A preliminary study on machine learning and google earth engine for mangrove mapping

Muhammad Kamal¹, Nur Mohammad Farda¹, Ilham Jamaluddin¹, Artha Parela¹, Ketut Wikantika², Lilik Budi Prasetyo³, Bambang Irawan⁴

¹Department of Geographic Information Science, Faculty of Geography, Universitas Gadjah Mada Sekip Utara, Bulaksumur, Yogyakarta 55281 Indonesia
²Remote Sensing and Geographical Information Sciences Research Group, Faculty of Earth Science and Technology, Bandung Institute of Technology Jl. Ganesa 10, Bandung 40132 Indonesia
³Department of Natural Resource Conservation and Ecotourism, Faculty of Forestry, IPB University Kampus IPB Darmaga, Bogor 16680 Indonesia
⁴Biology Study Program, Faculty of Science & Technology, Airlangga University Jalan Mulyosari, Surabaya 60115 Indonesia

e-mail: m.kamal@ugm.ac.id

Abstract. The alarming rate of global mangrove forest degradation corroborates the need for providing fast, up-to-date and accurate mangrove maps. Conventional scene by scene image classification approach is inefficient and time consuming. The development of Google Earth Engine (GEE) provides a cloud platform to access and seamlessly process large amount of freely available satellite imagery. The GEE also provides a set of the state-of-the-art classifiers for pixel-based classification that can be used for mangrove mapping. This study is an initial effort which is aimed to combine machine learning and GEE for mapping mangrove extent. We used two Landsat 8 scenes over Agats and Timika Papua area as pilot images for this study; path 102 row 64 (2014/10/19) and path 103 row 63 (2013/05/16). The first image was used to develop local training areas for the machine learning classification, while the second one was used as a test image for GEE on the cloud. A total of 838 points samples were collected representing mangroves (244), non-mangroves (161), water bodies (311), and cloud (122) class. These training areas were used by support vector machine classifier in GEE to classify the first image. The classification result show mangrove objects could be efficiently delineated by this algorithm as confirmed by visual checking. This algorithm was then applied to the second image in GEE to check the consistency of the result. A simultaneous view of both classified images shows a corresponding pattern of mangrove forest, which mean the mangrove object has been consistently delineated by the algorithm.

1. Introduction

Mapping the location of mangroves is very important to know the status of mangrove areas, both at local, national and global scale. This data can be used as a baseline in the context of sustainable management and monitoring of mangrove ecosystems. In this case, remote sensing data can be effectively used to support the mangrove mapping effort [1, 2]. The problem that is often experienced in the use of remote sensing data is the efficiency of the mapping process. If the mapped area is large, it requires huge resources in terms of energy, time, and cost. So that conventional scene-by-scene mangrove mapping with remote sensing imagery is considered inefficient.

Accurate mangrove mapping using remote sensing data, especially in very large areas, is challenging. For example, the input image must not be covered by clouds, fog and other disturbances to obtain an
acceptable classification result [3]. Such input images can be produced by combining more than one satellite image collected over a certain period. Landsat sensors have been collecting Earth Observation (EO) data at frequent intervals since the 1970s [4] (https://www.usgs.gov/land-resources/nli/landsat). All Landsat imagery files can be openly accessed since 2008 [5]. This kind of satellite imagery archive is very useful for mapping mangroves over large areas, but is still underutilized because collecting, storing, processing, and manipulating multi-temporal remote sensing data covering large geographical areas cannot be done using conventional software on a PC workstation based. This is known as the "Geospatial Big Data" problem and demands technology and resources that can handle large-scale satellite imagery [6].

The presence of cloud processing technology such as Google Earth Engine (GEE) provides an alternative image processing source that is more efficient [7]. GEE provides a cloud platform to access and seamlessly process large amount of freely available satellite imagery. Algorithms developed through GEE can be replicated and applied for image processing over large areas, including for mangrove mapping. This research is a preliminary study to develop a machine learning algorithm that automatically delineates mangroves from OLI's Landsat 8 imagery. Therefore, the objective of this research is the development of GEE machine learning for mangrove mapping using Landsat 8 OLI data.

2. Materials and Methods

2.1. Study site and image datasets
Two sets of Landsat 8 OLI scenes of Agats and Timika areas in Papua Indonesia were used in this research (figure 1). We used Landsat data path 102 and row 64 image (acquired at 2014/10/19) for data training to develop local training areas for the machine learning classification algorithm for discriminating mangrove and non-mangrove objects. While the Landsat data path 103 and row 63 image (acquired at 2013/05/16) was used as a test image to apply the algorithm for GEE on the cloud.

These Landsat acquisition dates were selected because these scenes contain minimum cloud cover. To enable fair comparison between these two images with different acquisition dates, these images have been radiometrically calibrated up to top of atmosphere (TOA) reflectance level using GEE image calibration function following the procedure of Chander et al. [8]. In this research, Agats and Timika areas have been selected for this study because it has one of the largest extents of mangrove forest in Indonesia which is easily recognized from Landsat images. The mangrove forest in this area also considered as pristine ecosystem where the disturbance is minimum, thus make the classification comparison resulted from two different years of Landsat scenes possible.

![Figure 1. Two Landsat 8 OLI scenes used in this research.](image-url)
2.2. Machine learning algorithm
The machine learning algorithm applied in this research is support vector machine (SVM) which was introduced by Vapnik in 1979 [9]. SVM is a supervised non-parametric statistical learning algorithm, thus there is no assumption made on the underlying data distribution. SVM is an alternative classification technique [10], which has been applied to various classification problems of remote sensing data [11]. It is based on statistical learning which aims to determine location boundary decisions to produce optimal separation between classes (figure 2). SVM is a machine learning method that works based on the principle of Structural Risk Minimization (SRM) to minimize the probability of misclassification in order to find the best hyperplane that separates two classes of input space [10].

![Separation of two classes with optimum hyperplane in SVM.](image)

The principle of SVM classification is to find a classifier function by conducting the training process by entering training data into a vector space and finding an optimal hyperplane to separate class classes from training data. The classification of new data points is done by looking at the side of the class where the new data points are located. The best hyperplane separator between the two classes can be found by measuring the margin of the hyperplane and finding its maximum point. Margin is the distance between the hyperplane and the closest pattern of each class. The closest pattern is called a support vector. The solid line in figure 2 shows the optimum hyperplane, which is located right in the middle of the two classes, while the filled circle and triangle dots are support vectors. The effort to find the location of a hyperplane is the core of the learning process in SVM.

2.3. Training samples and result validation
To support SVM algorithm, we purposively selected a total of 838-point samples across the 2014 image. These sample points were collected to represent four major classes found on the image, which includes mangroves (244 points), non-mangroves (161 points), water bodies (311 points), and clouds (122 points) classes (figure 3). Although SVM has advantages in terms of their ability to generalize classification well even with limited training samples [12], we collected large number of samples to (1) cover spectral variation of targeted classes on the image and (2) enable accurate and reliable detection of targeted objects. To assess the effectiveness of the automatic delineation, the accuracy assessment was performed by directly comparing the mangrove polygons resulted from GEE with the result of mangrove polygons from visual interpretation.
3. Results and Discussions

3.1. SVM algorithm implementation in GEE
The SVM classification algorithm was developed in GEE using Agats Landsat 8 image (path 102, row 64, acquired at 2014/10/19). Only eight out of eleven bands of Landsat 8 OLI were relevant in this research, these are blue (450-510nm), green (530-590nm), red (640-670nm), near infrared (850-880nm), shortwave infrared1 (1570-1650nm), and shortwave infrared2 (2110-2290nm) bands. The coastal blue (band 1) band contains too much noise, and the 8th to 11th bands were irrelevant to this research. The sequence of steps summary for the classification algorithm is presented in Table 1, while the algorithm code implemented in GEE is presented in Appendix 1.

Table 1. Classification algorithm summary.

| Step | Task description |
|------|------------------|
| Setting up the model image and SVM algorithm | To define the Landsat 8 bands used as input for classification |
| 1    | To define the selection of Landsat 8 OLI with several parameters such as correction level, date, cloud cover |
| 2    | To make the median value of Landsat 8 OLI pixel values |
| 3    | To retrieve the ROI that has been created on googleasset ee.FeatureCollection(table) |
| 4    | To define the training variable used to input pixel values at each sample point |
| 5    | To define the type of algorithm used (i.e. SVM) |
| 6    | To define the variable for classification |
| 7    | To define a Landsat 8 image that will be applied with a predefined classification variable |
| 8    | To set center map script |
Applying SVM algorithm to the test image

1. To create a script to call Landsat 8 images with filter band, path row, cloud cover, date, and level correction
2. To create a script to change the pixel value into the median value
3. To make sample points for four targeted classes which are then included in Google assets on the Google Earth Engine
4. To retrieve for sample points that has been created
5. To create training variables that are used as training areas where the existing sample points will take their pixel values (pixel value of Landsat model path 102 row 64) and serve as a guide to the classification of machine learning
6. To create a script to define the type of machine learning algorithm used (i.e. SVM)
7. The results of the training variables are then applied to the Landsat model image (path 102 row 64) with the SVM algorithm
8. Then the training variable is re-applied to the next image scene (path 103 row 63) which is explained based on the pixel value of each class from the input image model (path 102 row 64)
9. The results of the classification of the two images are then exported

3.2. Mangrove delineation results
The final classification result indicates that most of mangrove forest has been properly delineated by the SVM algorithm applied in GEE (figure 4). The green feature along the coastline and lower river streams of Agats and Timika areas in figure 4 represents mangrove objects found on the images. This is a good indication of the effectiveness of the classification algorithm because mangrove habitats are commonly associated with coastline areas and lower river streams where there are tidal fluctuations and anaerobic sediments formed [13]. The water body features were also properly classified by the algorithm. The blue-coloured features represent water bodies object including sea water and river streams. Both model and test images confirmed the effectiveness of the algorithm in delineating water bodies. Non-mangroves features including land vegetation, built-up objects, agricultural areas, exposed soil, etc. were also correctly delineated by the algorithm, as well as the cloud objects.

However, a closer inspection of the map result revealed some misclassified mangrove objects, especially in the upper land part of the research site. Some classification errors were also found in association with cloud and hill shadow as indicated in the test image. These objects may have similar spectral characteristics to mangrove objects, thus misclassified as mangroves.

![Figure 4. The SVM classification result.](image_url)
3.3. Result validation
To assess and evaluate the performance of the algorithm developed, a direct comparison of mangrove map produced from GEE and from visual interpretation was performed. In this case, the mangrove map produced from image visual interpretation was used as reference to validate the GEE derived mangrove map. Figure 5 shows the overlay result between these two maps for the test image. The green features represent the correctly classified mangroves, and at the other hand the red features represent the misclassified mangrove objects. Figure 5 confirms that most of the misclassified mangrove objects were in the upper mainland of the research site. These classification errors mostly correspond to darker area of hill and cloud shadows where the spectral reflectance is similar to mangrove objects.

![Figure 5. The result of mangrove classification result validation of the test image.](image-url)

4. Conclusions
This research provides a preliminary study of SVM algorithm in GEE to map mangrove objects from Landsat 8 OLI data. The developed algorithm effectively delineated most of the main mangrove object along the coastline of the study area. However, there are many misclassifications found due to the shadows from hills and clouds. Therefore, future work is needed to improve the algorithm in order to obtain a more accurate delineation of mangrove features.

5. Acknowledgements
This research is funded by Indonesian Collaborative Research – WCU (World Class University) year 2019, contract number 635/UN1/DITLIT/DIT-LIT/LT/2019. The authors would like to thank (1) Department of Geographic Information Science, Faculty of Geography, Universitas Gadjah Mada for providing support for this research, and (2) USGS for providing free access to Landsat 8 OLI data used in this research.

6. References
[1] Kuenzer, C., et al., Remote sensing of mangrove ecosystems: A review. Remote Sensing, 2011. 3(5): p. 878-928.
[2] Heumann, B.W., Satellite remote sensing of mangrove forests: Recent advances and future opportunities. Progress in Physical Geography, 2011. 35(1): p. 87-108.
[3] Gallant, A., The challenges of remote monitoring of wetlands. 2015, Multidisciplinary Digital Publishing Institute.
[4] Xie, Y., Z. Sha, and M. Yu, Remote sensing imagery in vegetation mapping: a review. Journal of plant ecology, 2008. 1(1): p. 9-23.
[5] Wulder, M.A., et al., Opening the archive: How free data has enabled the science and monitoring promise of Landsat. Remote Sensing of Environment, 2012. 122: p. 2-10.
Appendix

//First task
var bands = ['B2', 'B3', 'B4', 'B5', 'B6', 'B7'];

var landsatmodel = ee.ImageCollection('LANDSAT/LC08/C01/T1_TOA')
  .filter(ee.Filter.lt('CLOUD_COVER', 5))
  .filterDate('2013-01-01', '2019-03-30')
  .filter(ee.Filter.eq('WRS_PATH', 102)).filter(ee.Filter.eq('WRS_ROW', 64));

var medianlandsatmodel = landsatmodel.median();
var points = ee.FeatureCollection(table)
  .remap([1, 2, 3], [1, 2, 3], 'Class');
var training = medianlandsatmodel.sampleRegions(
  collection: points,
  scale: 30
);

var classifier = ee.Classifier.svm(
  kernelType: 'RBF',
  gamma: 0.5,
  cost: 10
);
var trained = classifier.train(training, 'Class', bands);
var classified = medianlandsatmodel.select(bands).classify(trained);
Map.setCenter(138.124012, -5.527976, 7);
var visParams = {
  bands: ['B5', 'B6', 'B7'],
  min: 0,
  max: 0.4,
};
var palette = [
  'aec3d4',
  '152106',
  '225129',
  '369b47',
  '30eb5b',
  '387242',
  '6a2325',
  'c3aa69',
  'b76031',
  'd9903d',
  '91af40',
  'cc0013'
]
Map.addLayer(medianlandsatmodel, visParams, 'clipped composite');
Map.addLayer(classified, {min: 0, max: 3, palette: palette}, 'classification');

//Second task
var landsatuse = ee.ImageCollection('LANDSAT/LC08/C01/T1_TOA')
  .filter(ee.Filter.lt('CLOUD_COVER', 10))
  .filterDate('2013-01-01', '2019-03-30')
  .filter(ee.Filter.eq('WRS_PATH', 103))
  .filter(ee.Filter.eq('WRS_ROW', 63));
print(landsatuse)

var medianlandsatuse = landsatuse.median();
var classified2 = medianlandsatuse.select(bands).classify(trained);

Map.setCenter(138.124012, -5.527976, 7);
var visParams = {
bands: ['B4', 'B3', 'B2'],
  min: 0,
  max: 0.4,
};
var palette = ['aec3d4',//air
  '152106', '225129', '369b47', '30eb5b', '387242',//as
  '6a2325', 'c3aa69', 'b76031', 'd9903d', '91af40', //asa
  'cc0013',//ada
};
Map.addLayer(medianlandsatuse, visParams, 'clipped composite');
Map.addLayer(classified2, {min: 0, max: 3, palette: palette}, 'classification');

Export.image.toDrive({
  image: classified2,
  description: 'klasifikasi2',
  scale: 30,
  region: geometry,
  crs: 'EPSG:4326',
  maxPixels: 10000000000000
});