Classification of Tree Presence in a Region at a Per Pixel level

A comparison of the effectiveness of Support Vector Machines and Random Forest classifiers for tree detection

University of Cape Town
CSC2005Z Research Report
Luc Hayward – HYWLUC001
Supervisor: Prof. Patrick Marais

ABSTRACT

The abundance of relatively low-cost consumer drones on the market has opened up new opportunities for high quality aerial scans to be captured. Analysing these scans by hand is both time-consuming and error prone. This report examines the viability of using machine learning models to generate bitmasks representing the location of trees in an image using a combination of colour photography and near infrared data. Support Vector Machines and Random Forest classifiers were chosen as they are popular for use in binary classification scenarios such as this. A combination of analytical tests and qualitative comparisons against manually generated ground truths were used to evaluate the effectiveness of the classifiers. It was found that the Random Forest classifier produced the most accurate results without compromising on training time for the model.

1. Introduction

Consumer drone technology, such as that from DJI [1], has advanced to the point where extremely high quality aerial photography is achievable without requiring the use of micro-planes or helicopters as were once relied upon. This has allowed companies such as Aerobotics [2] (who provided the data for this study) to provide detailed high quality aerial surveyance of farms and crops in order to produce new insights into crop health. However, in order to further analyse the aerial scans however, a mask must be drawn marking the areas on the scans in which trees are present. Currently this has largely been done by hand, a time intensive and error prone task as each scan may have hundreds of individual trees of varying sizes and shapes.

This research aimed to evaluate the viability of creating a machine learning classifier which could create the tree masks from the scans and remove (or significantly reduce) the need for human intervention in the process. Specifically, Support Vector Machines and Random Forest classifiers were to be compared as two of the most popular binary classifiers in use.

The current process for creating the tree masks has required Aerobotics to manually draw the masks for the trees by hand. This is not only a slow process but has resulted in large differences in the quality and uniformity of the masks created. An Artificial Intelligence (AI) classifier would be able to produce results that are far more uniform and better make use of all available data as the depth maps and Normalised Difference Vegetation Index (NDVI) data are not as easily understood as the colour imagery by humans. This paper aims to achieve and evaluate the following:

Produce a model which, given the scan data, will produce an accurate mask of tree locations in the image.

The mask produced by the classifier should be a binary mask of the image which classifies the location of trees in the image at a per pixel level. The mask should be at least 80% accurate relative to the best manually produced masks provided by Aerobotics using the Intersection over Union method [3]. Both classifiers will be trained on the same dataset and evaluated against the same test data so one model should consistently produce better results than the other. Finally, the classifier should not require an excessive amount of time to produce results after having been trained.

2. Background and Related Work

Classification using Machine learning and AI has been the subject of a vast amount of research due to its applicability in many different areas of computer science research. Classifications models are required in fields ranging from image recognition and object detection to spam filters and loan applications. As such the vast majority of the work required to produce the models used in this paper and show their viability has already been completed. SVM Random Forest classifiers have both been shown to produce good results when used for binary classification and are more than capable of handling the relatively low dimensionality of the datasets in use [4].

Random Forest classifiers are conceptually far simpler than SVM classifiers as well as requiring few user defined parameters than SVMs. This results in a model which is easier to optimise and requires potentially less tuning of the parameters than an equivalent SVM model. Despite this relative simplicity the Random Forest classifier continues to be widely used and produces results which are as accurate as those produced by SVMs with comparable training times. [5]

Three previous studies on agricultural classification models have also looked at the potential for both SVM and Random Forest models although the most relevant to this study, a paper which also examined the use of NDVI data was unfortunately not freely available for comparison [6]. However, the most cited and widely discussed paper in this area examined the difference between pixel-based and object-based analysis using a range of different machine learning algorithms. The authors made a compelling case for the use of Random Forest and SVM for crop classification which should extend well to simple classification of tree presence. The data from this study suggests that there is little positive impact of complex object-based analysis over the simpler pixel-based approach for either style of classifier. Furthermore, the pixel-based approach used fewer variables without impacting the accuracy of the classifications. [7]

SVM Classifiers: This style of classifier is able to be used both for classification and regression. For classification each data item is plotted as a point in n-dimensional space and a hyperplane is fitted that best separates the two classes of data.

New data can the efficiently be classified based on which side of the hyperplane it falls when plotted in the n-dimensional space.

Random Forests: This style of classifier is conceptually simple but very effective in classifying data into different classes without compromising on time taken to train or classify new data. A random forest model generates many different decision trees when being trained. When producing a classification for a given data item, each decision tree 'votes' for a particular class and the results of all the trees are calculated. The class with the most 'votes' is 'chosen' as the class into which that data item should be classified. Random forest models can suffer from overfitting however, if the training data is not diverse enough although this can be alleviated by limiting the depth to which a single tree is grown.

The results of (Raczko and Zagajewski)[8] suggested that the SVM classifier may produce slightly more accurate results the Random Forest approach however the addition of the NDVI data may change the relative performance of each classifier. Previous works have also shown that tuning the parameters for an SVM model is often required for optimal results depending on the data. Furthermore, the quality of the ground truth directly impacts the ability to analytically compare the accuracy of the models. This inconsistency would reduce the significance of the 6% difference in overall accuracy found by Raczko & Zagajewski, were those similarities to arise in this study.

The data provided by aerobotics for this study was produced using consumer DJI drones fitted with high resolution cameras and near Infrared sensors which were used to produce the NDVI data (albeit at lower resolution than the colour images). NDVI data has been shown to closely correlate with the presence and density of vegetation in an area [9]. By combining the image data with the NDVI results it is thought that a more robust classifier would be created.

3. Project Overview

This project examined the relative accuracy of SVM and Random Forest classifiers for producing masks of tree presence in an image using colour and NDVI data as input. Using Python and the Scikit-Learn frameworks, two separate models were created, trained and tested on real world data. The results were compared experimentally to determine the most effective approach in this scenario.

4. Methodology

Here we review the reasoning behind the different algorithms, frameworks and programs used throughout the project. Additionally, the methods for creating and evaluating the models as well as to create the models and handle the raw data provided are examined.

The Random Forest and SVM classifiers were chosen due to their ease of use and long track record of producing effective results in similar areas. Additionally, both SVM and Random Forest models are easily created using Python[1] and the Scikit-learn[2] frameworks which enabled rapid iteration on design and parameter tweaking without the need to create the models from scratch.

By using Scikit-learn a far more performant model could be used which could run in any environment capable of running a python script. The framework provides methods for model persistence, easy parameter tuning and libraries for implementing cross validation effectively and efficiently in order to ensure that the models could be trained on a diverse set of data as well as allowing for parameter tuning that did not simply overfit to a particular subset of the data and would be unusable in new environments.

In addition to Scikit-learn for the machine learning models; the numpy[3], matplotlib[4], skimage[5] and Pillow[6] libraries were all used as well as the QGIS software[7].

Numpy is a highly used maths library used for various data science and machine learning applications in python thanks to its matrix math support and high performance [10]. This allowed for easy manipulation of the image data once it had been read into the program using skimage and pillow, add or remove the data from different scans convert the data to and from a format which can be used by the machine learning models.

Matplotlib was initially used to both read and write the scans and bitmasks respectively however, when testing the use of the NDVI data in conjunction with the RGB colour images, matplotlib was unable to read the NDVI data as a 32bit grayscale value and would attempt to convert these to 4-channel 8bit images. This resulted in unusable data and necessitated finding a new image library, skimage and pillow.

Finally, the QGIS software is an open source geographic information system capable of being run on Windows and Mac (as well as Linux) which provided an easy method for exploratory data analysis and categorisation of scans before training and testing.

Aerobotics provided sixty-two (62) real world scans from past surveys across a range of different farms with different tree sizes, types and densities. These different characteristics resulted in three "categories" or types of farms in the dataset, A, B and C.

Figure 1. The three different styles of farm/crop

This allowed for a range of different scans to be trained on to help reduce the amount of overfitting and allow the resultant models to remain effective in the future. Unfortunately, many of the scans provided had accompanying masks (which would be the "ground truth" against which to compare the models results) which were so inaccurate as to be entirely unusable with the masks often having almost no relation at all to where the trees actually were in the images (see Figure 2 below).

[1] [19]
[2] [16]
[3] [11]
[4] [15]
[5] [17]
[6] [12]
[7] [18]

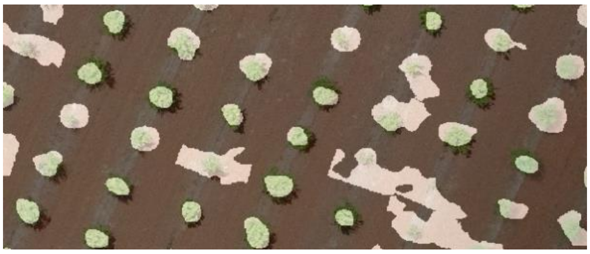


Figure 2. An example of the incorrectness of many ground truth masks (unused)

Others, though not quite as bad, were deemed too inaccurate for the Intersection over union tests as large areas of the mask appeared to have been "shifted", indicated trees where the actual tree was slightly off center with the mask. This would have resulted in either the models learning incorrect classifications or, if used for testing, even a perfectly accurate model would appear to be making mistakes as the ground truth was incorrect (see Figure 3).

Figure 3. An example of an almost accurate ground truth mask wherein the mask appears to be shifted (unused)

This left 20 high quality scans against which the model could be trained, tested and evaluated. These "good" scans covered each of the three categories however farm type A dominated with only one scan each of types B and C. Although the model was trained on these in its final training, the data related to those scans would not be ideal due to the lack of variation available.

Before the data could be used to train the models, it first needed to be read in and converted to a format useable by scikit-learn. After the scans were read in from the survey folder, each of the features were separated out from the input image data into separate arrays. This allowed them to be easily manipulated before joining them into an array of data points for an image and passed to the model for training or classification.

After separating the 5 features from the input scan, the RGB channels were left unaltered whilst the depth map (from the alpha channel of the colour image) and the NDVI data were resized to fit the shape of the RGB data. There should be no issues of artefacting occurring as the RGB data is both the highest resolution image in all the scans provided as well as always being the same aspect ratio as the other features. This was done to preserve as much of the details as possible, rather than down sampling the RGB data to fit the often-lower resolution NDVI data.

There was hope that combing all 5 features for each data point in order to supply the models with the maximum amount of data would produce the most optimal results however it quickly became apparent during validation and the process of tuning the parameters that the depth map data was too inconsistent between different datasets to provide useful information and so was not used in the final training or testing of the models.

In order to train the models for the final time after the validation and parameter tuning phase, 5 scans were selected at random from the "good" data. From each of these 5 scans, a spread of 100 "tree present" and 100 "tree not present" data points (pixels containing RGB and NDVI data) were selected from random locations in the image. By spreading the training data across several scans, it ensures a variety of tree types are present in the data with an equal mix of true and false values. This reduced the chance of any overfitting in the final models and matches up with the methods employed during validation.

The resultant masks produced by both models, whilst appearing accurate to the eye, were irregular and noisy particularly around the edges of trees where individual leave and branches would be separated out rather than creating a more regular bounding volume for the tree.
In order to smooth the boundaries of trees in the mask and reduce the noise caused by grasses and other shrubbery (which should be ignored for the final mask), a median filter with kernel size of 10 was applied across the initial mask with the final values rounded in order to clamp the results to 0/1. Median filters are particularly suited to this type of "salt and pepper noise" where the filter is intended to reduce the speckling affect in the image without affecting details too heavily [10]. Though some details may be lost due to the relatively high kernel size, the smoothing effect on the boundaries was preferable to the noise before filtering.

Evaluation of the results was done quantitatively using a confusion matrix and calculating the intersection over union, a method of evaluating the overlap between the predicted mask and the ground truth mask (with higher scores indicating a better agreement in the classifications).

The intersection over union tests were performed on three different scans which were held out and used only for the final test (these three were randomly selected from the good portion of the data before beginning work on the models). The results from these three scans were then averaged together to produce the final score for both the classifiers. The IoU test requires that a confusion matrix be generated detailing the proportion of the predictions which are true and false positives and negatives. "A confusion matrix contains information about actual and predicted classifications done by a classification system." [3] An IoU score ranges from 0 to 1 with 1 being a perfect recreation of the ground truth classification and zero being no similarities between the classifiers at all.
The formula for calculating the IoU score is given below along with an example confusion matrix:

$$\frac{TP}{TP + FP + FN}$$

Table 1. Example confusion matrix

n=127	Predicted: Not-Tree	Predicted: Tree	
Actual: Not-Tree	TN = 36	FP = 14	50
Actual: Tree	FN = 12	TP = 65	77
	48	79	

In the context of this study, in which sampling was done at the per pixel level, each value in the confusion matrix represents a number of pixels. Thus the "Predicted: Tree" value is the number of pixels in the scan which were predicted by the classifier to contain a tree whilst the "Actual: Not-Tree" value is the number of pixels where there is no tree in the ground truth.

Some standards and terms:

1. True positive/negative (TP/TN): Where the ground truth and the classifiers predictions match.

2. False positive/negative (FP/FN): Where the classifier incorrectly predicts the value of a pixel compared to the ground truth value.

3. Accuracy: A measurement of how often the classifier is correct as a value from 0 to 1. Higher accuracy is always better. $(TP + TN)/total$

4. Sensitivity and Specificity are measures of how often it classifies as true when it is actually true and how often it classifies as false when it is actually false respectively. Values closer to 1 are better.

5. Results and Discussion[8]

During the validation and parameter tuning phase it was initially thought that providing the maximum amount of data to the models would yield the best results. However as mentioned in the Data Processing section, the depth map proved to be inconsistent and merely resulted in inaccurate, noisy masks. It is thought that the variation in height as well as topology for the different farms did not allow for the models to learn anything useful from the depth data that could be applied to scans on which it had not already been trained as the artefacts were far less common if the models were trained and rerun on the same data again. Removing the depth data from the feature set largely reduced the noise present in the predicted masks as well as improving perceived consistency in different environments.

All testing was performed on a mid-2015 MacBook Pro with an i7-4770HQ @2.2GHz with 8GB RAM.

It is immediately clear that both the Random Forest classifier and the SVM classifier both produce results which exceed the initial expectations of 80% accuracy in the majority of the tests.

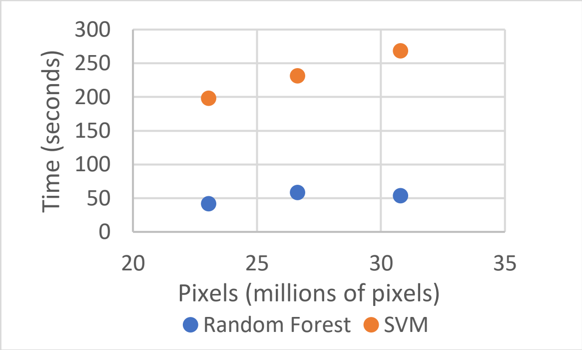
Table 2. Comparison of averaged results of test scores for the three scans using Random Forest and SVM classifiers

	Random Forest	SVM
IoU TP/TP+FN+FP	0.845	0.849
Accuracy (TP+TN)/total	0.948	0.949
Sensitivity (TP/actual P)	0.903	0.919
Specificity (TN/actual N)	0.974	0.967
Precision (TP/predict P)	1.076	1.092
1000 Pixels/sec	457.836	100.618

The averaged results of the three scans (making up over eighty million pixels in total) show a strong IoU indicating that each classifier was able to produce accurate results that were comparable to each other to within a margin of error. In all measurements of "correctness" tested, each classifier produced very similar results with neither showing statistically significant differences between their results.

One interesting observation that would become clearer with more reliable test data in that the SVM tended towards higher sensitivity and lower specificity than the Random Forest. This showed that the SVM tended to over classify pixels as being positive for tree presence while the Random forest was more conservative, preferring to over classify pixels as not containing a tree.

In spite of the close results produced by both classifiers, the Random forest pulls ahead significantly in classification speed as can be seen in Table 2. **Comparison of averaged results of test scores for the three scans using Random Forest and SVM classifiers** The Random forest classifier performed almost five times faster than the SVM taking less than a minute for each of the three scans whilst the SVM took over 4 minutes in one instance to produce a result. This could quickly compound in larger scans as the time required appeared to grow far faster for as resolution increased for the SVM.



[8] The full tables of results are available in the appendix

Figure 4. Comparison of time taken to classify as resolution grows

In addition to the analytical tests performed above, the results of the scans were overlaid using a difference filter in order to visualise the differences between the scans themselves and the original masks. These overlaid difference images supported the observation that the SVM tended to classify more of a region as containing trees which could be seen from its higher sensitivity scores. This could be seen in Figure 5 by the white borders around the regions containing trees.

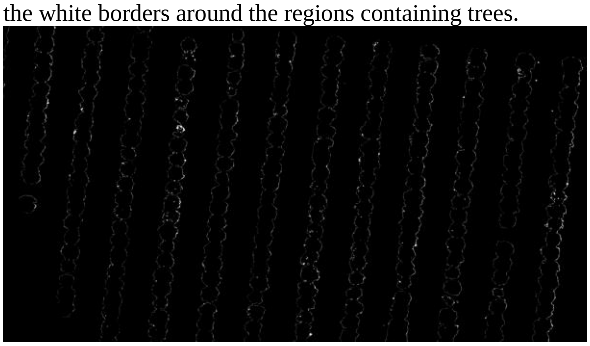


Figure 5. SVM mask difference overlaid on the RF mask. White indicates difference in the masks.

Similarly, this technique can be applied to the ground truth mask to evaluate possible reasons for the imperfect IoU scores. It quickly became clear that in spite of the careful binning of scans to only include those with the most accurate ground truth masks, not all areas accurately reflected the size and shape of every tree. Many of the smaller, less developed trees were "drawn" in the mask provided as full size, regularly shaped trees. However, both the Random Forest and SVM classifiers were able to accurately mask out the tree in their smaller size, preserving the irregular shape (see Figure 6). These differences are likely to have compounded to result in the lower IoU scores seen in scan 2 (see central appendix column) compared to the other two tests scans.

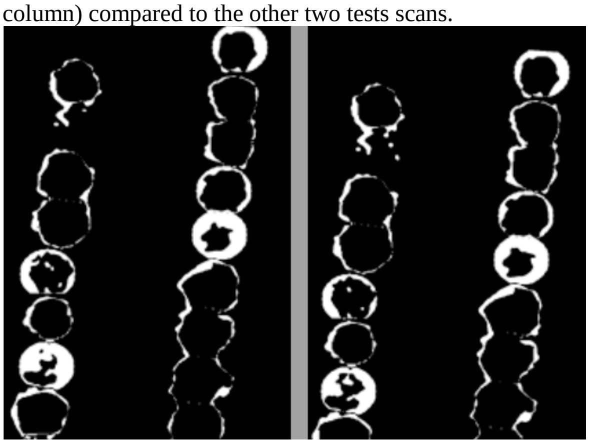


Figure 6. Predicted masks overlaid using a difference filter on the ground truth mask. Left: Random Forest, Right: SVM

6. Limitations

There are a couple of limitations on the results of this study with regards to the availability of quality data and the lack of time in which to obtain more qualitative data from others who would be familiar with the field and the data in question.

Though the data provided covered a range of crop types and landscapes with hundred of trees in each sample, the viability of the ground truth scans was lacking in the majority of the scans. This could be addressed in future work by spending the time to hand draw more accurate ground truth masks for those scans. This would allow for further analytical tests to be completed on different crop types.

Furthermore, the quality of the ground truth masks affected the ability to accurately test the correctness of the predicted masks. Even with the noise reduction filters applied, the masks representing the ground truth against which the predicted masks were compared did not perfectly reflect the extent of tree in the image. Often the hand drawn mask would over or under approximate the size of the tree canopy, including ground in the mask or leaving out the very edges, whilst that area may have been correctly classified by the ML models (see Figure 7 below).

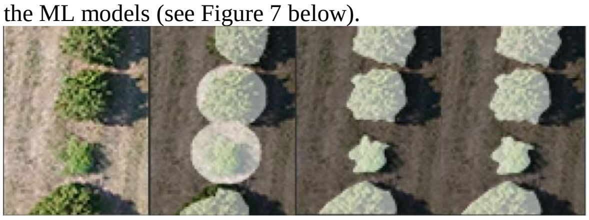


Figure 7. Comparison of Tree Masks overlaid at 50% opacity on original image. (From left to right; no mask, ground truth, Random Forest, SVM)

7. Conclusions and Future Works

This project sought to evaluate the effectiveness of SVM and Random Forests models for automatic generation of tree masks from scanned data. Initial visual comparisons suggested that both models were able to outperform the typical standard that had been deemed acceptable for the hand created masks.

Further experimental tests showed that the initial visual perceptions were correct and that both SVM and Random Forest classifiers produced excellent results. By comparing the results of both classifiers, it is clear that in the current use case, both models produce masks which are quantitatively similar to within a margin of error. However, the Random Forest model trained faster and produced high quality results in significantly shorter amounts of time. This makes it a far better solution as the rate of data collection and the resolution of that data is only likely to increase in the future.

There is much potential with regards to future work. While this report can be viewed as a proof of concept for the use of machine learning to classify tree presence it cannot represent a definitive best practice. It is not clear whether the classifiers would continue to perform optimally if the style of data (the farms) were to shift drastically.

Two possible avenues for further research in this area could be to evaluate other machine learning techniques or to examine additional techniques looking at identification of structures as opposed to per-pixel classifications. Neural networks would be one such additional technique that could be examined. Alternatively, there is the possibility of further research looking at spatial features rather than single pixel data. At this point many different paths could be examined such as using edge detection models to estimate the boundary of the trees before classification.

8. Acknowledgements

Many thanks to Patrick Marais for the guidance throughout this research project and to Aerobotics for generously providing all of the data needed to train and evaluate the models. Additional thanks to Jarrod Finnis and Ntokozo

Motsumi for their help in understanding the ground truth masks.

8. References

[1] DJI, “DJI - The Future Of Possible,” 2018. [Online]. Available: https://www.dji.com. [Accessed 11 11 2018].

[2] Aerobotics, “Aerobotics,” 2018. [Online]. Available: https://www.aerobotics.io. [Accessed 31 August 2018].

[3] T. R. Patil and S. S. Sherekar, “Performance Analysis of Naive Bayes and J48 Classification Algorithm for Data Classification,” *International Journal Of Computer Science And Applications,* vol. 6, no. 2, pp. 256-261, 2013.

[4] M. Sonka, V. Hlavac and R. Boyle, Image processing, Analysis, and machine learning, 4th ed., Cengage learning, 2014.

[5] M. Pal, “Random forest classifier for remote sensing classification,” *International Journal of Remote Sensing,* vol. 26, no. 1, pp. 217-222, 2005.

[6] J. K. Raj and S. SivaSathya, “SVM and Random Forest Classification of Satellite Image with NDVI as an Additional Attribute to the Dataset,” *Proceedings of the Third International Conference on Soft Computing for Problem Solving,* vol. 258, pp. 95-107, 2014.

[7] D. C. Duro, S. E. Frnaklin and M. G. Dubé, “A comparison of pixel-based and object-based image analysis with selected machine learning algorithms for the classification of agricultural landscapes using SPOT-5 HRG imagery,” *Remote Sensing of Environment,* vol. 118, pp. 259-272, 15 March 2012.

[8] E. Raczko and B. Zagajewski, “Comparison of support vector machine, random forest and neural network classifiers for tree species classification on airborne hyperspectral APEX images,” *European Journal of Remote Sending,* vol. 50, no. 1, pp. 144-154, 2017.

[9] T. N. Carlson and D. A. Ripley, “On the relation between NDVI, fractional vegetation cover, and leaf area index,” *Remote Sensing of Environment,* vol. 62, no. 3, pp. 241-252, 1997.

[10] S. Jayamaran, S. Essakkirajan and T. Veerakumar, Digital Image Processing, Tata McGraw-Hill Education, 2009, p. 272.

[11] NumPy, “NumPy,” 2018. [Online]. Available: www.numpy.org. [Accessed 04 09 2018].

[12] Pillow, “pillow: The friendly PIL fork,” 2018. [Online]. Available: https://python-pillow.org/. [Accessed 11 11 2018].

[13] C. J. Tucker, “Red and photographic infrared linear combinations for monitoring vegetation,” *Remote Sensing of Environment,* vol. 8, no. 2, pp. 127-50, 1972.

[14] A. HAY, “The derivation of global estimates from a confusion matrix,” *International Journal of Remote Sensing,* vol. 9, no. 8, pp. 1395-1398, 1988.

[15] SciPy, “SciPy.org,” 2018. [Online]. Available: www.scipy.org. [Accessed 04 09 2018].

[16] scikit-learn, “scikit-learn,” 2018. [Online]. Available: http://scikit-learn.org/stable/index.html#. [Accessed 04 09 2018].

[17] scikit-image, “scikit-image: Image processing in Python,” 2018. [Online]. Available: https://scikit-image.org/. [Accessed 11 11 2018].

[18] QGIS, “QGIS A Free and Open Source Geographic Information System,” 2018. [Online]. Available: https://qgis.org/en/site/. [Accessed 04 09 2018].

[19] Python Software Foundation, “Python,” 2018. [Online]. [Accessed 2018].

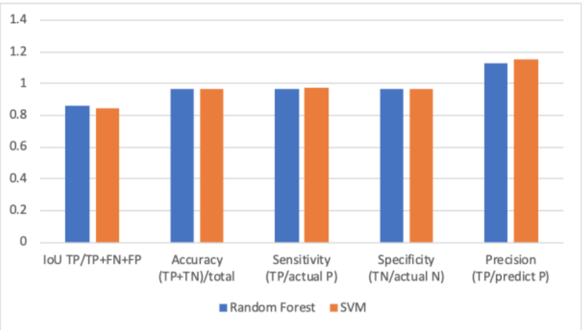
Appendix:

Scan 1

RF	Predicted: Not-Tree	Predicted: Tree	
Actual: Not-Tree	18013292	563382	18576674
Actual: Tree	150567	4333783	4484350
	18163859	4897165	23061024

SVM	Predicted: Not-Tree	Predicted: Tree	
Actual: Not-Tree	17909056	667618	18576674
Actual: Tree	117263	4367087	4484350
	18026319	5034705	23061024

	Random Forest	SVM
IoU TP/TP+FN+FP	0.858560439	0.84765414
Accuracy (TP+TN)/total	0.96904088	0.96596504
Sensitivity (TP/actual P)	0.966423896	0.97385061
Specificity (TN/actual N)	0.969672612	0.96406149
Precision (TP/predict P)	1.129997741	1.1528749
Predict time	41.90795279	198.179815
1000 Pixels/sec	550.2779894	116.364141

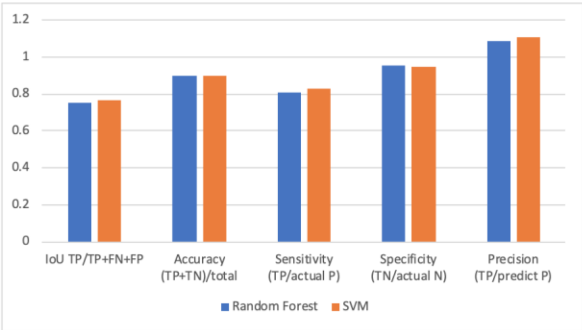


Scan 2

RF	Predicted: Not-Tree	Predicted: Tree	
Actual: Not-Tree	18081404	834476	18915880
Actual: Tree	2276343	9596087	11872430
	20357747	10430563	30788310

SVM	Predicted: Not-Tree	Predicted: Tree	
Actual: Not-Tree	17880507	1035373	18915880
Actual: Tree	2027926	9844504	11872430
	19908433	10879877	30788310

	Random Forest	SVM
IoU TP/TP+FN+FP	0.755186746	0.76267851
Accuracy (TP+TN)/total	0.898961034	0.90050448
Sensitivity (TP/actual P)	0.808266463	0.82919032
Specificity (TN/actual N)	0.955884897	0.94526435
Precision (TP/predict P)	1.086960029	1.10517269
Predict time	53.68715525	268.529466
1000 Pixels/sec	429.5445325	85.878933



Scan 3

RF	Predicted: Not-Tree	Predicted: Tree	
Actual: Not-Tree	18560821	91251	18652072
Actual: Tree	534415	7454137	7988552
	19095236	7545388	26640624

SVM	Predicted: Not-Tree	Predicted: Tree	
Actual: Not-Tree	18517477	134595	18652072
Actual: Tree	371460	7617092	7988552
	18888937	7751687	26640624

	Random Forest	SVM
IoU TP/TP+FN+FP	0.922564201	0.9377021
Accuracy (TP+TN)/total	0.976514589	0.981004386
Sensitivity (TP/actual P)	0.933102395	0.95350096
Specificity (TN/actual N)	0.995107729	0.992783912
Precision (TP/predict P)	1.012241659	1.017670129
Predict time	58.57737112	231.5123923
1000 Pixels/sec	393.6848575	99.61032225