COMPARISON OF DIFFERENT MACHINE LEARNING CLASSIFIERS FOR BUILDING EXTRACTION IN LIDAR-DERIVED DATASETS

Ivan Marc H. Escamos[1], Allen Roy C. Roberto[1], Edwin R. Abucay[1], Gillian Katherine L. Inciong[1] ,Miyah D. Queliste[1], Joseph Anthony C. Hermocilla[1]

[1] University of the Philippines Los Baños, Los Baños, Laguna, 4031, Philippines
Email: ivanhescamos@gmail.com

KEY WORDS: Object Based Image Analysis, Feature Extraction

ABSTRACT: Building extraction in remotely sensed imagery is an important problem that needs solving. It can be used to aid in urban planning, hazard assessments and disaster risk management among others. Light Detection and Ranging or LiDAR, is one of the most powerful remote sensing technologies nowadays. Many studies have used the fusion of LiDAR data and multispectral images in detecting buildings. This study seeks to maximize the power of LiDAR imagery to be able to classify buildings without the aid of multispectral imagery. This work follows the Object Based Image Analysis (OBIA) approach. Instead of the traditional pixel-based classification methods, pixels are segmented into logical groups called objects. From these objects, features for building extraction are calculated. These features are: the number of returns, difference of returns, and the mean and standard deviation of positive surface openness. These objects are then classified using different machine learning classifiers such as Support Vector Machines, K-Nearest Neighbors, Naïve Bayes Classifier, Decision Trees, and Random Forests. A comparative assessment was done on the performance of these different machine learning classifiers. The classifiers performed similarly with the Random Forest Classifier slightly outperforming the others.

1. INTRODUCTION

Building extraction in remotely sensed imagery is an important problem that needs solving. It can be used to aid in urban planning, hazard assessments and disaster risk management among others. The development of a building extraction method is one of the objectives of the UPLB Phil-LiDAR 1 project.

Light Detection and Ranging (LiDAR) is one of the most powerful remote sensing technologies nowadays. The LiDAR is an instrument similar to radar, but it uses laser pulses instead of radio waves. (Campbell, 2011) Geospatial products, like Digital Elevation Models (DEMs), can be created using LiDAR instruments on aircraft (NOAA, 2015).

Many recent studies have used LiDAR in conjunction with multispectral imagery. The different frequencies of the electromagnetic spectrum provide a lot of information helpful for building extraction and remote sensing in general. But high resolution multispectral imagery is not easily accessible. In UPLB Phil-LiDAR 1's case, orthophotos corresponding to LiDAR DEMs are available. These orthophotos have issues like misalignment with the LiDAR DEMs, areas with no coverage, inconsistency of colors due to changing atmospheric conditions, etc.

Our study follows the Object Based Image Analysis (OBIA) approach. Instead of the traditional pixel-based classification methods, pixels are segmented into logical groups called objects. (Blaschke, 2010). From these objects, features for building extraction are calculated. Features are quantities that describe an object. (Kohavi, 1998.) Machine learning classifiers, together with these features are used to classify objects.

The general objective of this study is to evaluate different machine learning methods in building detection by object based image analysis using only LiDAR derivatives. The specific objectives are as follows:

a. To identify features from LiDAR derivatives suitable for building extraction;
b. To extract buildings using different machine learning classifiers; and
c. To evaluate and compare the different machine learning classifiers

2. REVIEW OF RELATED LITERATURE

Through the years, airborne LiDAR technology has reached new heights of interest in remote sensing especially in applications such as building extraction. A number of techniques have been developed to correctly identify buildings, and these methods vary in performance and precision.

One traditional approach in building extraction is pixel-based analysis, where images are classified per-pixel. Meng et al. (2009) developed a building detection method that first removes ground using a filtering algorithm, and then further removes remaining non-building pixels by using morphological operations on the elements' size, shape, structure, height and height difference of its first and last returns. Their study, which only uses LiDAR data, yielded an overall accuracy of 95.46%. They added that fusing LiDAR data with other multi-spectral images may cause errors introduced by resolution differences, geo-referencing, time differences and high-rise building displacement problems. In 2006, Singh used a parameter called Openness in building extraction using LiDAR data of an urban area. It is a feature which expresses the degree of dominance or enclosure of a location on a surface (Yokoyama, Shirasawa, and Pike, 2002). His algorithm managed to remove most of non-ground, and non-building features from the data, and medium to large size buildings were successfully obtained in the result.

With the technology advances in remote sensing, one problem rises regarding pixel-based methods applied to higher resolution imagery. These techniques often produces "salt and pepper" effect or noises that significantly affects the inaccuracy of the classifications (Campagnolo and Cerdeira, 2007; Gao and Mas, 2008). For this reason, OBIA were introduced and now widely used by researchers in spatial analysis. In 2014, Uzar developed an object oriented image analysis with a rule-based classification method for building extraction that incorporates both LiDAR data and ortho-images. The first step of his method is segmentation, which divides the image into logical regions or objects with common properties (Navulur, 2007). Finally, rules regarding shadow, nDSM, slopes and vegetation were defined to classify the building objects. His work reported results of completeness (81.71%) and correctness (87.64%).

Other than rule-based classification, training and automatic classification of objects using machine learning classifiers were also widely used in OBIA. In 2014, Qian, along with other researchers, integrated both LiDAR data and high resolution images in classification to evaluate and compare the performances of four machine learning classifiers namely Support Vector Machine (SVM), Normal Bayes (NB), Classification and Regression Tree (CART) and K-Nearest Neighbour (KNN). He concluded that SVM and NB were superior to the other two classifiers as both achieved high classification accuracy (>90%). He also added that NB and KNN were more sensitive to the sample sizes.

This study evaluated the effectiveness of using only LiDAR data in object based image analysis classification, whereas most techniques use multi-spectral imagery. The proposed methodology focused on separating buildings from non-building objects and different machine learning algorithms were tested.

3. METHODOLOGY

3.1 Study Area and Data

The study area is located at the municipality of Pagsanjan, Laguna, Philippines in between longitudes 121.451° and 121.461° and latitudes 14.268° and 14.277°. It encompasses the barangays Maulawin, Barangay I, Barangay II, and Pinagsanjan. The area has mixed land cover, with built-up, forested, and agricultural areas. Noticeably, it is trisected by the famous Pagsanjan River. (Figure 1) We used LiDAR data from DOST and UP Diliman's Disaster Risk Exposure and Assessment for Mitigation (DREAM) Program's Data Acquisition Component. The LiDAR Data was in .las format. (Figure 2)

Figure 1. Orthophoto of Study Site

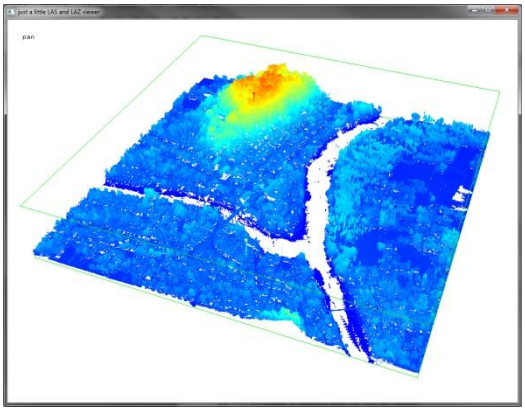

Figure 2. LiDAR Data of the Study Site visualized using LASView

3.2 Software and Technologies Used

ESRI's ArcMap was used primarily to visualize the different data in an interoperable fashion. LiDAR derivatives were generated using the LAStools software package by rapidlasso (rapidlasso, 2015). Trimble's eCognition was used for image segmentation. SAGA GIS, an open source GIS (SAGA Development Team, 2008), was used for some feature calculations. The main programming language used was Python, specifically Python version 2.7. For extending the power of Python in scientific applications, the SciPy stack was used (Jones, 2001). Scikit-image was the library used for digital image processing (van der Walt, 2014). All of the machine learning classifiers were implemented using the scikit-learn library (Pedregosa, 2011).

3.3 Preparation of Inputs

3.3.1 LAStools-Generated LiDAR Derivatives

The LAStools software package, specifically *lasground, lasgrid,* and *blast2dem* was used to generate the needed Digital Elevation Models. *Lasground* was used on the unclassified .las file to identify the ground points. *Blast2dem* was run using different parameters to generate the Digital Surface Model (DSM), Digital Terrain Model (DTM), and Only Last Returns (OLR) raster. *Lasgrid* was used to create the Number of Returns (NOR) raster. These derivatives were generated as Tagged Image Files (TIFF) with a cell size of 0.5m and UTM projection 51N. (Figure 3a-d) There were some problematic areas in the river area of the DEMs. (Figure 4)

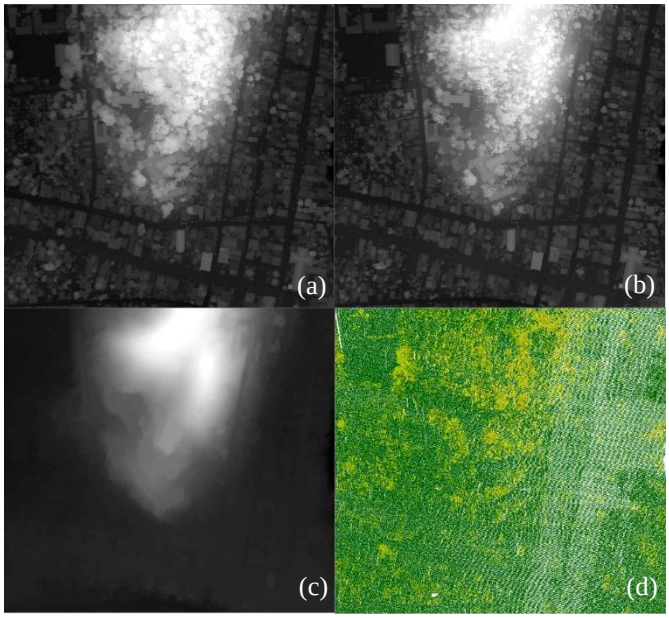

Figure 3. LASTools Generated DEM (zoomed in)
(a) DSM (b) OLR (c) DTM (d) NOR

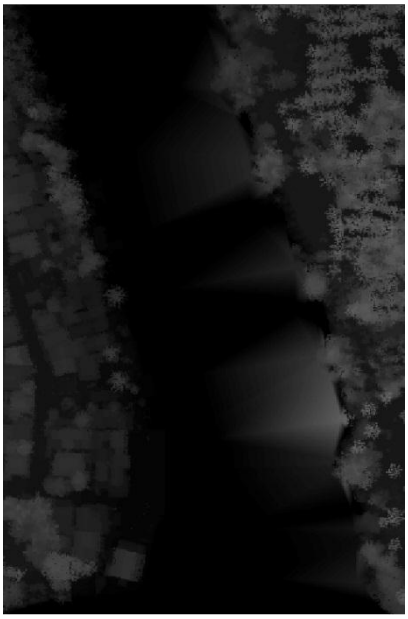

Figure 4. Problematic area in the river

3.3.2 Normalized Digital Surface Model (nDSM)

The Normalized Digital Surface Model or nDSM is a DSM that doesn't have the effect of terrain, as if the surface features were lain on a flat plane. The DSM and DTM were subtracted to create an initial nDSM. The initial nDSM was then thresholded with a value of 3 meters (we assumed that buildings are at least 3m high) and was cleaned using area thresholding. (Figure 5)

3.3 Image Segmentation

Multi-resolution segmentation is a powerful region based segmentation algorithm proposed by Baatz and Schäpe in 2000 and has also been commercially available through Trimble eCognition. It is a bottom-up region merging technique that group areas of neighboring pixels into meaningful segments or objects based on the homogeneity criteria. This will result to homogeneous areas having larger objects than heterogeneous areas. It uses three parameters in segmentation namely scale, shape, and compactness.

Multiresolution segmentation with shape of 0.5 and compactness of 0.1 was used on the nDSM. The nDSM was segmented into 10813 objects. The figure below shows the results of segmentation. (Figure 6) After removing

objects that are mostly ground pixels (mean value < 3m), the number of objects to be considered for classification goes down to 9558.

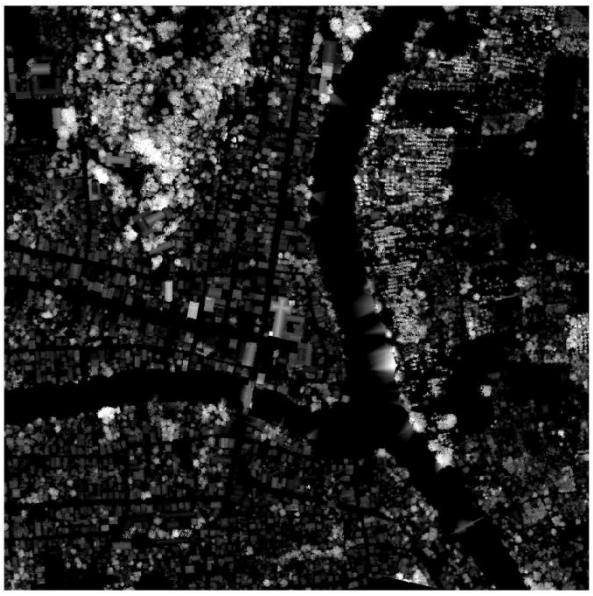

Figure 5. Normalized Digital Surface Model (nDSM)

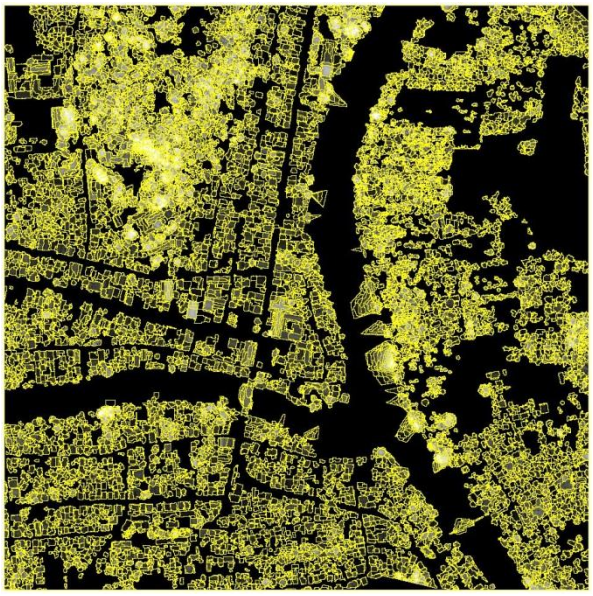

Figure 6. Result of Multiresolution Segmentation

3.4 Feature Extraction

Features were calculated on every image object generated by segmentation. These features are: *Mean Difference of First and Last Returns, Mean Number of Returns, Mean and Standard Deviation of Positive Surface Openness*.

3.4.1 Mean Difference of First and Last Returns

As mentioned earlier, the laser pulses of the LiDAR instrument can have multiple returns depending on what the pulses hit. Multiple returns can be caused by power lines, canopies, branches, some form of obstructions, etc. Building roof points usually have little difference between the first and last returns. A lower value for this feature could indicate that the object is of class building. To calculate this feature, the OLR must be first subtracted from the DSM. Then, calculate the mean of the difference for every object.

3.4.2 Mean Number of Returns

If a surface has more than one return, it is highly likely that the point in the surface is not part of a building. A lower value for this feature could indicate that the object is of class building. To calculate for this feature, calculate the mean of the NOR for every object.

3.4.3 Mean and Standard Deviation of Positive Surface Openness

Openness, a parameter which was developed by Yokoyama, Shirasawa, and Pike in 2002, is essentially an angular measure of the relation between surface reliefs and horizontal distances. This measure can be expressed in two perspectives - positive and negative openness – as shown in Figure 7. Positive values expresses the convex features of the topography or simply the openness above the surface while negative values describes the concave features or the openness below the surface.

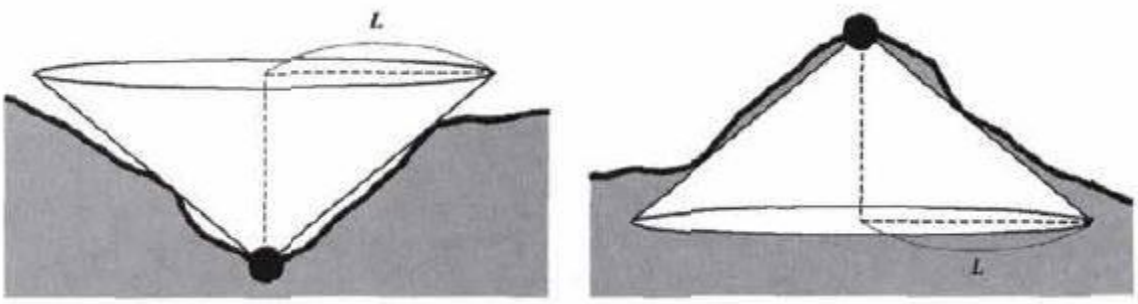

Figure 7. Positive and Negative Openness (from Yokoyama, 2002)

In this study, mean and standard deviation of positive openness in objects were used as features. Given that buildings tend to be more homogeneous than trees and vegetation, building objects usually have higher mean of positive openness and lower values of standard deviation. These features were computed using SAGA GIS.

3.5 Object Classification Using Machine Learning

The remaining step in OBIA is to classify the objects into different classes. For this study, two classes were specified: Building and Non-building.

The machine learning classifiers used in this study fall under the category of supervised learning methods. Supervised learning tries to learn a mapping between input and output variables. This mapping is then applied to previously unseen data to predict outputs. (Cord, 2008) Supervised learning uses labeled training data as input to create models for predicting outputs.

3.5.1 Machine Learning Classifiers

K-Nearest Neighbors (KNN) is one of the simplest machine learning classifiers. Examples are classified based on the class of the nearest neighbor on the feature space. Usually, more than one neighbor is taken into account to improve results, hence, K-Nearest Neighbors. (Cord, 2008)

Support Vector Machines (SVM) is a very popular machine learning technique in Object Based Image Analysis. It seeks out the optimal hyperplane that effectively separates the classes (Figure 7). Kernel functions are used to map non-linear decision boundaries to higher dimensions where they are linear. This study used the popular radial basis function (RBF) kernel. (Tzotsos, 2008) The parameter C is for adjusting the tradeoff between large margins and few classification errors. (Cord, 2008)

The Naïve Bayes (NB) Classifier is based on Bayesian statistics. It evaluates classifications using probability distributions (Cord, 2008). This study assumes that the distribution of the different features is Gaussian.

Decision Trees (DT) create models for classification by using a series of decision rules. These decision rules branch out forming trees of different depths, where deeper trees can make fitter models ("1.10. Decision Trees", n.d.). Scikit-learn's implementation uses the Classification and Regression Trees (CART) algorithm that constructs binary trees by using ideal features and thresholds to create better trees.

Random Forests (RF) is an ensemble learning method that uses a group of decision trees. Results of random forests are usually better models due to its randomness ("1.11. Ensemble methods", n.d.).

The *scikit-learn* implementations of the classifiers were used in this study.

3.5.2 Preparation of Training Data

In this study, different sizes of training data were used to see the effect of increasing training data size. Out of the 9558 objects for classification, 50,100, and 150 samples were randomly selected for both classes. The method of sampling used was stratified random sampling. We pre-classified the samples by visual inspection with the aid of the orthophoto.

3.5.3 Training, Classification, and Parameter Tuning

We used the following scikit-learn classes in this study: *KNeighborsClassifier* for K-Nearest Neighbors, *SVC* for Support Vector Machines, *GaussianNB* for Naïve Bayes, *DecisionTreeClassifier* for Decision Trees, and *RandomForestClassifier* for Random Forests.

We explored the different parameters for instantiating the classifiers. We tried different values for the number of neighbors in KNN, C in SVM, the maximum depth in DT, and the number of estimators in RF. We also trained these classifiers with different training data sizes to see the effect of differing sample sizes.

3.5 Accuracy Assessment

Ground truth samples were selected by stratified random sampling with 200 non-building objects and 100 building objects. Since the samples are scattered all throughout the dataset, they were not shown in this paper due to paper length constraints. The problematic areas of the nDSM were not considered for ground truth selection.

The performance of the classifiers was measured using the metrics *Overall Accuracy* and *Cohen's Kappa*. These metrics are derived from error matrices. A value of 0.81-0.99 is generally considered "almost perfect agreement." (Viera, 2005)

4. RESULTS AND DISCUSSION

The graphs below show the performance of the classifiers on different sizes of training data and values of parameters. KNN worked well with 50 training samples. Increasing sample size needs higher values for K-Neighbors. For this study's case, a value of 10 was found as the ideal value for C in SVM. Increasing C values led to worse performance. Increased sample size led to better performance for the case of NB. For DT, too many samples led to overfitting, thus reducing performance. It's difficult to look for a trend in RF's results due to its randomness.

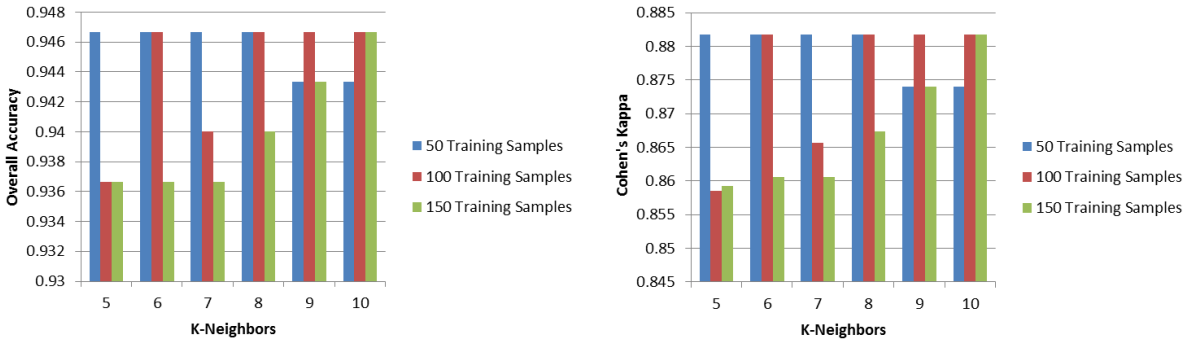

Figure 8. Performance of KNN configurations

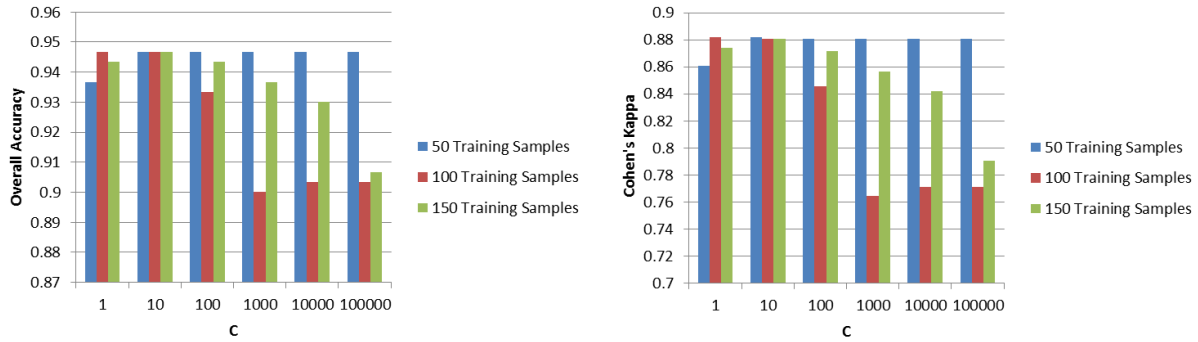

Figure 9. Performance of SVM configurations

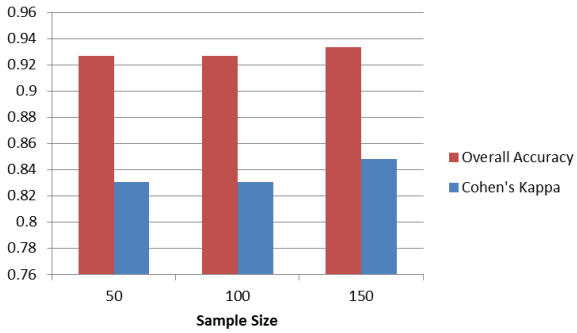

Figure 10. Performance of NB configurations

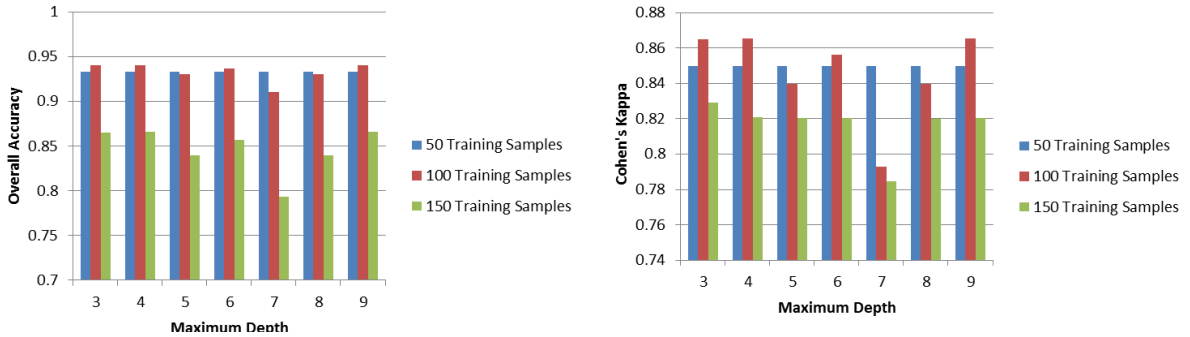

Figure 10. Performance of DT configurations

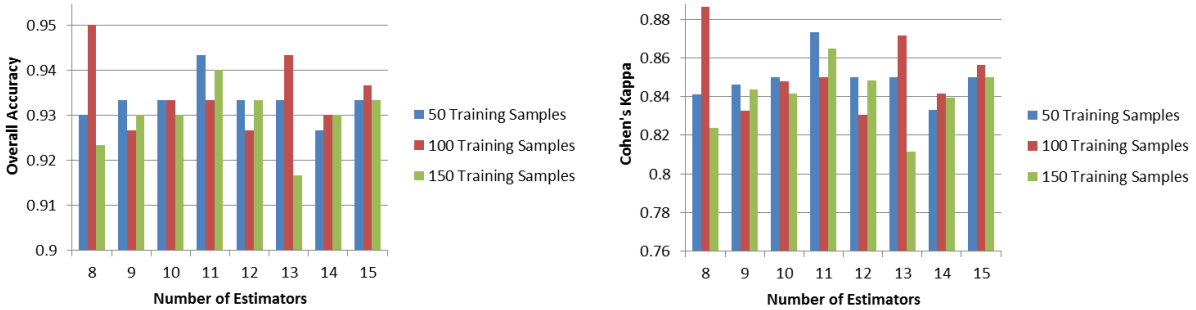

Figure 11. Performance of RF configurations

It can be noted that the overall accuracy and kappa of the classifiers are very close to each other, with almost insignificant differences. For this study's case, increasing the size of training data did not improve classification accuracy significantly. Also, tuning the different parameters did not change the performance significantly.

Table 1. Best performing configurations of the classifiers

Classifier	Training Data Size	Parameter	Value	Overall Accuracy	Cohen's Kappa
K-Nearest Neighbors	50	K-Neighbors	5	0.946666667	0.881773399
Support Vector Machines	50	C	10	0.946666667	0.881773399
Naïve Bayes	150	N/A	N/A	0.933333333	0.848101266
Decision Trees	100	Maximum Depth	4	0.94	0.865336658
Random Forest	100	Number of Estimators	8	0.95	0.886649874

The table above shows the best performing of all the configurations tried for every classifier and the output of these classifiers zoomed in on a certain area. RF with 8 estimators and 100 samples per class was the best performing classifier with an overall accuracy of 95% with a kappa value of 0.8866. The images below show the outputs of these best performing classifiers. It can be seen that the results are very similar.

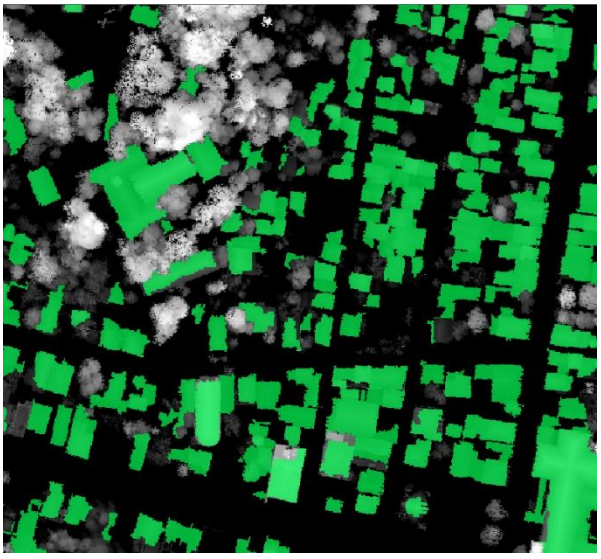

Figure 12. Result of best performing KNN configuration

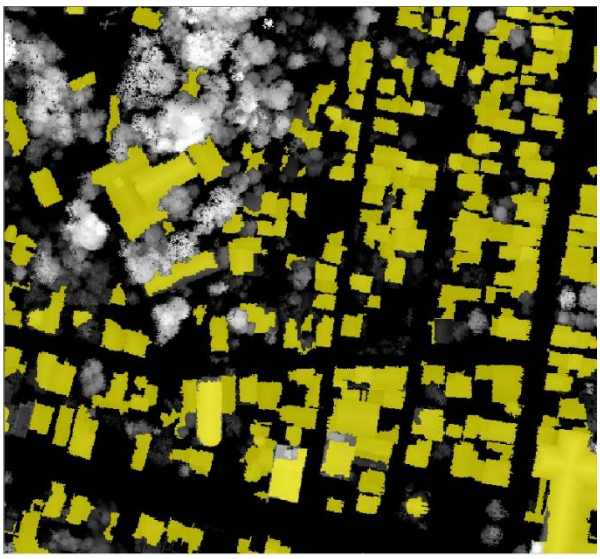

Figure 13. Result of best performing SVM configuration

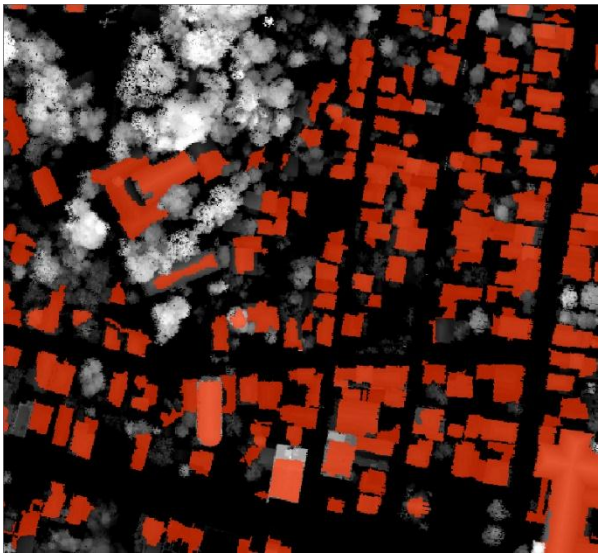

Figure 14. Result of best performing NB configuration

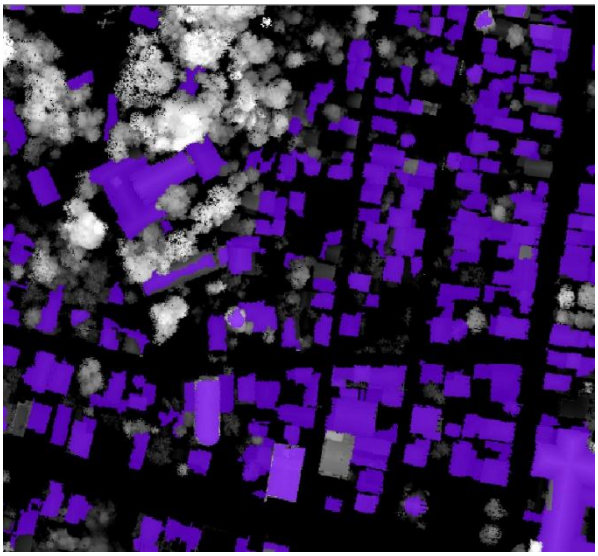

Figure 15. Result of best performing DT configuration

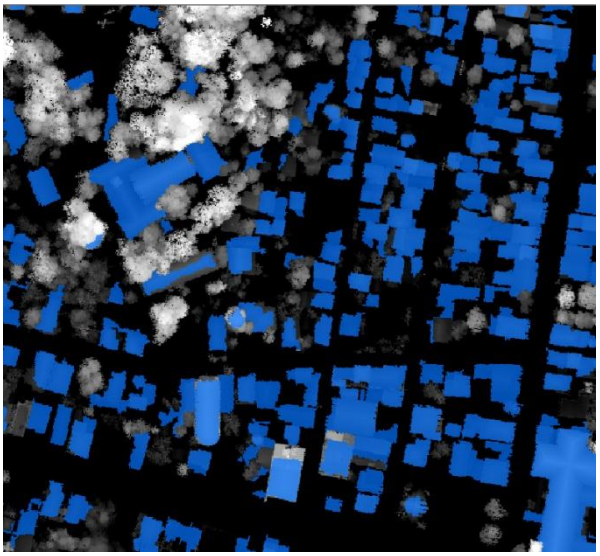

Figure 16. Result of best performing RF configuration

It was shown earlier that there were problematic areas in the nDSM. These areas were misclassified (Figure 17) by the classifiers since they exhibit features similar to buildings. These misclassifications were ignored in this study since it was a problem with the data and outside the scope of this study.

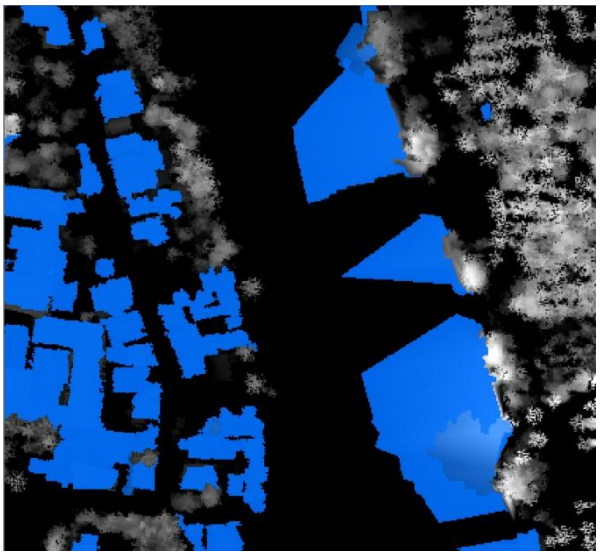

Figure 17. Misclassification of problematic river area

5. CONCLUSION AND FUTURE WORK

In this study, we sought to maximize the power of LiDAR information by deriving features like the mean difference of first returns, mean number of returns, and the mean/standard deviation of positive surface openness. We evaluated and compared five different machine learning methods for building extraction, namely K-Nearest Neighbors, Support Vector Machines, Naïve Bayes, Decision Trees, and Random Forests. We trained these classifiers with varying training data size. We tested different parameter values for the classifiers; we tried different values for the number of neighbors in KNN, the regularization parameter C in SVM, the maximum depth in DT, and the number of estimators in RF.

The Random Forest classifier with 8 estimators and 100 samples per class was found to be the best performing classifier. The results are very similar with each other. We found out that the overall accuracy and kappa of all the classifiers are very close to each other, with almost insignificant differences. We found that for our case, increasing the size of training data and tuning the parameters for the classifiers did not improve classification accuracy that much. These probably are indicators of the effectiveness of the selected features in classifying buildings.

The classifiers in this study were only tried for one dataset, this study could be extended to handle all datasets. Further research can be done on deriving more features from LiDAR data. Better features will result to even better classifications. With more features, studies can be done on classification of more specific classes like farm areas, forest types, building types, etc.

BIBLIOGRAPHY

1.10 Decision Trees. n.d. Retrieved August 30, 2015 from http://scikit-learn.org/stable/modules/tree.html

1.11 Ensemble Methods. n.d. Retrieved August 30, 2015 from http://scikit-learn.org/stable/modules/ensemble.html

Blaschke, T., 2010. Object based image analysis for remote sensing. ISPRS journal of photogrammetry and remote sensing, 65(1), pp. 2-16.

Baatz, M., & Schäpe, A., 2000. Multiresolution segmentation: an optimization approach for high quality multi-scale image segmentation. Angewandte Geographische Informationsverarbeitung XII, pp. 12-23.

Campagnolo, M. L., & Cerdeira, J. O., 2007. Contextual classification of remotely sensed images with integer linear programming. Proceedings of CompIMAGE-Computational Modelling of Objects Represented in Images: Fundamentals, Methods and Applications, pp. 123-128.

Campbell, J.B., Wynne, R.H., 2011. Introduction to Remote Sensing. The Guilford Press, New York, p. 243
Cord, M., & Cunningham, P. (Eds.)., 2008. Machine learning techniques for multimedia: case studies on organization and retrieval. Springer Science & Business Media., p. 1-18, 29

Gao, Y., & Mas, J. F., 2008. A comparison of the performance of pixel-based and object-based classifications over images with various spatial resolutions. Online Journal of Earth Sciences, 2(1), pp. 27-35.

Jones, E., Oliphant, E., Peterson, P., et al., 2001. SciPy: Open Source Scientific Tools for Python, from http://www.scipy.org/

Kohavi, R., Provost, F., 1998. Glossary of terms. Machine Learning, 30(2-3), 271-274.

Meng, X., Wang, L., & Currit, N., 2009. Morphology-based building detection from airborne LIDAR data. Photogrammetric Engineering & Remote Sensing, 75(4), pp. 437-442.

Navulur, K., 2006. Multispectral image analysis using the object-oriented paradigm. CRC press, Taylor & Francis, Boca Raton, p. 205.

NOAA, 2015. What is LiDAR?, Retrieved August 28, 2015, from http://oceanservice.noaa.gov/facts/lidar.html

Pedregosa et al., 2011. Scikit-learn: Machine Learning in Python. JMLR 12, pp. 2825-2830.

Qian, Y., Zhou, W., Yan, J., Li, W., & Han, L., 2014. Comparing machine learning classifiers for object-based land cover classification using very high resolution imagery. Remote Sensing, 7(1), pp. 153-168.

rapidlasso GmbH, 2015. "LAStools - efficient LiDAR processing software" (version 150202, commercial), from http://rapidlasso.com/LAStools

Singh, L., 2006. Extraction of different buildings from LiDAR data. In ASPRS Annual Conference. Reno, Nevada.
Tzotsos, A., & Argialas, D., 2008. Support Vector Machine Classification for Object-Based Image Analysis. Object-Based Image Analysis, Lecture Notes in Geoinformation and Cartography, pp. 663-677.

Uzar, M., 2014. Automatic building extraction with multi-sensor data using rule-based classification. European Journal of Remote Sensing, 47(8).

van der Walt, S., Schönberger, J. L., Nunez-Iglesias, J., Boulogne F., Warner, J. D., Yager, N., Gouillart, E., Yu, T., et al., 2014. scikit-image: Image processing in Python. from http://scikit-image.org/
Viera, A. J., & Garrett, J. M., 2005. Understanding interobserver agreement: the kappa statistic. Fam Med, 37(5), pp. 360-363.

Yokoyama, R., Shirasawa, M., & Pike, R. J., 2002. Visualizing topography by openness: a new application of image processing to digital elevation models. Photogrammetric Engineering & Remote Sensing, 68(3), pp. 257-266.