually counting cherries is faster than taking pictures. However, the object-detection algorithm can be easily converted to a video-based algorithm that would presumably drastically increase the speed of obtaining images. The drawback to this approach is increased data usage – we would need to manually classify many more photos.

In an initial Proof of Concept report on the feasibility of using object detection algorithms, the object detection algorithm counted 84% of cherries from images (this refers to the 'sensitivity' of the algorithm), and when it classified an object as a coffee cherry, it was correct 96% of the time (this refers to the 'precision' of the algorithm). The Proof of Concept algorithm was measured against human counts of cherries *from* images, not on human counts from branches. Figure 2 displays some examples of the algorithm's output. Based on these results, the imaging program was rolled out to 98 farmers from the KE Yield Study sample, and we now 'ground-truth' the algorithm (i.e. test the algorithm against enumerator cherry counts of branches).

Figure 2: Object Detection Examples



Sampling Methodology

There are 5 different levels of sampling in the study:

1. Farmer sample
2. Tree Sample
3. Stem Sample
4. Branch Sampling
5. Image Sampling

While the farmer sample was taken a priori, the enumerators took the tree, stem, branch, and image samples during the survey. We now discuss the sampling for each level.

Farmer Sample

From the original sample of 600 farmers in the Kenya Yield Study, 100 farmers were selected as part of the imaging study based on several criteria.[3] Farmers to be included in the imaging sample frame were required to meet the following criteria:

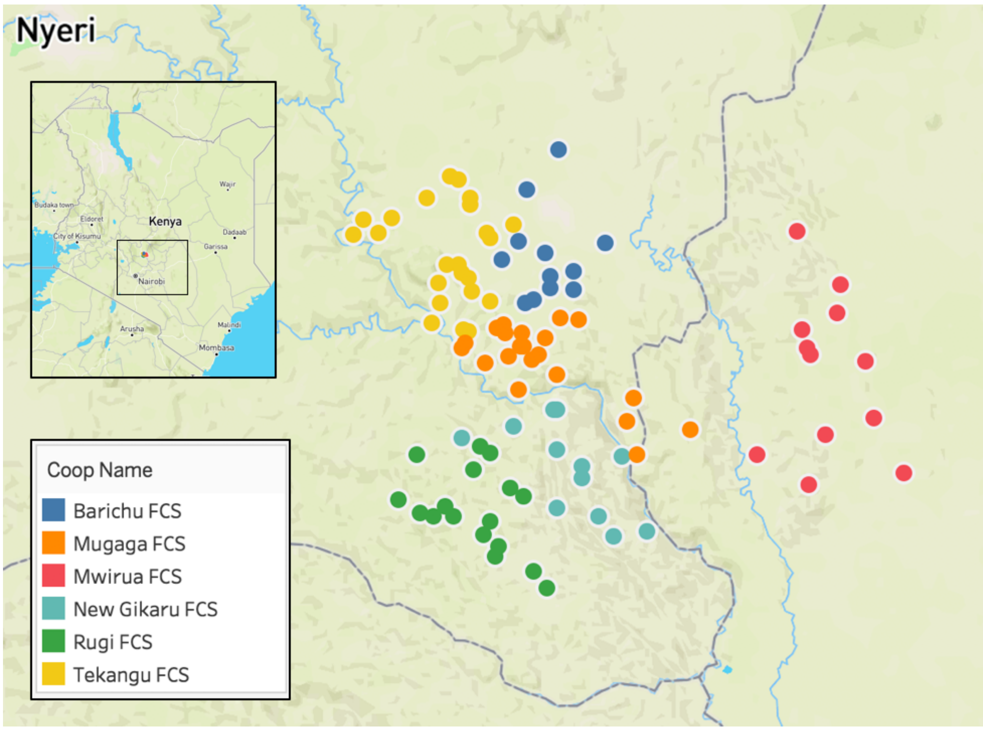
[3] Since this portion of the study is not meant to be representative of the sample and is only testing whether a crop-modelling methodology works, it is not necessary to have a completely random representative sample.

1. Have only one plot;
2. Do not report side-selling;
3. Have a cooperative-reported yield greater than 0;
4. Are on the Karatina side of the 2018C study area;
5. Do not report receiving or giving gifts of coffee;
6. Do not report having post-harvest losses;
7. Had more than 20kg of cooperative-reported yield in the 2017/18 harvest year;
8. Have fewer than 500 trees.

These criteria were chosen because they limit the sample to farmers whose cooperative-reported yield will (presumably) be the true yield if the farmers' behaviour is the same in 2018/19. Secondly, these criteria are chosen because they make the logistics of data collection simpler by reducing the area of the study location and eliminating very large farms that would require lengthy farm visits.

292 farmers met these criteria, and 100 farmers were randomly chosen to be part of the imaging sample. Of the intended sample of 100 farmers, surveys for 98 farmers were successfully completed. See Figure 3 for a geographic distribution of the sample.

Figure 3: Map of the Imaging Study Area



Tree Sample

For each farm, we use Slovin's formula[4] with an error tolerance of 0.20 to calculate the number of trees to sample.[5] For the trees sampled, we want to count the number of branches on each tree to get a representative sample of branches per tree on each farm. The number of trees to be sampled ranges from 10 to 24, and is 21 on average.

[4] $n = \frac{N}{(1+N\,e^2)}$, where *N* is population size, *n* is sample size, and *e* is error tolerance.

Enumerators employed a systematic random sample of trees on each farm. Based on the sample population numbers given by Slovin's formula for each farm, enumerators were instructed through SurveyCTO to sample every nth tree starting at a random tree number between 1 and n. For example, an enumerator may sample 20 trees from a farm of 300 trees. So, they were instructed to sample every 15 trees starting at a random tree between tree 1 and 15. Exact instructions for this sample can be found in Appendix 1.

Stem Sample

Enumerators randomly sampled 1 of the main stems of each coffee tree. Most coffee trees have more than one main stem and can have up to 4 main stems. When running the sampling analysis conducted above for trees for stems, the total number of samples need does not change (by the nature of Slovin's formula, whereby the required sample size increases very slowly after the population size grows larger), so it does not affect our analysis to sample stems instead. As a result, enumerators were told to count the number of main stems, and then were given a random stem to sample. See Appendix 1 for details.

Branch Sample

For each farm, we use Slovin's formula to calculate the number of branches to be sampled with an error tolerance of 0.15. The lower error tolerance was chosen because we thought it is logistically possible to sample more branches than the 0.20 error tolerance allows. We generated estimates of the number of branches in a field by multiplying the number of trees on each farm by b, where b is the number of branches per tree and ranges from 10 to 50. In each case (b = 10, 20, 30, 40, 50), Slovin's formula indicates that 44 branches should be sampled on each farm[6].

Branch sampling was also done with a random walk approach. Enumerators first counted the branches, and then they were instructed to sample a random branch using a systematic random sample. This methodology attempts to curb enumerator bias in picking branches with either small or large amounts of cherries. Exact instructions can be found in Appendix 2.

Image Sample

¼ of sampled branches required the enumerator to also take an image. During the pilot, enumerators were instructed to take images of every branch sampled, but this proved to be highly inefficient in terms of time. We decided to take images of ¼ of the branches sampled. As a result, 1444 images were taken for 98 farms.[7]

Data Collection

Data collection for the Imaging study was carried out in early October 2018. The dates for data collection were chosen to ensure that data collection was carried out as soon as possible to harvest without the main harvest having started[8]. For the 15-day date collection, a team of 6 enumerators was deployed. Results from piloting suggested that pairs would increase the efficiency, reliability, and accuracy of data collection, so enumerators worked in three teams of 2.

[5] The total number of trees for each farm was counted during the initial Yield Study data collection. The formula was run for stems, rather than trees. We assumed that each tree had 3 stems. The sample population does not depend on using trees or stems.

[6] This is the case because at large population sizes, the curve generated from Slovin's formula levels off. 44 may not seem very large, but our error tolerance is relatively high. With an error tolerance of .05, we should sample 366 trees per farm if we assume 30 branches per tree.

[7] Since multiple images were taken for some branches, we only have data for 1067 branches.

[8] In Kenya, one drawback to this approach is that there are two coffee harvesting seasons and the image analysis is only done before the main season.

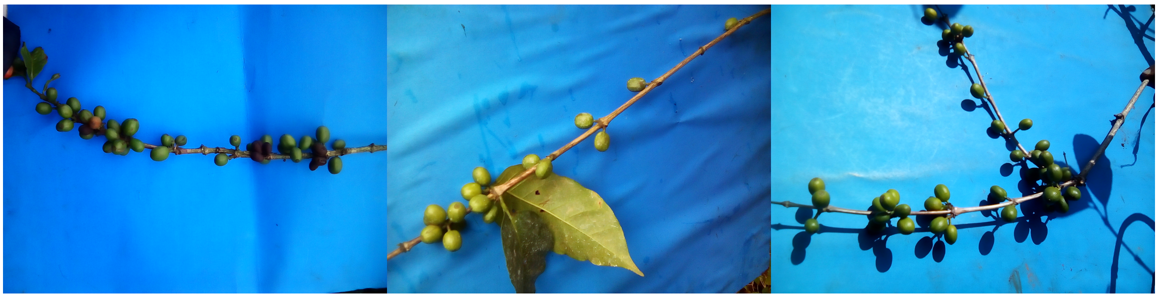
Cherry Counting Methodology

The enumerators were instructed to count all green and red cherries, but not brown cherries or diseased cherries that would not yield any marketable cherries. To make the process easier for large branches with many sub-branches, enumerators could count and enter into the tablet each sub-branch individually. See Appendix 2 for the complete protocol on cherry counting.

Imaging Methodology

To take photographs of the branches, enumerators were instructed to take photos of the entire branch, while capturing all cherries possible in the photo. One enumerator from each pair held a standard-sized (61 cm by 91 cm) royal blue poster board behind the branch for a solid-coloured backdrop, while the other enumerator took photos with the tablet. The poster board ensured that other branches were not captured in the photos, helping eliminate over-prediction from the algorithm. The poster-board also drastically improved algorithm performance during the initial stages of development. Some examples of photos can be seen in figure 4. See Appendix 3 for complete details on the photo-taking methodology.

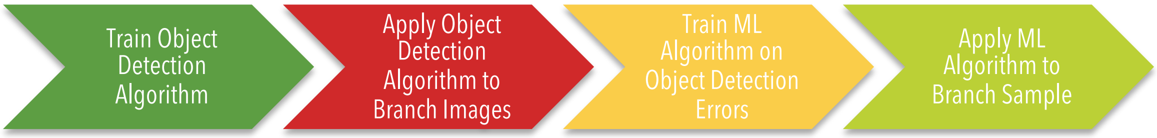
Figure 4: Cherry Image Examples



Prediction Methodology and Results

The model for predicting the ground-truth number of cherries has two layers. In the first layer, the object detection algorithm predicts the number of cherries from each image. In the second layer, we apply a machine-learning algorithm to correct for the errors of the object detection algorithm. Figure 4 demonstrates the process for predicting the 'true value' of cherries on each branch.

Figure 5: Cherry Prediction Process


Train Object Detection Algorithm
Apply Object Detection Algorithm to Branch Images
Train ML Algorithm on Object Detection Errors
Apply ML Algorithm to Branch Sample

Step 1: Object Detection Algorithm Training

The details of the object detection algorithm are given in the Proof of Concept report; however, we provide a brief overview here. The object detection algorithm is an application of Google's Python-based TensorFlow software. We implement a Faster Regional Convolution Neural Network (Faster – RCNN) algorithm to train the model (Ren, Girshick, and Sun, 2015). Because training is computationally expensive, we trained on a cloud server through FloydHub (an interface for interacting with Amazon Web Service cloud servers).

The algorithm was trained on 218 images, which included 6,071 images and was tested on 73 images containing 2,112 cherries. The images were obtained during the research team's visits to several farms during the first round of the Kenya Yield data collection[9]. The results indicated that the algorithm was able to detect 84% of cherries in the photos and when it did detect a cherry, it was correct 96% of the time.

Step 2: Apply the Object Detection Algorithm To Study Images

We applied the trained model to the 1068 branches from the imaging study. These images were not seen by the model beforehand and were not classified by a human. Using a 50% probability threshold (as is standard practice), the models results were stored and linked to the enumerator counts. We then compared the results from the model with the enumerator counts.

The results at this stage provide unconvincing evidence that the algorithm can count cherries as accurately as an enumerator. On average, the algorithm count about 11.4 cherries compared to the average of 16.2 cherries counted by enumerators. This result is unsurprising given that many cherries may not be visible to the algorithm (e.g. cherries that are behind other cherries or behind leaves). Further, the object detection approach shows a trend of over-estimating the number of cherries for branches with a low number of enumerator-counted cherries and grossly under-estimating cherry counts for branches with high numbers of enumerator-counted cherries. Table 1 displays this trend.

Table 1: Object Detection Counts vs. Enumerator Counts

Group (Based on Enumerator Counts)	Object Detection Mean	Enumerator Mean	Difference[10]	Observations
0	0.95	0.00	0.95***	206
0-5	3.70	2.81	0.89***	300
5-10	8.03	7.64	0.39**	158
10-15	12.04	12.92	-0.88***	93
15-25	16.26	20.25	-3.98***	105
25-50	23.93	36.17	-12.24***	123
50+	46.69	91.82	-45.20***	82
Overall	**11.40**	**16.28**	**-4.86*** **	**1067**

From this analysis, we can see that relying solely on the model's predictions is not enough for accurate cherry counting.

[9] The images used for training were captured using an iPhone6 (8 megapixels, ƒ/2.2 aperture, and autofocus), while the images in data collection were taken on Kruger & Matz Eagle 701 tablets with only 2 megapixel cameras. So, the quality of photos in the training set is much higher than the quality of photos captured in data collection.
[10] We tested the significance level of the differences using t-tests. p < 0.01: ***, p<0.05: **, p<0.1: *

Step 3: Train Machine Learning Algorithm for Error Correction

Since the errors seen in the previous section demonstrate a pattern, we can think of it as systematic error in counting the cherries and correct for this error via machine learning algorithms. Intuitively, the machine-learning algorithm 'learns' the errors of the object detection algorithm and attempts to predict the true values (enumerator counts) based off of the object detection algorithm's value. Ideally, this corrects for the systematic over and under-prediction of the object detection algorithm. The only variable used in this prediction is the prediction value from the object detection algorithm.

This approach requires enumerator counts to train the model and cannot be implemented solely from images, making this approach not purely image-based. Despite this drawback, the data we have already collected can be used to implement the model in the future, meaning that we do not necessarily have to collect enumerator counts in the future for the algorithm to work. If this methodology were to scale up, future studies would not require enumerator counts. However, having some human estimates in future studies would be best practice to validate the model.

We take the standard machine-learning approach and partition the 1068 observations into training (70%), test (15%), and validation (15%) sets. The algorithm is only trained on the training set. The test set is used to tweak the algorithm in an effort to improve prediction results. The validation set is only used after the model has been finalized and is not used in the design of the model. The use of the validation set provides evidence that the models' predictions are not simply a result of over-fitting. See Table 2 below for an overview of the partitions. We ensure that the mean number of cherries is not statistically different across the three groups by conducting t-tests – none of the differences is significant, indicating that we have properly balanced sets.

Table 2: ML Training, Testing, and Validation Sets Comparison

Variable	Training	Testing	Validation
Enumerator Count	16.44	15.7	16.04
Object Detection Count	11.39	11.98	10.92
Observations	746	161	161

Step 4: Apply the Nadaraya-Watson Algorithm to the Object Detection Predictions

We then fit a Nadaraya-Watson model to the training data (Nadaraya, 1964; Watson, 1964). This algorithm is very similar to the K-Nearest Neighbours algorithm, which we also implement in Appendix 5. See Appendix 4 for details on these algorithms. We present the results for the validation set in Table 4. The results for the testing set can be found in Appendix 6.

Table 3: Nadaraya-Watson Regression Results

Group (Based on Enumerator Counts)	NW Prediction Mean	Enumerator Mean	Difference[11]	R^2	Observations
0	0.17	0.00	0.17	NA[12]	35
0-5	2.29	2.43	-0.14	0.87	40
5-10	6.73	6.82	-0.09	0.97	28
10-25	15.21	15.26	-0.05	0.99	27
25+	60.79	60.71	0.08	0.98	31
Overall	**16.03**	**16.04**	**0.01**	**0.99**	**161**

From Table 3, we can see that the algorithm is able to predict the enumerators' cherry counts with very little error. After running basic t-tests, there are no statistical differences between the predictions and the enumerator counts at any level. The R^2 terms indicate the algorithm's ability to capture most of the variation in the dependent variable, and the low error terms reiterate the accuracy of this method. We saw that the basic object detection algorithm over-predicted for low number of cherries and under-predicted for high numbers of cherries. We see no evidence of this trend with the Nadaraya-Watson algorithm. There are low sample sizes for most groups, but overall (with 161 observations) the algorithm can predict the enumerator count within 0.01 cherries. We run the same analysis with a basic KNN regression, and see very similar results (see Appendix 5).

Discussion and Next Steps

Caveats and Improvements to the Image-Based Approach

The AI approach is an effective tool in counting cherries with very little difference between the model counts and enumerator counts. However, the algorithm is only able to be successful with a second layer of modelling – the ML approach to correct for errors. This second layer requires ground-truth values of enumerator cherry counts to predict the number of cherries from an image. While we already have a database of images and ground-truth cherry counts which will aid in future deployment of the model, different varieties of coffee may need more enumerator counts for calibration.

The object-detection model (without the second layer of error correction) can be improved by increasing the number of training images. Currently, the model is trained on 218 images with most of the images containing a small amount of cherries. The object-detection algorithm performs worst with images with large numbers of cherries. Increasing the number of training images requires a human to manually mark each cherry in each new photo. Further, training images on photos taken with the same camera as our study images may improve performance.

Finally, the largest drawback to the image-based approach is the time involved on the field. In order to obtain this data, enumerators have to perform a geo-trace of the entire farm, count all of the trees in the farm, sample about 20 trees on average, count their branches, and take photos of each branch. These activities require enumerators to work in pairs, increasing budget costs. Further, the time spent taking photos was higher on average than the time spent counting cherries manually.

[11] We tested the significance level of the differences using t-tests. $p < 0.01$: ***, $p<0.05$: **, $p<0.1$: *

[12] The algorithm predicted the same value, 0.167, for each observation. There is no variation in the predictions or the enumerator counts (which are all 0), so an R^2 value cannot be calculated. For the same reason, t-statistics can also not be found.

This last point can be resolved by switching the object detection algorithm from a picture-based method to a video-based method. While relatively easy to create from a coding perspective, a video-based approach brings its own set of challenges. Standardization of videos is more difficult than that of images, and increased data requirements make implementation on the field more difficult – especially in low-to-no Internet situations.

Next Steps

Thus far, we have conducted two surveys (the main Kenya Yield Survey in June 2018 and the Imaging Study Survey October 2018), established proof of concept for AI to count cherries, shown the issues with farmer recall data, and proven that AI is as effective as enumerators in counting cherries. The final component of the study is creating a model to predict coffee yields based off of household data, farm-level data, tree-level data, and climate data. We are collecting cooperative delivery data in August 2019. This aspect of data collection has been delayed due to the late payments of cooperatives to the farmers. Once this data is collected, the modelling phase will begin, and we will attempt to predict coffee yields. At that point, we will have evidence as to whether this approach is suitable for measuring yields.

References

Hall, P., Racine, J., & Li, Q. (2004). Cross-validation and the estimation of conditional probability densities. *Journal of the American Statistical Association*, *99*(468), 1015-1026.

Li, Q., & Racine, J. (2004). Cross-validated local linear nonparametric regression. *Statistica Sinica*, *14*(2), 485-512.

Nadaraya, E. A. (1964). On estimating regression. *Theory of Probability & Its Applications*, *9*(1), 141-142.

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).

Watson, G. S. (1964). Smooth regression analysis. *Sankhyā: The Indian Journal of Statistics, Series A*, 359-372.

Appendix 1

Tree and Stem Sampling

The survey will then ask you to walk through the farm (just like the tree counts last time) and sample every n trees starting with tree m. For example, it may ask you to sample every 5 trees, starting with the 3rd tree. So, you will sample the 3rd, 8th, 13th, 18th… etc. trees until you have completed the required number of trees given by the survey. You may be asked to sample between 15 and 24 trees in total. It is of the utmost importance that you sample the correct trees because the study relies on random samples of trees, and if we selectively sample, then our results will be skewed.

Once you arrive at the correct tree, you will enter the number of main stems on the tree. You will be asked to sample one of the stems. You and your partner should label the stems from left to right such that the left-most stem is 1. This way, the sample can be truly random, which is required for our study.
The diagram below demonstrates how you should walk through the farm. It is exactly the same as when you conducted the tree counts in the previous study.

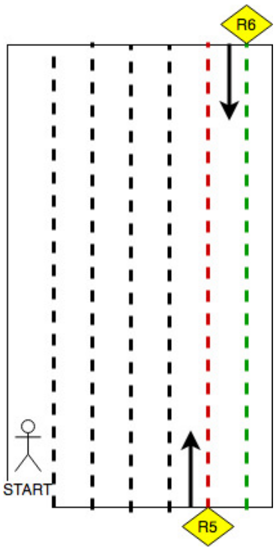


Figure 6: Diagram of Tree Sampling Approach

Appendix 2

Branch Counting and Sampling

The following is taken from the training manual for the Imaging Study:

Branch Counting

After you select the randomly assigned stem, you are asked to count the main branches starting from the bottom of the stem to the top. We are counting main branches, NOT sub-branches. So, please make sure you are counting the branches from where they are attached to the stem, not from the outside of the tree. You and your partner will have ribbons that you should place on every 5th branch. Using the ribbons is important so that if you lose count, then you can return to a checkpoint instead of starting over completely. The ribbons will also be important when you are asked to sample branches because the ribbons will make the branches easier to locate. Once you have completed counting the main branches, you will enter the number into the survey.

You will be given a counter to assist you in counting.

Cherry Counting

The survey now will now tell you how many branches are to be sampled from the tree. The survey will assign a branch sample size of 1, 2, or 3. Then, the survey will tell you which branch numbers to sample (drawn from the number of branches you counted). Note that many branches have sub-branches and we want to count the cherries on ALL of the sub-branches of the sampled branch. To make the counting easier, you may enter the number of sub-branches into the survey. You will then count all of the cherries on the branch, counting sub-branch by sub-branch and enter the counts into the survey by sub-branch by sub-branch.

Please only count green and red cherries. The brown cherries will not be harvested and should not be included in our yield estimates. It is easiest to count the cherries by using your hands to gently touch them, as you cannot see all of the cherries. Use the counter as an aid so you don't lose count. If the cherries are grouped, going group by group is a good way to ensure you have a checkpoint if you do lose count.

You and your partner should count the cherries simultaneously if there is more than one branch sampled. The enumerator with the tablet can enter the counts directly into the tablet. The other enumerator should use the provided notepads to write down the cherry counts. Be sure that you are entering the cherry count that corresponds to the branch sampled in the survey.

Again, it is absolutely imperative that you sample the branches provided by the survey. If you selectively choose branches (i.e. branches that look easier to count or have less cherries), then the results of the study will be skewed.

Also, please be careful to not knock any cherries off the branches. This will skew the study results as well, but more importantly will hurt the farmer.

Appendix 3

Branch Imaging

The final element of the survey is taking a picture of the sampled branches. You will take a photo of each branch for which you counted cherries. Make sure that the picture you are uploading is for the correct branch. The survey will indicate which branch number you are supposed to photograph and upload.

Each team will be provided with two poster boards. One is firm, and the other is flimsy. One person should place the poster board behind the branch to be photographed and the other person will take the photo. It is recommended, but not necessary, to use the stiff poster board for branches on the bottom of the tree and the flimsy poster board for higher up branches because higher branches tend be closer together. This will become apparent in the field.

Because some branches have many sub-branches and can be quite large, you may take up to 3 photos of each branch. We encourage you to take as few photos as possible, but image quality is the most important factor.

The photographer should check the photo after taking it. The photo should include ALL cherries in the branch, have only the poster board as the background, and have sufficient lighting. If you are taking more than one photo of the branch, then make sure that each photo contains a unique set of cherries. In other words, the combination of photos should contain all cherries, but never have the same cherry included in more than one photo.

Appendix 4

K Nearest Neighbors Regression

The K-Nearest Neighbors (KNN) algorithm is a non-parametric algorithm used for regression and classification, first proposed by (INSERT SOURCE). Since our outcome (number of cherries) is continuous, we use KNN for regression purposes. For simplicity in the explanation, we will use only training and test sets, but the logic is the exact same for the validation set.

The algorithm is attractive for it's intuitive logic. For each observation in the test set, the algorithm calculates the Euclidean distance[13] of the independent variable value to each independent variable value in the training set. The algorithm then picks the *k* nearest neighbors – the *k* points in the training set with the smallest Euclidean distance to the point in the test set. In our case, we are trying to predict the number of enumerator-counted cherries from the number of cherries counted from the object detection algorithm. So, the algorithm calculates the distance between the number of algorithm-counted cherries in the test set to the training set and picks the *k* closest observations.

After the *k* neighbors are selected, the algorithm takes a mean of the dependent variable values and that is the prediction. In our case, the mean of the enumerator-counted cherries of the *k* nearest neighbors is the prediction for enumerator-counted cherries for the test set observation.

We select the optimal value of *k* by empirical inspection. For each value of k from 1 to 10, we train the model and test it on our test data. We simply use the value of *k* with lowest root mean squared error (RSME) as our *k* for the final model. This value is 4 in our case. Intuitively, a *k* that is too low is conducive to over-fitting, but a *k* that is too high creates a model with low predictive power.

Finally, an advantage of this non-parametric model over linear models is that the model will not predict infeasible values (e.g. negative values).

Nadaraya-Watson Regression

The Nadaraya-Watson Regression follows the same intuition as the KNN regression. It was proposed by Nadaraya (1967) and Watson (1967) and is sometimes called a 'kernel' regression. However, instead of simply averaging over the values of each nearest neighbor, the Nadaraya-Watson algorithm assigns weights to each neighbor. The weights are assigned such that neighbors that are closer to the observation receive more weight than ones that are further away. The weights should improve prediction performance.

[13] The Euclidean Distance formula for the distance between points *i* and *j* is: $d_{ij} = \sqrt{(p_i - p_j)^2}$

The weights are determined by a mathematical function called a kernel weighting function. The goal of the function is assign the optimal weights to each neighbor. We use a Second-Order Gaussian kernel weighting function. The formulation of this function is beyond the scope of this report.

Finally, each kernel requires a bandwidth selection. We can think of the bandwidth as the domain over which to search for neighbors. A wider bandwidth means more neighbors are used, while a smaller bandwidth means fewer neighbors are used. Finding the optimal bandwidth is crucial for algorithm performance, as we want accurate results without over-fitting. This is similar to the KNN problem of finding the right number of neighbors to include in the regression. To find the optimal bandwidth, we employ the methods of Racine and Li (2004) and Hall, Racine, and Li (2004). We do not delve into the details of this algorithm here, but only mention that we are computing an optimal bandwidth rather than choosing an arbitrary one.

Appendix 5

We present the results of the KNN regression for the validation set below in Table 4. We can see that the KNN model performs slightly better on branches with low number of cherries, but the Nadaraya-Watson algorithm far outperforms KNN on branches with over 25 cherries. The stronger performance on branches with larger numbers of cherries drives the overall performance of the Nadaraya-Watson algorithm to better than that of KNN. However, these differences are quite small and the KNN's strong performance strengthens the argument that this analytical approach to counting cherries is viable.

Table 4: KNN Regression Results

Group (Based on Enumerator Counts)	KNN Prediction Mean	Enumerator Mean	Difference[14]	R^2	Observations
0	0.00	0.00	0.00	NA	35
0-5	2.48	2.43	0.05	0.90	40
5-10	6.79	6.82	-0.03	0.94	28
10-25	15.25	15.26	-0.01	0.99	27
25+	59.97	60.71	-0.50	0.91	31
Overall	**15.90**	**16.04**	**-0.14**	**0.96**	**161**

[14] We tested the significance level of the differences using t-tests. p < 0.01: ***, p<0.05: **, p<0.1: *

Appendix 6

We present the results of the Nadaraya-Watson and KNN algorithms for the test set below. The results for the test set are similar to the results for the validation set with the algorithms performing with little error. As in the case for the validation set, the Nadaraya-Watson algorithm performs better overall, largely driven by its superior performance on branches with large numbers of cherries. Again, the results presented below show that our model is able to accurately count cherries.

Table 5: Nadaraya-Watson Testing Set Results

Group (Based on Enumerator Counts)	NW Prediction Mean	Enumerator Mean	Difference[15]	R^2	Observations
0	0.17	0.00	0.17	NA[16]	34
0-5	2.08	2.24	-0.16	0.87	34
5-10	6.58	6.68	-0.10	0.96	22
10-25	16.27	16.32	-0.05	0.99	38
25+	52.13	52.16	-0.03	0.99	32
Overall	**15.67**	**15.70**	**-0.03**	**0.77**	**161**

Table 6: KNN Testing Set Results

Group (Based on Enumerator Counts)	KNN Prediction Mean	Enumerator Mean	Difference[17]	R^2	Observations
0	0.03	0.00	0.03*	NA	34
0-5	2.24	2.24	0.00	0.75	34
5-10	6.87	6.68	0.19	0.87	22
10-25	16.28	16.32	-0.04	0.96	38
25+	48.67	52.16	-3.49	0.66	32
Overall	**15.02**	**15.70**	**-0.68**	**0.77**	**161**

[15] We tested the significance level of the differences using t-tests. p < 0.01: ***, p<0.05: **, p<0.1: *

[16] The algorithm predicted the same value, 0.167, for each observation. There is no variation in the predictions or the enumerator counts (which are all 0), so an R^2 value cannot be calculated. For the same reason, t-statistics can also not be found.

[17] We tested the significance level of the differences using t-tests. p < 0.01: ***, p<0.05: **, p<0.1: *