1. Introduction

Water resources are very important in daily life, agricultural and industrial activities. Therefore, the rational usage of water is necessary in order to continue these activities in their normal and healthy form. Dams play a key role in water management and they are used in many activities like providing irrigation water for agricultural fields and connecting cities with drinkable water etc. So keeping track of their filling rate is of great value when it comes to optimising and planning the distribution of the water contained in those Dams. Remote sensing technologies are used to capture images with various information depending on the bands used by the sensing instrument and artificial intelligence provides a wide range of tools to process this information. So In this work we tried to assess the possibility of monitoring the filling rate of Dams using Artificial intelligence and remote sensing images.

1.1 Problem statement:

The goal of this work is to do semantic segmentation of remotely sensed multispectral images of dams in order to detect the area filled of water in dams. Semantic segmentation is "a computer vision task in which the goal is to categorise each pixel in an image into a class or object. The goal is to produce a dense pixel-wise segmentation map of an image, where each pixel is assigned to a specific class or object."[1] The classes we consider in this project are : Water and Background. We mean by the latter any other object that is not water.

1.2 Related work:

The task of semantic segmentation was studied intensively in computer vision research. Some of the most implemented models used for this task are: UNet, Mask R-CNN and MobileNet[1] which were designed to deal mainly with normal images (RGB images). One important work on this specific task; segmentation of multispectral images was proposed in this paper[2] The authors propose a very interesting method for land cover classification based on 1D CNN and 2D CNN. They used features extracted from both Spatial and Spectral values of each pixel:

- Spectral : the values of a certain pixel in each spectral band (channel) of the multispectral remote sensing image.
- Spatial: the values of the pixels surrounding the current pixel (pixel of interest centred patches) in all spectral bands (channels)

So the model makes use of both spatial and spectral information in order to do the segmentation, this method was tested on Image of more than 200 channels and 17 classes (segmentation objects) and it gave good results.

The next section will delve in details into the project, then we will discuss the results and finish with a conclusion.

---

[1] https://paperswithcode.com/task/

[2] https://www.researchgate.net/publication/317053979_Learning_and_Transferring_Deep_Joint_Spectral-Spatial_Features_for_Hyperspectral_Classification

2. Materials and Methods

Given the good results obtained in this paper[3], we have chosen to assess the usage of the architecture proposed by those authors with some modifications. The implementation details are presented in this section.

2.1 Study area

In order to assess the implementation of this model in segmenting images of dams to detect the area filled with water, we chose to do our experiments on AlMassira Dam which is “a gravity dam located 70 kilometres south of Settat on the Oum Er-Rbia River in Settat Province, Morocco.”[4] An aerial image of the dam is presented in figure 1.

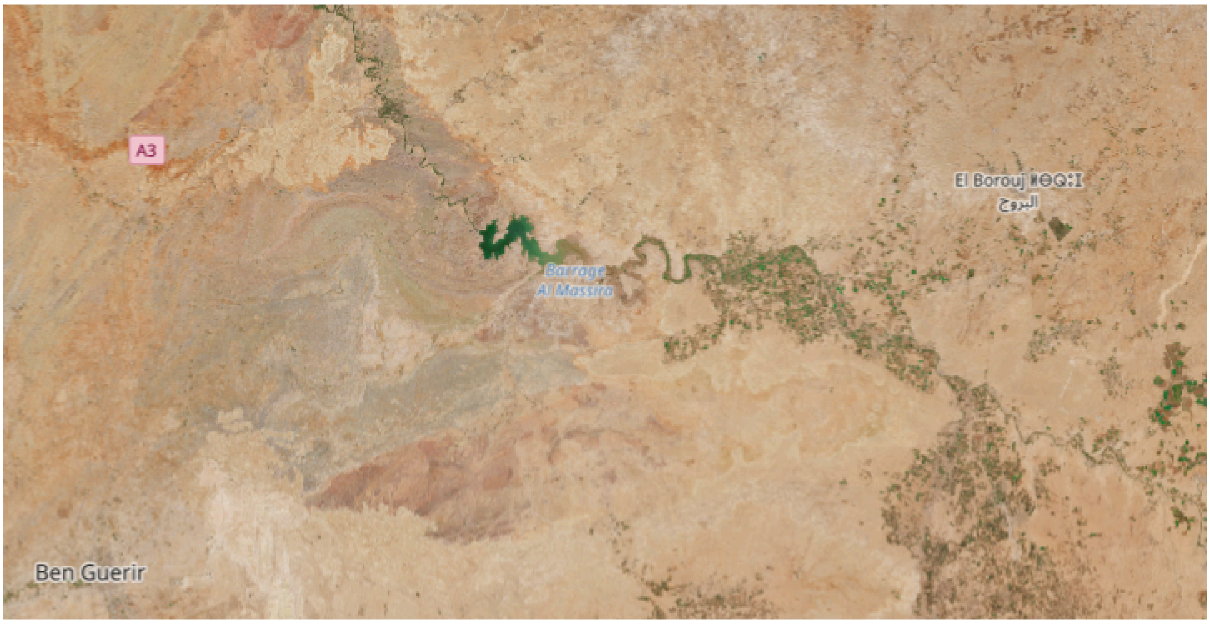

Figure 1, Al Massira dam image from copernicus browser

2.2 Dataset

To construct the dataset, images were collected from the copernicus hub project using sentinel 2 mission Images. The process from downloading the images until finished data points is presented in this section. We do an example focusing on the date 16/03/2024.

After selecting the area of interest, the date range is selected in the UI interface of the copernicus browser and then we run the search for available products as seen in figure 2. Two overlapping products were available for us to choose from, we have chosen the product where the area of interest is located in the upper side.

3 https://www.researchgate.net/publication/317053979_Learning_and_Transferring_Deep_Joint_Spectral-Spatial_Features_for_Hyperspectral_Classification

4 https://en.wikipedia.org/wiki/Al_Massira_Dam

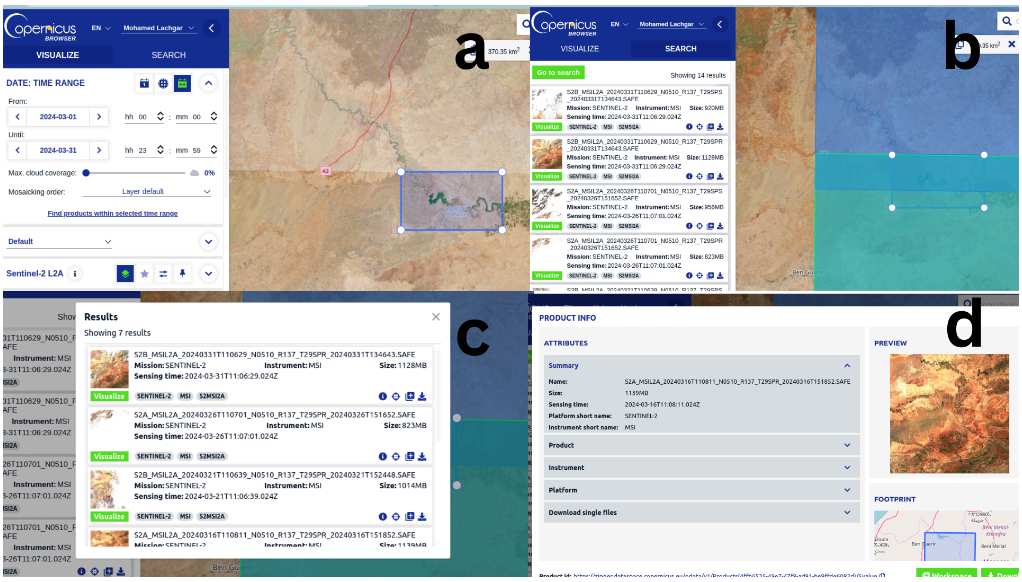

Figure 2: a) selection of the area and date range, b) choosing between products c) choosing the day, d) product info and download options.

Now, we can either download all the files in the product or just select the files we are interested in. In this work, we are interested in having as many channels (bands) as possible because the architecture of the model we are trying to adapt for this task was designed to handle multispectral images with a large number of bands (200+). So, we selected the images with 60 m resolution as we can get in total 17 bands in the multispectral image we are going to construct from each individual image. This is presented in figure 3.

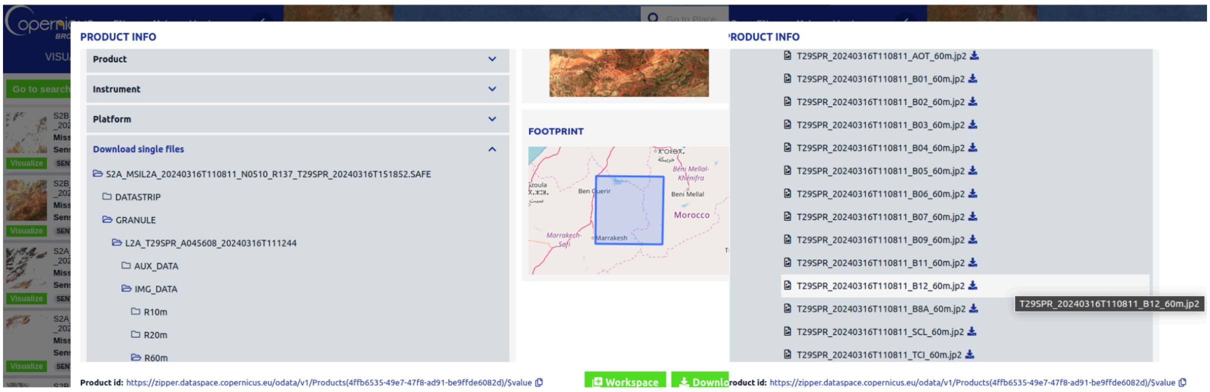

Figure 3: choosing the resolution and bands to download

We followed the same procedure for every day we considered in this study.

Now, we have the images in the different bands and a true color image (TCI) which is an RGB Image of the product containing the area of interest. However, the images are very large and cover way more than just the area of the dam, so we crop the images to a much manageable size that contains only the dam and a small area of the surroundings and we used an annotation tool[5] to label the area containing water as we see in figure 4.

[5] https://www.cvat.ai/

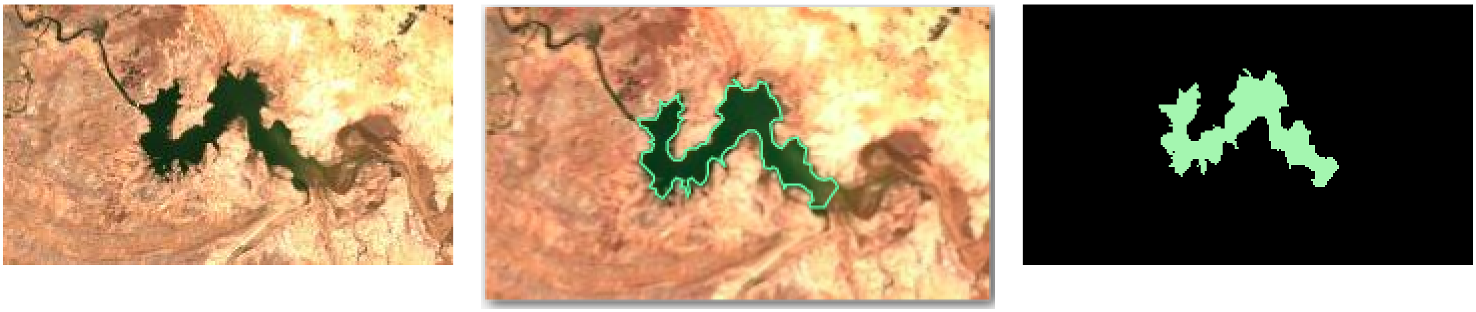

Figure 4: the cropped images we will work with, state of the dam in day 16/03/2024 and the segmentation mask

We selected available images from 01/01/2023 to 06/05/2024 to create our dataset and eliminated the images that had a cloud cover over the area of interest and corrupt images. We finished with 55 images in total. We used 50 images to construct the dataset and saved 5 randomly for testing the segmentation results as we will see later in this report.

After annotating the images, we used a python script to stack the bands for each image. Given that the model we are working with (presented in the next section) is segmenting based on individual pixels not the entire images by extracting the spectral vector and the spatial patch, we created a pytorch dataset where a data point is a tuple like this (sT,p,c) that is associated to a given pixel with sT is the spectral tensor with 17 spectral value, p is the spatial patch with size 11 centred with the pixel of interest and c is the class (0 for the background and 1 for water). The statistics of the generated dataset is presented in table 1.

The split was : 10% for testing, 72% for training and 18% for validation

Table 1: dataset statistics

	training	validation	testing	Total
Background	934488	233625	129852	1297965
Water	118512	29625	16398	164535

As one may expect, this is a highly imbalanced dataset because normally the area filled with water in the images of dams we are working with is very small compared to the area of the surrounding ground.

2.3 Model used and architecture

As already stated, we will adapt the model architecture proposed in the previously cited paper and presented in figure 5 with the needs of this specific task we are working on. The authors of that paper trained the model with a portion of pixels of a specific MSI image and then used it to segment the whole image, however, as one application of this work is to be able to track the filling rate of the dams automatically we will need multiple annotated images representing different days of the year of the same dam and prepare a dataset of pixels of classes: water and background. Then, we will use the trained model to segment multiple images of the same dam but on different days that ideally none of their pixels were used in the training process.

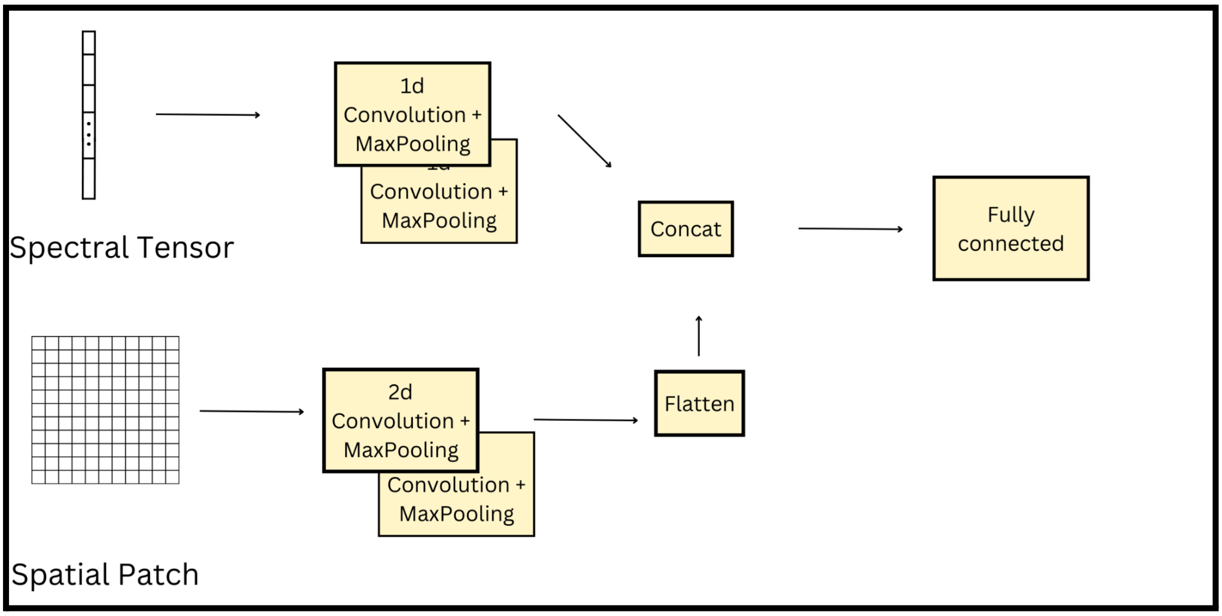

Figure 5: the model architecture

The first thing is transforming the model from multiclass classification model to binary classification model. To do that it is sufficient to set the number of nodes of the output layer to only one and to use the convenient loss function. Also, it was necessary to change the parameters of the architecture to suit the inputs of this specific task because we have a different number of channels (17 instead of 200+) and different size of the patch (11 instead of 21). The new parameters of the network architecture, the loss function and the optimization algorithm are presented in figure 6

```
MSIClassSpecSpat(
  (Conv1d_1): Conv1d(1, 20, kernel_size=(3,), stride=(1,))
  (Conv1d_2): Conv1d(20, 9, kernel_size=(3,), stride=(1,))
  (MaxPool1d): MaxPool1d(kernel_size=3, stride=3, padding=0, dilation=1, ceil_mode=False)
  (Conv2d_1): Conv2d(17, 30, kernel_size=(3, 3), stride=(1, 1))
  (Conv2d_2): Conv2d(30, 4, kernel_size=(3, 3), stride=(1, 1))
  (MaxPool2d): MaxPool2d(kernel_size=(2, 2), stride=(2, 2), padding=0, dilation=1, ceil_mode=
False)
  (FC1): Linear(in_features=43, out_features=256, bias=True)
  (FC2): Linear(in_features=256, out_features=100, bias=True)
  (FC3): Linear(in_features=100, out_features=1, bias=True)
)
```

```
loss_fn = torch.nn.BCEWithLogitsLoss()
optimizer = torch.optim.SGD(model.parameters(), 0.0001)
```

Figure 6 : parameters, loss function and optimizer

We trained the model for 4 epochs with batch sizes of 16.

3. Results and discussion

To assess the performance of the model, we considered the following metrics: accuracy, f1_score, recall and precision. The results of these metrics for both training and testing are presented in table 2. Recall that this segmentation is done by classification of individual pixels based on their spectral and spatial features, so the metrics are calculated for assessing the pixel's classification performance.

Table 2: training results

	Accuracy	F1 score	Recall	Precision
Training (average of 4 epochs)	98%	75% (0.759)	76%	77% (0.779)
Testing	98%	76%	76%	78%

The training accuracy and loss curves are presented in the figure 7

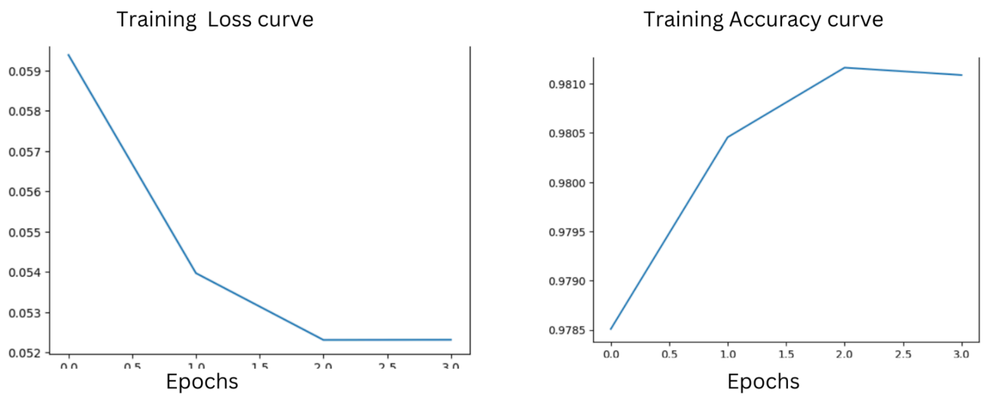

Figure 7: training loss and accuracy curves

As the dataset we are working with is highly imbalanced, unlike the f1 score measurement the accuracy - even though it is very high in this case - is not a very good indicator of the model performance[6] so we are using it just to see if the model is actually learning. A value of 76% of f1 score is still a good result and indicates that the model does well on this specific task. However, there is plenty of room for improvement and that could be done by using more annotated data or even increasing the number of classes to include for example but not limited to: a class representing the area of the dam's basin that was drought-stricken, the river or any other recognizable entity in the image.

After assessing the model's performance on the pixel's classification using metrics, a visual assessment is also required to see if we actually succeeded in segmenting the images to indicate the areas of water in the images. To do that, we run an inference using the resulting

---

6 https://www.kdnuggets.com/understanding-classification-metrics-your-guide-to-assessing-model-accuracy

model on the test images we have saved for just this purpose, in other words, the pixels of these images were never seen by the model in the training and they weren't a part of the testing dataset on which we had the results in table 2. The days that represent those images are : 31/03/2024, 11/01/2024, 28/09/2023, 08/09/2023 and 24/08/2023. The results are presented in figure 8

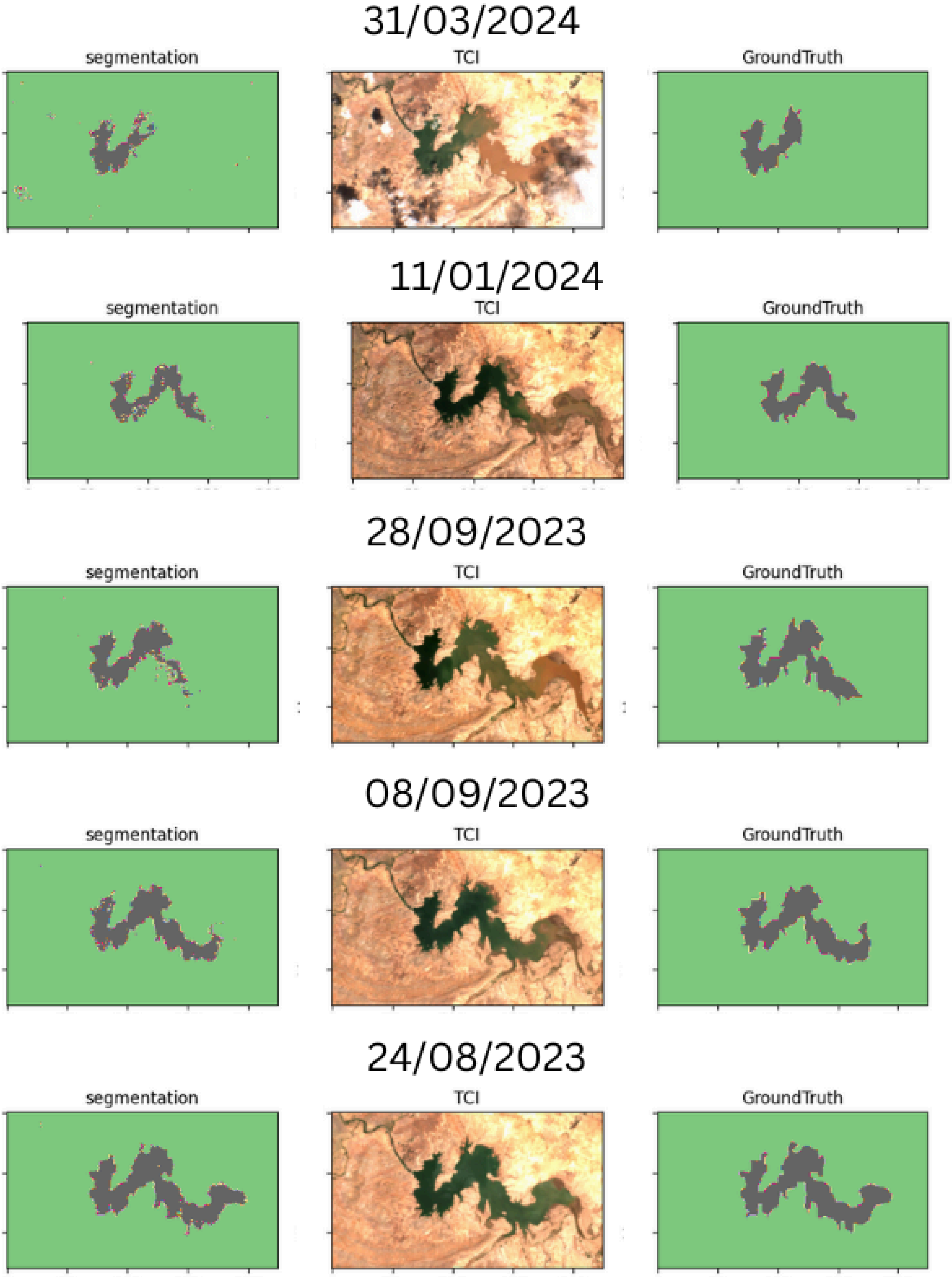

Figure 8: Segmentation of unseen images

As we can see visually in these figures, the model did a good job in segmenting the images.

4. Conclusion

The usage of both spectral and spatial features in segmenting multispectral images was successful on this task. However, the results could be improved further by using more annotated images and/or adding more classes among other techniques. The results of the segmentation could be helpful in estimating the filling rate of the dams remotely and other operational activities by the dams managers, for example the number of pixels that represent the water could be calculated and used to display charts to show the evolution of the water in the dam throughout the year. This work was done on a specific dam -Al Massira- but the exact same steps could be reproduced for other dams.