We first start with a time patch with its corresponding segmentation (Figure 1 and Figure 2).

The segmentation is then reduced to a single label as we are performing a classification task (1 if the time-patch contains a natural oil seepage, 2 if it contains an artificial oil spill). The time image is first processed through one of our time methods. The second step corresponds to the data augmentation, and then processed through the Deep Convolutional Neural Network (DCNN). Finally, the output of the DCNN is compared with the label.

We present in Figures 3 to 7 concrete examples of how the proposed time methods work. We added one of these Figures for each time-method to our revised version.

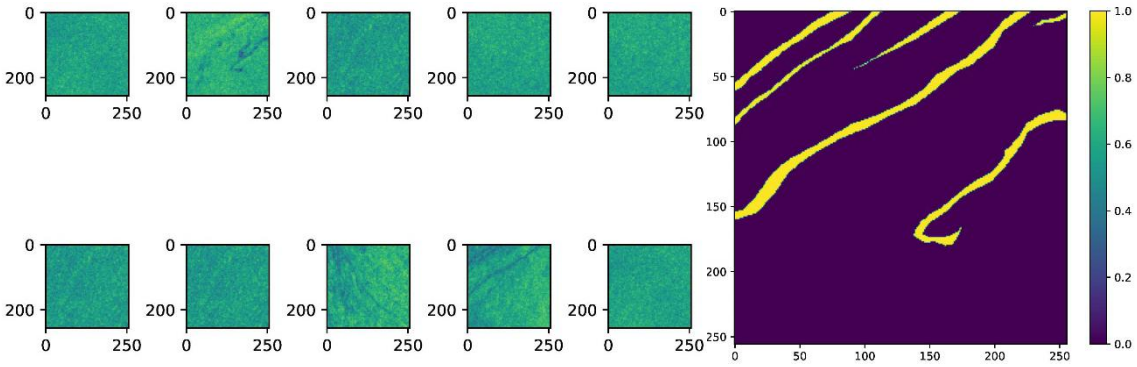

Figure 1: Exemple of a time patch and its unified segmentation. In this example, the single label obtained from the segmentation on the right would be 1 because the time patches contain a natural oil seepage (segmentation value 1).

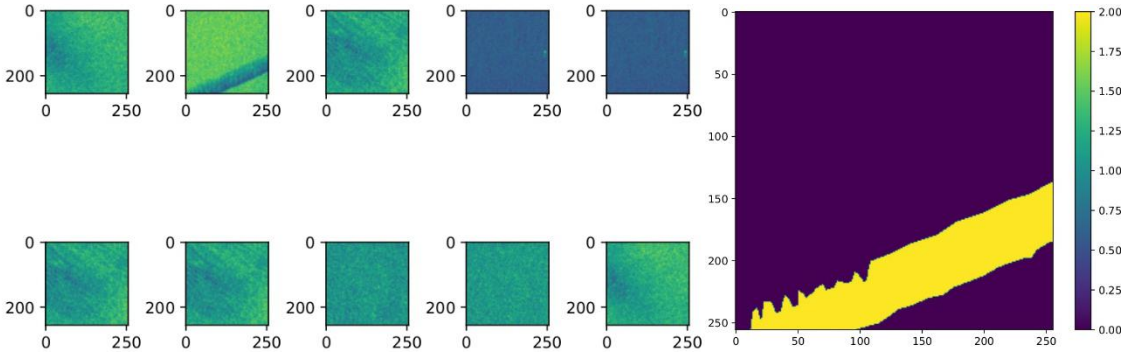

Figure 2: Exemple of a time patch and its unified segmentation. In this example, the single label obtained from the segmentation on the right would be 2 because the time patches contain an artificial oil spill (segmentation value 2).

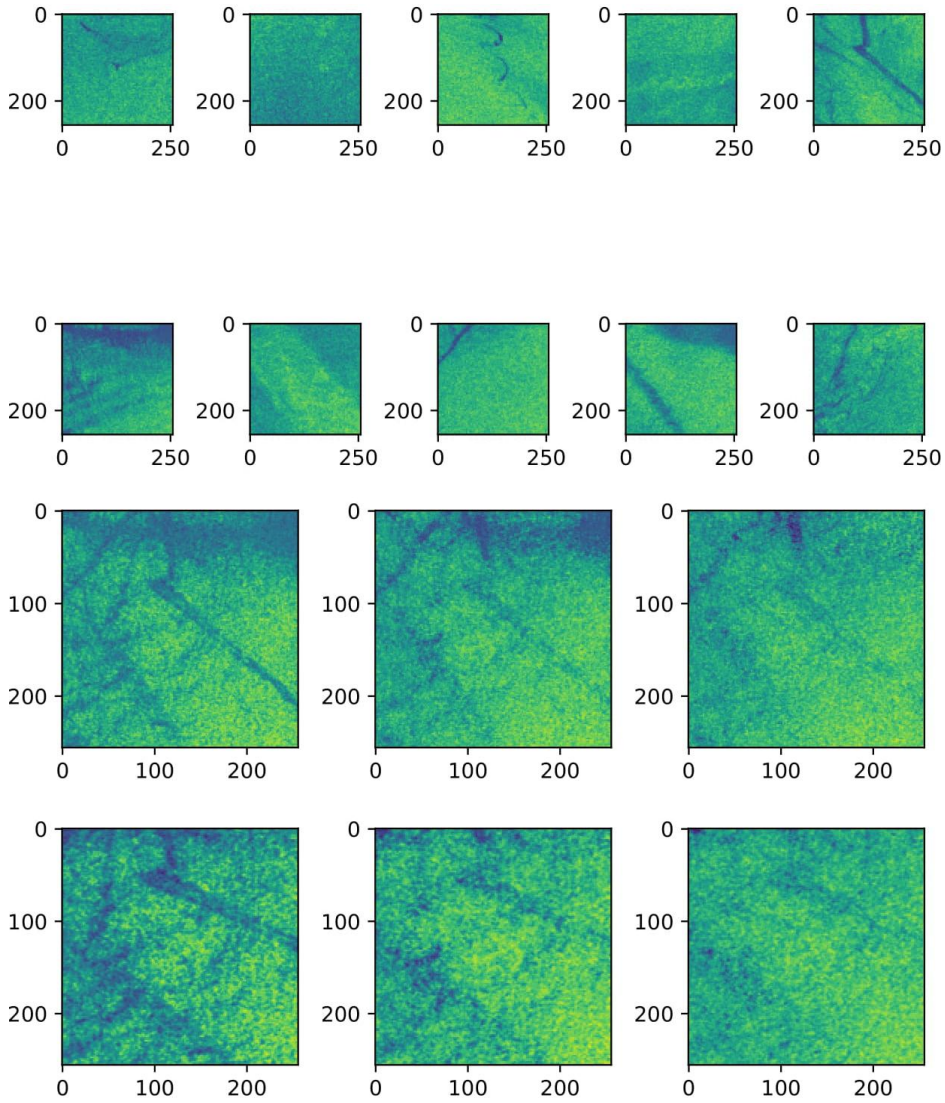

Figure 3: Exemple of a time-patch (upper image) going through the “sort” time method (middle image) and being data-augmented (lower image). As described, the “sort“ time method sorts the time images according to the time dimension, and only keep the three darkest pixel values.

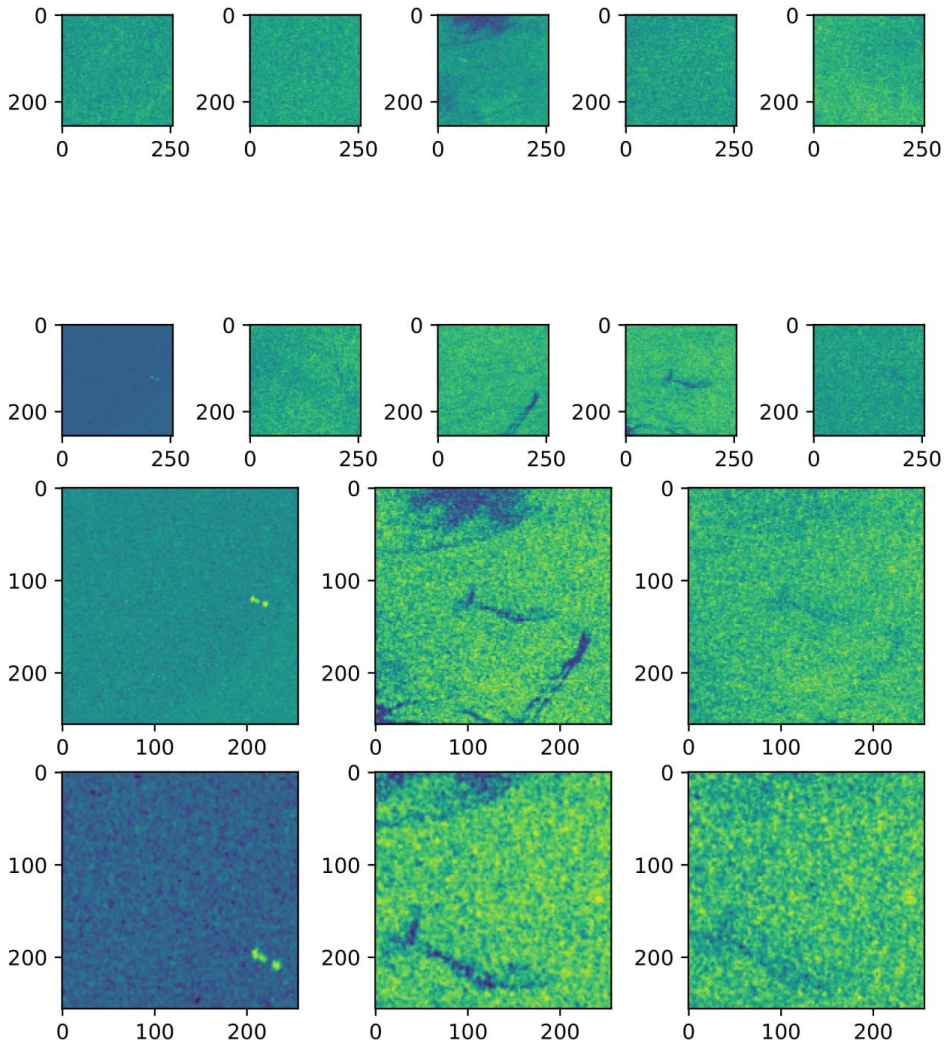

Figure 4: Exemple of a time-patch (upper image) going through the “statistical extraction” time method (middle image) and being data-augmented (lower image). As described, the “statistical extraction “ time method uses three complementary statistical tools. On this exemple, we can see that the first statistical tool: “variance” does not perform well because of a boat, but the third, and especially the second statistical tools are able to only keep important shapes that will be further processed by the neural network.

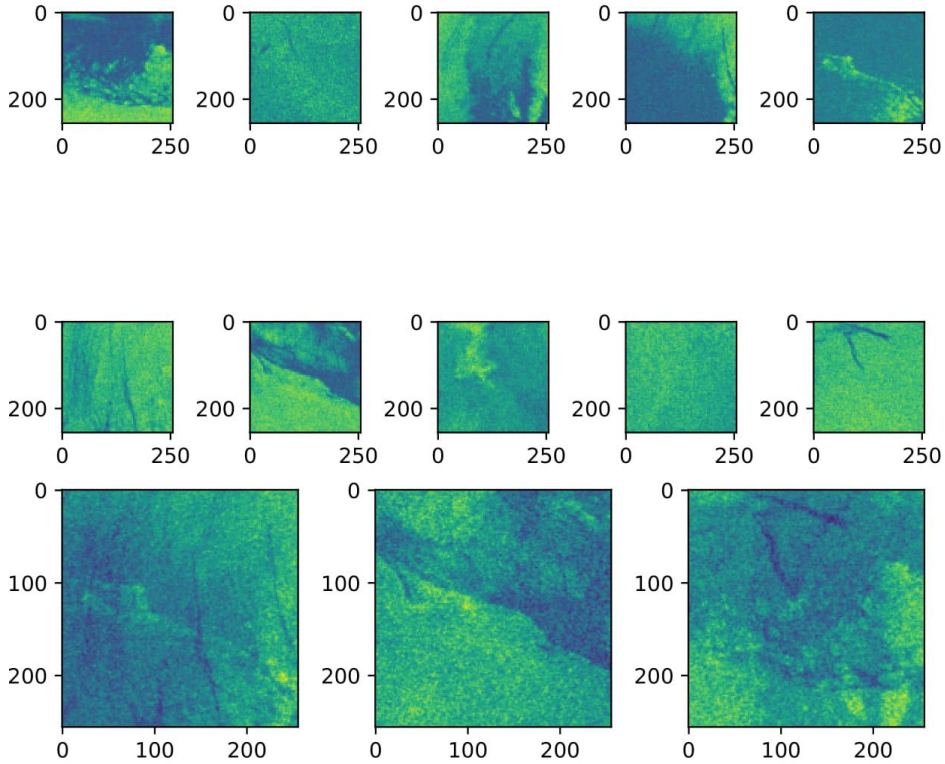

Figure 5: Exemple of a time-patch (upper image) going through the “average” time method (lower image). As described, the “average“ time method simply split the 10 images of the time-patch into 3 similar size clusters, and average the clusters into images. As expected, this method is not powerful enough to screen non meaningful shapes.

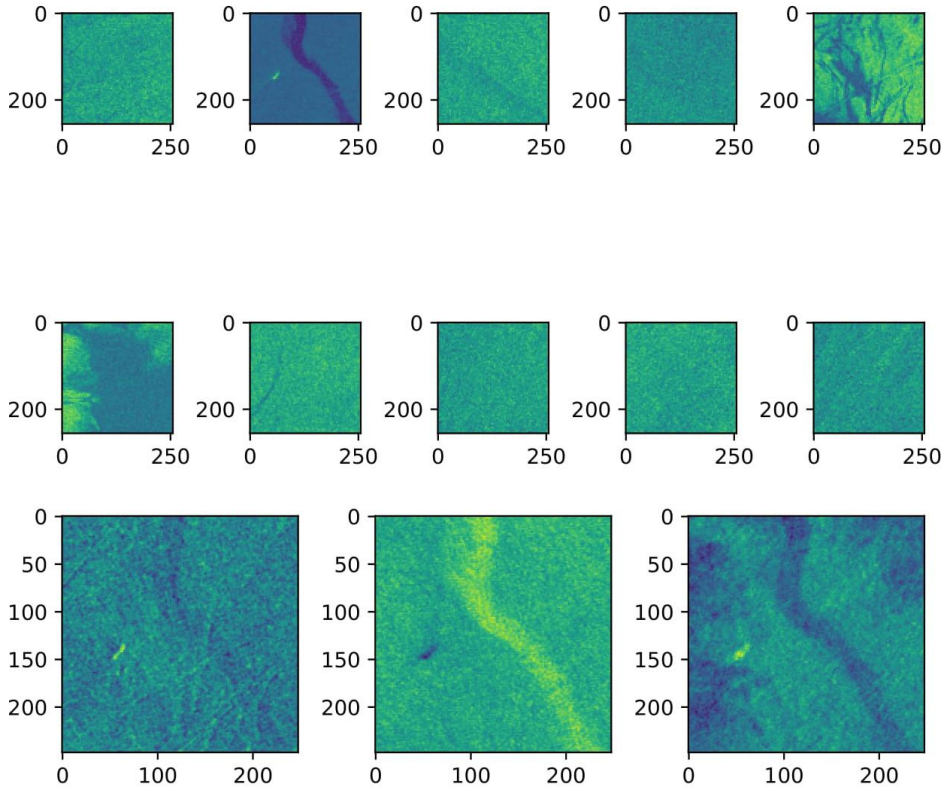

Figure 6: Exemple of a time-patch (upper image) going through the “full convolutional” time method (lower image). It is very clear here that the convolution extracts meaningful features and evicts lookalike forms.

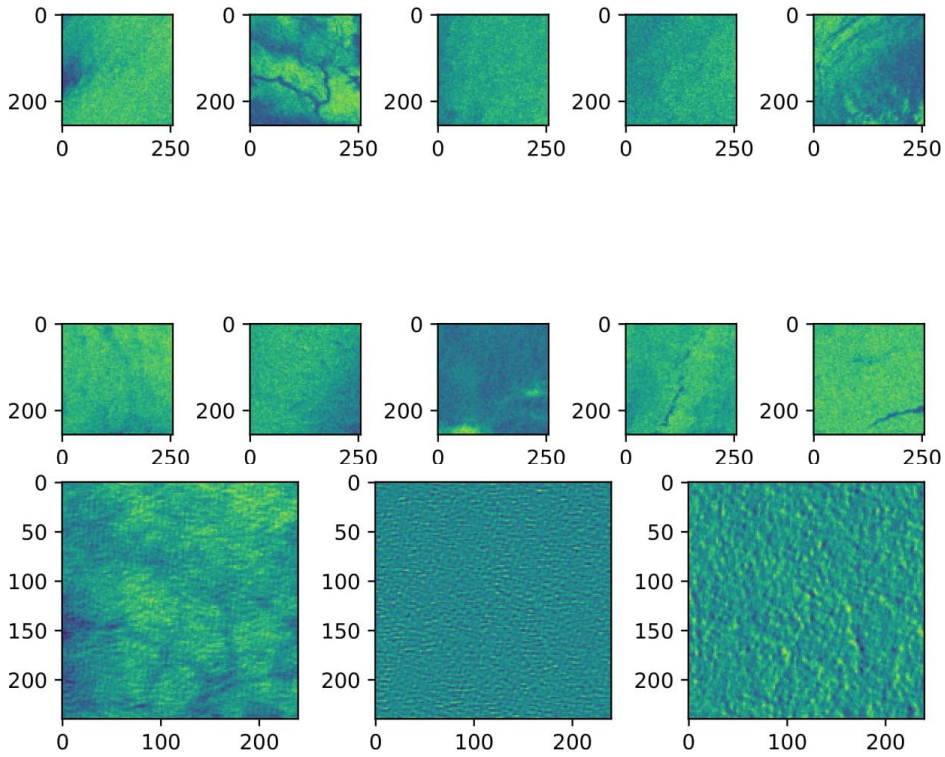

Figure 7: Exemple of a time-patch (upper image) going through the “constrained convolutional” time method (lower image). Unfortunately, convolutions can be less interpretable, but it is seems here that the first and third lower images capture the important forms.

Other time methods are not returning images with 3 channels and hence cannot be visualized, but as they are extensions of the constrained and full convolutional methods shown here, We can expect a similar behaviour.