Quantitative estimation of the area affected by the 2022 Jagersfontain tailings dam's collapse

Anonymous Authors[1]

Abstract

In this assignment, I show how a unsupervised classification algorithm (Kmeans) applied to multispectral satellite images is used to quantify the area affected by a recent tailings dam's collapse in South Africa. The first step is to gather relevant data, here two Sentinel-2 images, then classify each pixel in different classes for each image. The classified pixesl from both images are compared to see how much of a difference there is between both images. The results show that an unsupervised algorithm isn't the optimal option and the analysis will be done again with a supervised algorithm.

1. Introduction

Jagersfountain is a town of almost 6000 inhabitants in the Free State province of South Africa. A diamond mine was active from 1870 to 1970. On September 11th 2022, the tailings dam wall just outside the town collapsed due to a structural failure. This created a mudslide of a few kilometers that killed 10 people, injured 40 more and swept away houses and animals.

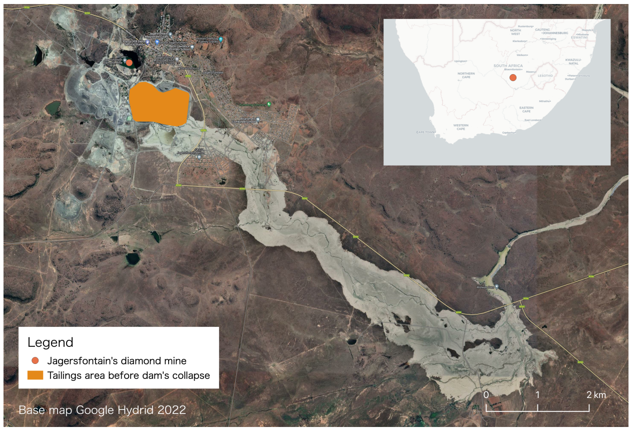
Legend
Jagersfontain's diamond mine
Tailings area before dam's collapse
0
1
2 km
Base map Google Hydrid 2022
SOUTH AFRICA

Figure 1. Situation map of the Jagersfontain town and mine

The collapse also affected the nearby Prosesspruit river as well as the ecosystem and biodiversity as the mudslide entered the river and tailings were transported further downstream.

1.1. Goal

The goal of this article is to find a method that can quickly and efficiently quantify the area affected by the disaster. This can then be used to see how far the mudslide went, how far the tailings were transported by the river or estimate costs to rehabilitate the area.

1.2. Region of interest

The selected region of interest is a rectangle of 2084.5 km^2 that includes Jagersfontain and the mudslide in its southern part and the Prosesspruit river (that then becomes the Kromellenboogspruit river) until the Kalkfontein dam reservoir north of it.

2. Data

2.1. Satellite images

Two Sentinel-2 satellite images were chosen to quantify the affected area: one before and one after the dam's collapse. The process to acquire them on Google Earth Engine is the following:

- Define a point on the Jagersfountain tailings dam at latitude -29.77068 and longitude 25.423884.
- Define two time frames; one before the dam's collpase from July 11th to Septembre 10th 2022 and another from Septembre 12th to Octobre 11th 2022. The goal here is to find the best image not too long before the event and the best one not too long after.
- Load the surface reflectance Sentinel-2 dataset (Level-2A orthorectified atmospherically corrected surface reflectance), filter it to only keep images that include Jagersfontain, and split it in two collections according to the predefined dates.
- Sort both collections by cloud cover to get the images with the least cloud. Luckily the output is 2 totally clear images of the zone of interest. One from July

The ”Before” collection	**The ”After” collection**
July 11th-September 10th	Septemb12th-October 11th
Total images: 12	Total images: 5

Table 1. The two collections

13th (Figure ??) and the other from September 16th (Figure 2).

- The bands used are blue, green, red, red edge 1, 2 and 3, near infrared, water vapor, SWIR 1 and 2 scaled to 10m.

Sentinel-2 was used instead of other satellite imagery because of its high spatial resolution of 10m/pixel, which is good as the area of interest is rather small. Another reason is its short revisit time of about 5 days, which allowed to have images right before and after the collapse.

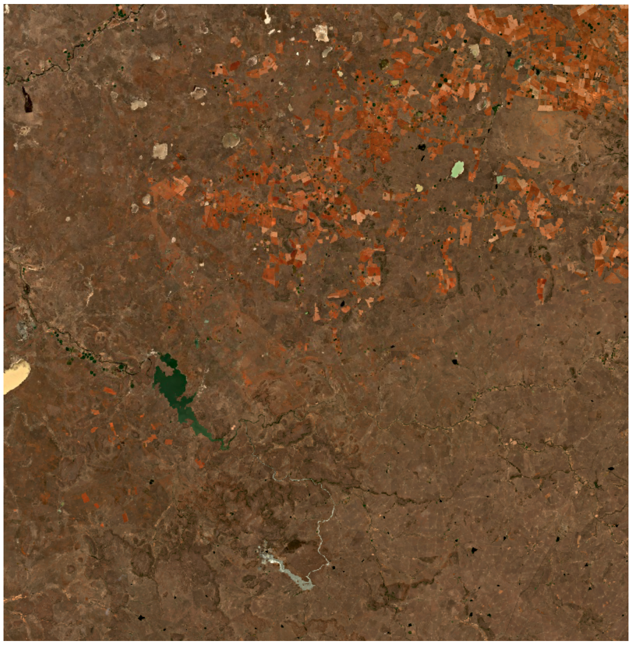

Figure 2. Sentinel-2 image from September 16th 2022 in the RGB spectrum

2.2. Training data

The training dataset consists of 4 different classes and was created in Qgis by manually defining polygons for each class. The shapefile was then imported into the python script and band information extracted for each points from the image from after the collapse. The image from September 16th was chosen instead of the one from July as it has more area with tailings, which will make it easier for the model to classify this class.

Class Name	Class Id	N° of pixels
Water	0	11521
Land	1	6340
Urban	2	2676
Tailings	3	8601

Table 2. Training set

3. Methodology

Before starting to train the algorithm, the annotated data was split into training data (70%), test data (15%) and validation data (15%).

The algorithm used is Random Forest, an ensemble algorithm. A first step was to perform a grid search to find the best hyperparemeters to use for this case. The metric used to determine the best model is the accuracy score, which is how many times the model predicted the correct output out of all the predictions. The final model reached an accuracy of 81% as shown in the Table ?? below.

4. Results and analysis

The accuracy of the final Random Forest model is 81% on the test data (Table ??). But to have a better idea of how the model performs, we can look at the F1-score. This is the harmonic mean of the precision (proportion of positively predicted label) and recall (the model’s ability to predict the positives out of actual positives). We can see that the water class has a very good score - unsurprisingly as it has more than double the training data compared to other classes. The urban class is the one that doesn’t do very well. This can be explained as the urbanized area is very diverse as seen in Figure 3. In the RGB range, pixels can take multiple colors while other classes are more homogeneous. Another reason is that there are not many towns in the region of interest, and there are very small. This is reflected by the lower number of points in the dataset. Adding new points could help the model to get better.

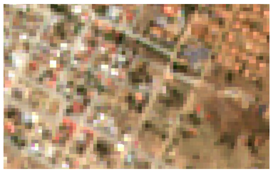

Figure 3. Close up of a urban area in Jagersfontain

This is confirmed by the normalized confusion matrix, where we see that **Water** and **Tailings** are the most correctly predicted classes, **Urban** is the least correct one.

The total number of pixels classified as tailings in the first image is 57’68’176. The total number of pixels classified as

	precision	recall	f1	support
Water	0.98	0.98	0.98	11521
Land	0.73	0.63	0.67	6340
Urban	0.64	0.46	0.54	2676
Tailings	0.70	0.82	0.76	8601
accuracy			0.81	29138
macro avg	0.76	0.72	0.74	29138
weighted avg	0.81	0.81	0.81	29138

Table 3. Classification on the test set

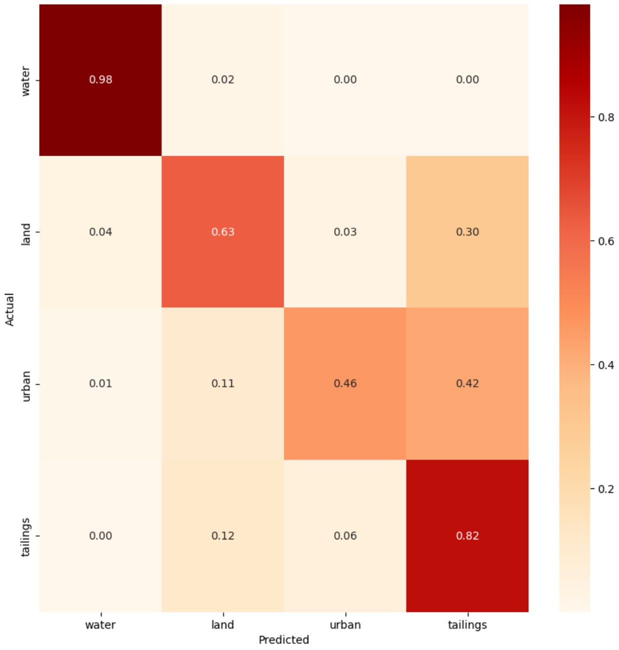

Figure 4. Confusion matrix on the test set

tailings in the second image is 133'48'874. The difference between them is 7'580'698 pixels.

5. Conclusion

In conclusion, the Random Forest did well at classifying the pixels. More things can further be added to this report, such as:

- solve the class imbalance problem
- add more points to the training data
- plot the pixels to visualize them geographically

Code is available at https://github.com/melindafemminis/ML$sentinel_Classification$
The Google Earth Engine script used to get the Sentinel-2 images is https://code.earthengine.google.com/f324b93024c9a6f8efd30fb9e63db37e