Water Body Detection from Satellite Images
Progress Report

D.S. Geldenhuys & S.R.K. Gasson

March 2020

1 Introduction

This report forms part an entry for the geosmart automatic water body delineation competition. In this report we investigated methods for delineating water bodies from geo-referenced images. The objective as stipulated by the guidelines of this geosmart competition, is to generate a geojson file containing the polygons that delineate water bodies in a given geo-referenced image.

In this application we used a two tier approach. In the first tier we segment the water bodies in the image. In the second tier we generate polygons from the segmented images.

Upon investigating the images we noticed the the variance in pixel intensities of water bodies are particularly low. We theorised that this could be used to determine whether or not an area forms part of a water body, and achieved promising results using simple image processing. There were areas where this did not perform adequately, notable the edges of the water bodies. Another downfall of this method was the need to manually determine the variance threshold. Similar methods have been attempted using entropy instead of variance [1]. One expects areas of low variance to also have low entropy. According to literature this method can work well on optical images [2].
It was decided that the method could benefit from other feature descriptors. Water bodies typically appear darker than their surroundings and contain edges with sharp contrast. Since we have training data we proposed to train a random forest classifier using some of these features to segment waters bodies in an image.

We encountered numerous problems and found solutions all of which are explained in the preliminary results and discussion section.

The details of the algorithm are discussed in the next section.

2 Methodology

We developed a two tier approach to generating polygons for water bodies in a given geo referenced image. The aim of the first tier is to extract the water bodies in the image, creating a binary mask.

The second tier aims to generate polygons from this mask.

2.1 Tier 1

In the first tier we used various feature descriptors to extract the local features for each pixel in the image where each pixel is given a binary label based on these feature values. Either 'within-water-body' or 'outside-water-body'.

The feature vectors with their corresponding labels were used to train a random forest to classify each pixel in an image. The result of the classification is called the predicted binary feature image. After some post processing on the binary image we end with the final extracted binary feature image. The various parts of this process are described in the following sections.

The main steps are listed below

1. Feature extraction
2. Training
3. predicting
4. post processing

2.1.1 Feature extraction

To extract features we used a multi resolution approach; we crop two windows of different sizes around a central pixel. From these windows we extract features using various features descriptors. the descriptor values are collated into a single feature vector.

The feature descriptors were chosen via observation, experimentation and literature.[1][2]. All the Feature descriptors used are listed in table 1.

2.1.2 Training data

We split the competition data 70:30 for training and testing respectively. For each training instance we extracted features for each pixel in the images in the training set as described in the **Feature extraction** section. To label these pixels we used the data provided in this competition to generate a set of binary feature images which we could then use to determine if each pixel is water-body or not. Finally we obtain a training set of feature vectors with binary labels.

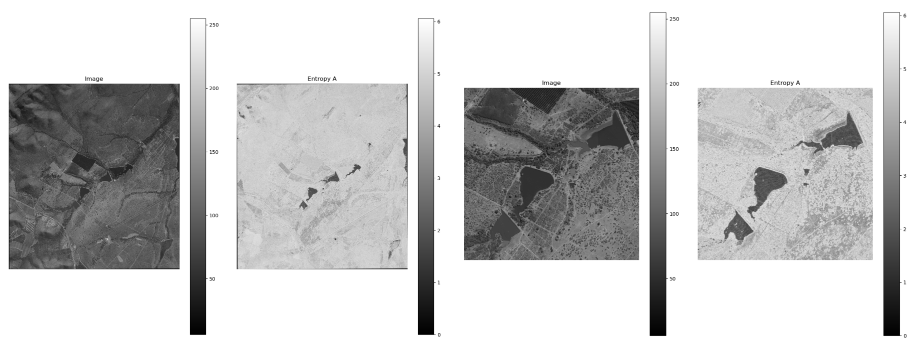
Table 1: Feature descriptors

Feature descriptor	description
Shannon's Entropy	calculates the amount of information in a given window
colour (red)	intensity of red band
colour (green)	intensity of green band
colour (blue)	intensity of blue band

2.1.3 Random forest classifier

The learning algorithm we used was a Random Forest since this approach is fast and robust to over fitting, and able to learn nonlinear relationships between descriptors and labels. One additional benefit is the ability to analyze the random forest and evaluate the influence of each feature descriptor. The entropy descriptor was noticeably more influential than the other descriptors. Figure 1 shows how entropy of the water body is noticeably lower than other parts of the image.The darker areas represent lower entropy.

Figure 1: Original image plotted against entropy plot



2.1.4 Post processing

In this step noise is reduced from the predicted results. Firstly, holes within the water bodies are filled before any areas of small positive noise surrounding the water bodies are removed.

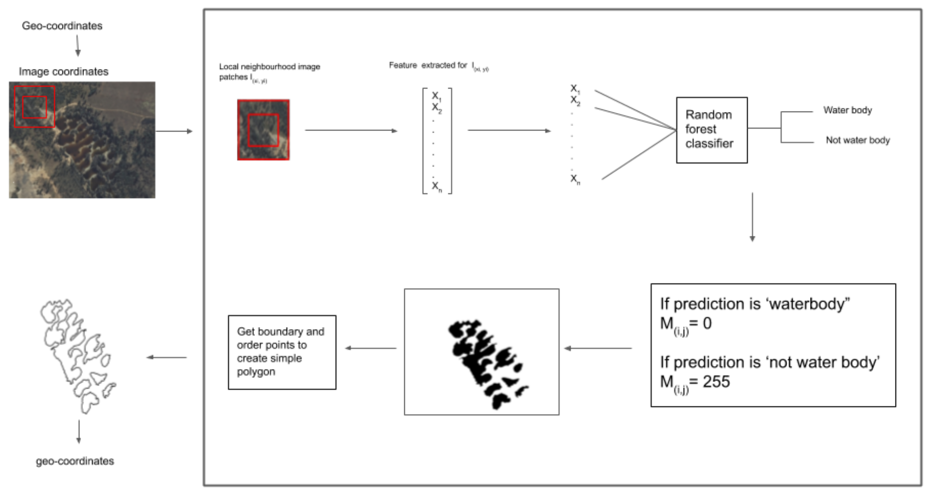
2.2 Tier 2

In this section we want to create a polygon from a binary feature image. To do this we created two functions, one to extract the edge pixels of the water-bodies and another one to order the pixels to generate a simple polygon, these polygon coordinates can then be mapped back to geo-coordinates and saved as geojson file.

2.3 Full algorithm

The deployment algorithm contains the trained random forest and all of the processing functions to produce geo-refenced polygons in the form of a geojson file given some geotiff file as an input. A schematic for the algorithm is shown below

Figure 2: workflow



3 Preliminary results and discussion

to train the random forest we must ensure that the data is sufficiently mixed and uniformly distributed. During the first model iterations the precision grew worse with more data. We identified that the problem was due to the data not being sufficiently mixed across the entire data set which skewed the learning problem. After we sufficiently mixed the training data the accuracy improved substantially.

Although the accuracy increased significantly, it was still lower than expected with 76% for commission and 94% for omission. Upon investigating the data more thoroughly we noticed that some of the water-bodies where not labeled, resulting in numerous false negative labels in the training set. This ultimately decreases the model performance significantly.

As a temporary measure to remove this fault, We trained the model on a small set of data that contained no mislabeled images, the model achieved very good predictions on some images. Some qualitative results are shown below for a single test image.

Figure 3: workflow

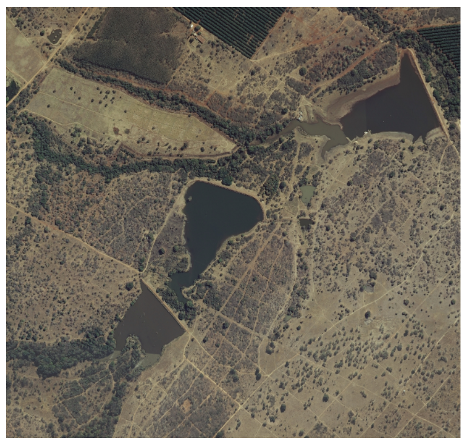
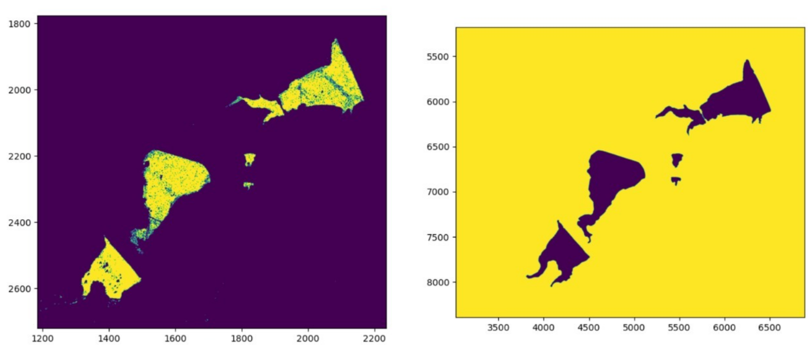


Figure 4: model predictions



It is worth noting that the model here does not perform well when the satellite images overlap, an example of this area is shown below. This is also evident in figure 8 where the boundaries deviate most noticeably.

Figure 5: satellite image overlap

Figure 6: predicted water-body and label water-body

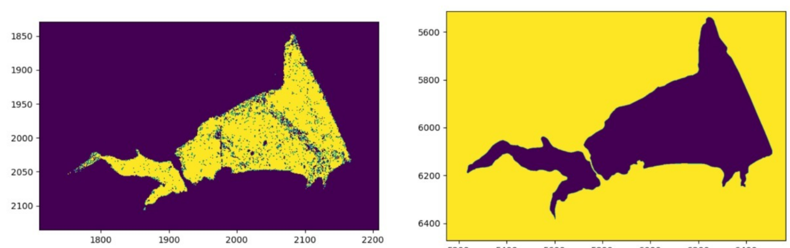


Figure 7: predicted water body and post-processed water-body

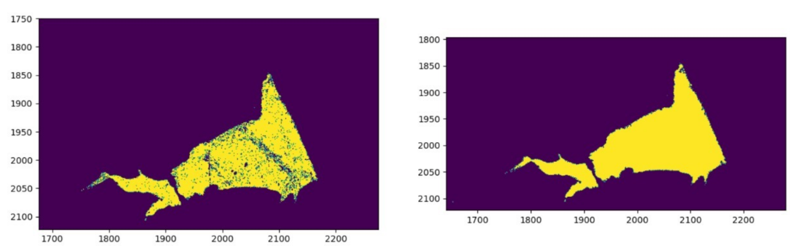
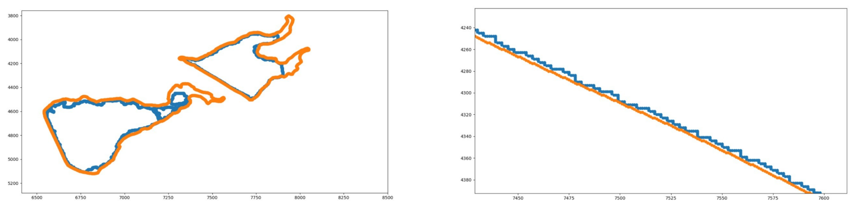


Figure 8: plots of reference and extracted boundaries



ideally we want predictions of around 95% and 95% for commission and omission. Since we already have 74% and 96% for incorrectly labeled training examples we expect the model to perform much better once we create a new corrected training set.

On the subject of computational efficiency, we recently improved performance significantly. Previously, extracting features took around 7 hours for the entire data-set. We have greatly decreased this with our new solution but cannot comment on the exact time as we must still run tests.

4 Going further

The first task is to create a new training set, excluding any unlabeled water-bodies. Do do this we will manually crop images. After creating the new data we will run the algorithm again to test precision. If the accuracy is not sufficient we will add another descriptor that should represent the learning problem more effectively producing better results.
The features we used seem to describe the learning problem sufficiently but we will need to run more tests with the corrected training data. Since we have managed to greatly increase computational efficiency we should be able to add more feature descriptors which should further improve the accuracy if necessary. In addition to optimizing the feature extraction and random forest model we would also like to perform fine-tuning across the entire process; model prediction and post-processing. Currently we are manually choosing parameters for post-processing, fine-tuning will allow us to fit optimal parameters across the entire process.

In the event that the final random forest model does not perform adequately or to find a more computational efficient method we will attempt to use a CNN segmentation model which learns the features and attempts to segment the water-bodies, outputting the binary feature image directly as an output. We did attempt to use a CNN instead of a random forest for classification but this preformed worse than the random forest. We suspect a segmentation model is more suitable since it directly computes our binary feature image from an original image.

References

[1] Z. Zhaohui, V. Prinet, and M. Songde, "Water body extraction from multi-source satellite images," vol. 6, pp. 3970 – 3972 vol.6, 08 2003.

[2] R. K. Nath and S. K. Deb, "Water-body area extraction from high resolution satellite images-an introduction, review, and comparison," *International Journal of Image Processing (IJIP)*, vol. 3, no. 6, pp. 265–384, 2010.