**Change detection in remote sensing images using semantic segmentation**

Change detection in remote sensing images is an important task for many application. Some example application for such systems include tracking new buildings in suspicious areas for military purposes, tracking deforestation for climate change research, and even XXXXXX.

The database contains pairs of images that were taken at the same geographical location at two different times. We define the change detection tasks to identify differences between 2 photos that are a result of appearance of new or disappearance of existent objects in a scene. The images have differences between them that are a result of many other factors such as seasonal changes (snow, trees with/ without leaves and many more), changes in brightness, shifted images due to slight changes in the capture angle and many more.



Image 1: example of a pair of images that has changed a great amount due to seasonal changes (snow) which should not be recognized as a change in our system.

Our project is based on a previously published paper which implemented the described task using a Generative adversarial networks (GAN). In this project we choose to use a discriminative model to solve the same task.

Generative adversarial networks is a class of machine learning systems in which two neural networks contest interact with each other in order to mutually improve. Given a training set, this technique learns to generate new data with the same statistics as the training set. Traditionally, this technique was used for unsupervised tasks, though it quickly spread to works done on semi and even fully supervised learning tasks.

Lebedev et. al (2018) used a generative model to generate a binary change map. Then another network was trained to distinguish between the real change maps and the fake ones generated by the GAN. This input was propagated back to the GAN network, further improving the quality of the change maps, making them similar to the ground truth maps.

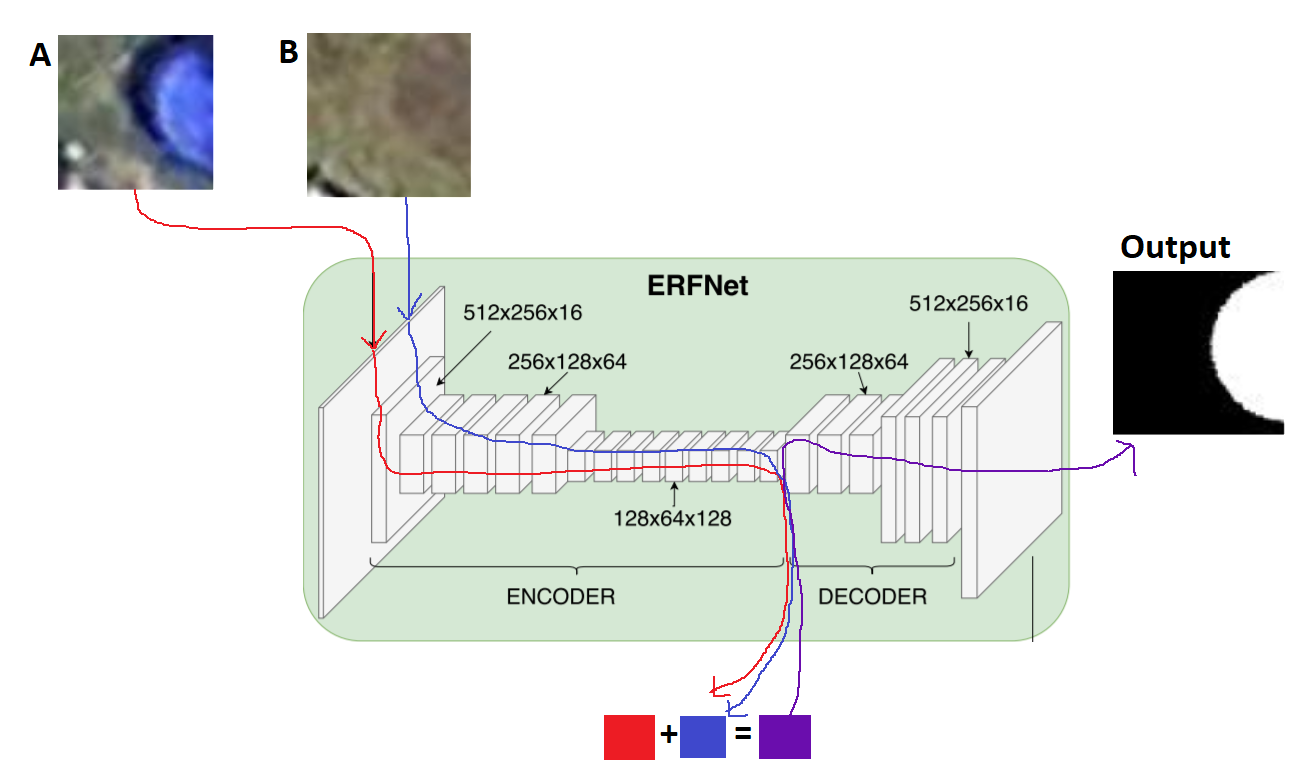
We chose to solve the change detection task using a convolutional neural network and semantic segmentation. Semantic segmentation refers to the process of linking each pixel in an image to a class label. Semantic segmentation does not differentiate between different instances of the same class, but simply links a label class to each individual pixel. We use this technique here with two class labels: change and no change. The semantic segmentation model is simpler and requires less computational demands, and is therefore a significant improvement, given that they are equally successful in change detection.

ERF

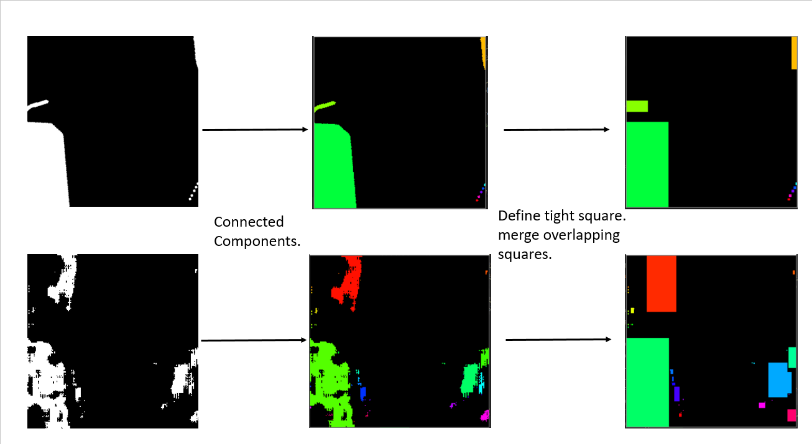
We based our network on the ERFnet (Efficient Residual Factorized ConvNet for Real-time Semantic Segmentation). The ERFnet introduces a new layer which combines the use of residual connections and factorized convolutions in order to remain efficient while retaining remarkable accuracy.

Siamese network

Since the input of our system is composed of two images, we decided to create a Siamese neural network. A Siamese neural network is a network in which one set of weights is used for the analysis of 2 different input vectors. Then, the information received from the network is combined and inserted into another part of neural network as one object. In our project we used the encoder of the network to analyze each input image separately. We then combined the information gathered by concatenating the 2 outputs of the encoder, and fed the concatenated image to the decoder. Image XXX shows the assembled network.



In order to check the performance of our system and compare it to the performance of the GAN system (XXX), we calculated average precision and recall on the detected changes. To do so, first we used the connected components algorithm in order to extract connected regions from the ground truth labels and from the difference map synthesized by our network. We then defined a tight square area around each connected region in which we would then calculate the IOU. overlapping squared regions were merged in order to prevent the calculation of a certain area multiple times, thus increasing its weight in the total average. We define an area to be detected if IOU is greater than some threshold. Then, for the obtained classification values, the average values of Precision and Recall were calculated for the entire test dataset.



Results:

We trained the network for 150 epochs on 10,000 training images. We then used the test set to evaluate the performance of our network.

In order to understand the general performance of our model we calculated the average IOU (Intersect over union) across all images in the test set. It was found to be 53%. While this was not done in the original GAN paper and is therefore not used for the comparison of the two models, it is useful to gain a general understanding on our system. This means the 53% of the pixels in the test data were labeled correctly. While this is not a very high value, if we take under consideration that the purpose of this system is to find changes and report them, and not necessarily to create a perfect change map- this value could be easily sufficient. In addition, since the IOU value is used to calculate the precision and Recall metrics, it was important to make sure that the thresholds used in the GAN paper make sense for the testing of our system as well.

As shown in Figure XXX, for IOU thresholds of 0.3, 0.1, the recall values calculated for our model were higher than the values calculated for the GAN model (figure XXX). For the IOU threshold of 0.1 the system had a recall value of 0.987, meaning that almost all changes in the ground truth were in fact found. For the IOU threshold of 0.5 our model produced a lower recall value than the one calculated for the GAN model. This could indicate that though our system is highly likely to find part of every change, and it is able to notify with high confidence that there was a change in a certain area of the image, it does not necessarily find the whole area of the change.

As expected, the lower the IOU threshold, the higher the Recall and Precision values were. This is because the trained network does not often find the entire area in which the change has occurred but only some of it.