Published in Image Processing On Line on 2022–11–26.
Submitted on 2022–10–11, accepted on 2022–10–11.
ISSN 2105–1232 © 2022 IPOL & the authors CC–BY–NC–SA
This article is available online with supplementary materials,
software, datasets and online demo at
https://doi.org/10.5201/ipol.2022.439

A Deep Learning Model for Change Detection on Satellite Images

Elyes Ouerghi

Université Paris-Saclay, ENS Paris-Saclay, Centre Borelli, Gif-sur-Yvette, France
elyes.ouerghi@ens-paris-saclay.fr

Communicated by Jean-Michel Morel *Demo edited by* Elyes Ouerghi

Abstract

Change detection is a classical problem in satellite imaging. The change detection problem aims at the analysis of changes between two images. In this work, we test a deep learning model proposed in 2019 by Caye Daudt et al. on data from the Sentinel-2 satellite with images between 10 m and 60 m of spatial resolution. The model uses the early fusion technique combined with a U-net architecture and can be used on color or multispectral images. The tests are performed on the Onera Satellite Change Detection (OSCD) dataset, which was already used for testing deep learning methods for the change detection problem. Here we propose some experiments to evaluate the performance and limits of the algorithm by Caye Daudt et al.

Source Code

The source code and documentation for this algorithm are available from the web page of this article[1]. Usage instructions are included in the `README.md` file of the archive. The original implementation of the method is available here[2].
This is an MLBriefs article, the source code has not been reviewed!

Keywords: deep learning; change detection; satellite; remote sensing

1 Introduction

Change detection is a classical problem in satellite imaging. The changes to be analyzed can be various: cloud detection, appearance of roads and buildings, deforestation monitoring or agricultural crop monitoring. In practice, we consider two co-registered images I_1 and I_2 of the same area and of the same size, $N \times M$ pixels, and we want to compute a change map CM. The change map CM is a binary map where the value of a pixel is 1 if a change between I_1 and I_2 is detected and 0 otherwise. Here, we will work with data from the Sentinel-2 satellite, with images with a spatial resolution

[1] https://doi.org/10.5201/ipol.2022.439

[2] https://github.com/rcdaudt/fully_convolutional_change_detection

between 10 m and 60 m. We will use the Onera Satellite Change Detection (OSCD) dataset [2], which is already used for testing deep learning methods for the change detection problem.

In this paper, we present a method proposed in [4], which uses a convolutional neural network for change detection on the OSCD dataset. First, we present the algorithm and the dataset on which it was trained and tested. Then, we report some experiments and tests we performed to evaluate the algorithm. In particular, we present other tests than the one already included in the OSCD dataset. Lastly, we provide some information about the demo of the method linked to this paper.

2 Algorithm

The algorithm proposed in [4] is a convolutional neural network which uses the early fusion technique combined with a U-Net architecture. The architecture of the proposed network is named FC-EF-res.

Before passing each pair of images through the network, they are pre-processed. The images are normalized by their mean and standard deviation. Then, the images are concatenated along the spectral axis, meaning that each image of the pair is treated as a different color channel. This last step is usually referred to as *early fusion.*

The proposed architecture is based on the classical U-Net model [6]. The encoder-decoder architecture with skip connections is combined with residual blocks [5]. Skip connections help to solve the degradation problem, induced by the encoder-decoder architecture, by linking together the layers with the same subsampling scale in the encoder part and the decoder part of the network. Skip connections also allow to combine the abstract information of the last layers and the spatial details contained in the first layers.

The residual blocks also include skip connections. The idea is to divide the network into several blocks. At the beginning of each block, the input x of the block is saved. Then the input goes through each layer of the block, producing the output $F(x)$, where F regroups all the transformation made by the block. At the end of the block the input and the output are added, meaning that we compute $F(x) + x$. An example of residual block is shown in Figure 1. Besides helping with the degradation problem, these blocks facilitate the training of the network.

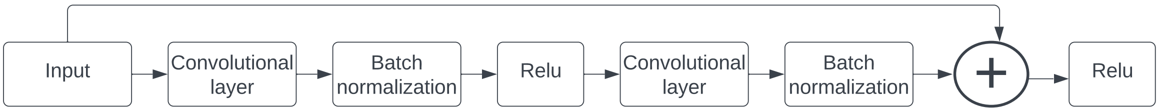


Figure 1: Architecture of the standard block used in the model. The upsampling and subsampling blocks follow the same architecture but with an additional upsampling or subsampling layer.

Here, three kinds of blocks are used: standard residual blocks, subsampling residual blocks and upsampling residual blocks. Each standard block is composed of three types of layers which are: convolutional layer, batch normalization and ReLU. Because the blocks are residual blocks, the input of a block is added to its output. The precise architecture of the standard block is depicted in Figure 1. Subsampling blocks are based on the standard block architecture but with an additional max pooling layer before the second convolutional layer. Upsampling blocks are also based on the standard block architecture but with a transposed convolutional layer at the beginning of the block. The full architecture of the network is shown in Figure 2.

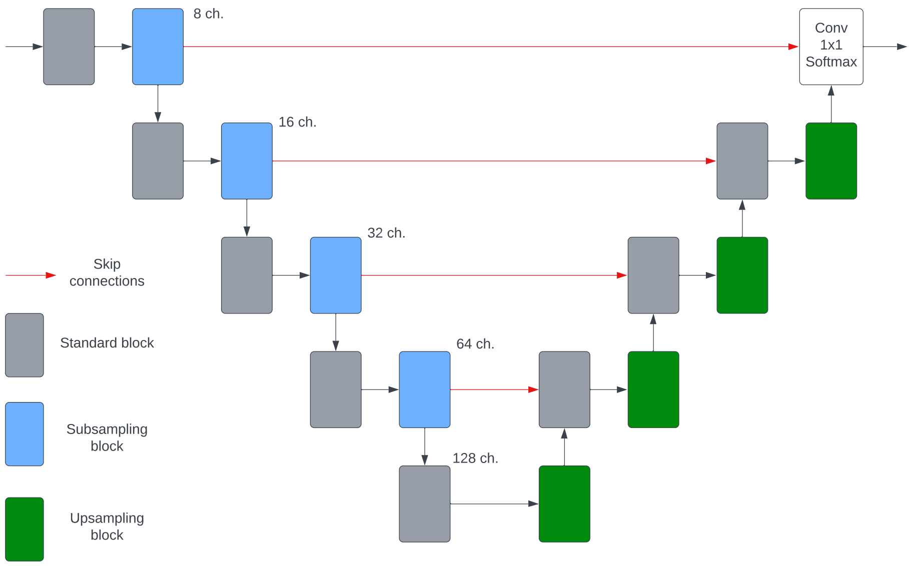


Figure 2: Architecture of the model which is a standard U-net architecture. Before passing through the first block, the images are concatenated along the spectral axis.

3 Dataset

For training and testing the network architectures, we use the OSCD dataset [2]. The dataset is already split between training images and testing images. It contains 24 pairs of Sentinel-2 images acquired between 2015 and 2018. All the pairs are already co-registered and of the same size.

Each Sentinel-2 image is a multispectral image composed of 13 bands. The spatial resolution of those bands varies between 10 m, 20 m and 60 m. However it should be noted that bands 2, 3 and 4, which are the ones located in the visible spectrum, are at the same spatial resolution of 10 m.

The training set provided is composed of 14 of the images pairs. The changes were annotated manually and are mainly urban changes, such as new buildings and new roads. However, the dataset also contains other kind of changes, for example in the images from Saclay, in France, one can see changes in agricultural land.

4 Results

We compare the results of the method presented here with other architectures tested on the same dataset [1, 2]. The network is trained on the OSCD dataset [2] in two different cases. In the first case the network is trained with the 13 bands available on the Sentinel-2 images. In the second case, we use only the 3 bands that belong to the visible spectrum, which are bands 2, 3 and 4. These bands also have the advantage of being at the same spatial resolution. The advantage of having the network trained with only the 3 bands of the visible spectrum is that we can eventually use it to detect changes on other images than those of Sentinel-2.

Here, we tried to reproduce the result of the architecture named FC-EF-res [4]. To avoid any confusion, we will refer to our training of the algorithm as FC-EF-res 2. Since we used the same architecture as FC-EF-res, the only differences between FC-EF-res and our training are the parameters used. When reproducing the results, the parameters used in [4] where not available. Results are summarized in Table 1 for the 13-bands case, and in Table 2 for the 3-bands case. In those tables we compare the results of FC-EF-res with other architectures. The first one is Siam [3] which is a

Siamese network. The second one is EF [3] which is very similar to FC-EF-res: it uses the same technique of early fusion coupled with a U-net architecture. However, it is not a fully convolutional network as FC-EF-res because it has fully connected layers. Moreover, there are no residual blocs in EF. Finally, FC-EF [1] is the same architecture as FC-EF-res but without the residual blocs.

Network	Precision	Recall	Global acc.
Siam [2]	24.16	**85.63**	85.37
EF [2]	28.35	84.69	88.15
FC-EF [1]	**64.42**	50.97	**96.05**
FC-EF-res [4]	54.93	66.48	95.64
FC-EF-res 2 [4]	56.86	55.16	95.56

Table 1: Change detection results on the OSCD dataset. Networks are trained with all the 13 bands. The scores are in percent. FC-EF-res does not outperform Siam in recall or FC-EF in precision but it offers a good compromise between recall and precision.

Network	Precision	Recall	Global acc.
Siam [2]	21.57	79.40	76.76
EF [2]	21.56	**82.14**	83.63
FC-EF [1]	44.72	53.92	94.23
FC-EF-res [4]	**52.27**	68.24	**95.34**
FC-EF-res 2 [4]	45.26	58.16	94.26

Table 2: Change detection results on the OSCD dataset. Networks are trained with the 3 color bands. The scores are in percent. FC-EF-res outperforms all the other methods except Siam and EF in recall. However, it has a way better precision with a decent recall.

From these two tables, we can see that the method proposed in [4] mainly produces an improvement in the case of 3 bands. In the case of 13 bands, FC-EF-res exhibits worse accuracy and worse global accuracy than FC-EF. However, the method still offers a good compromise between precision and recall.

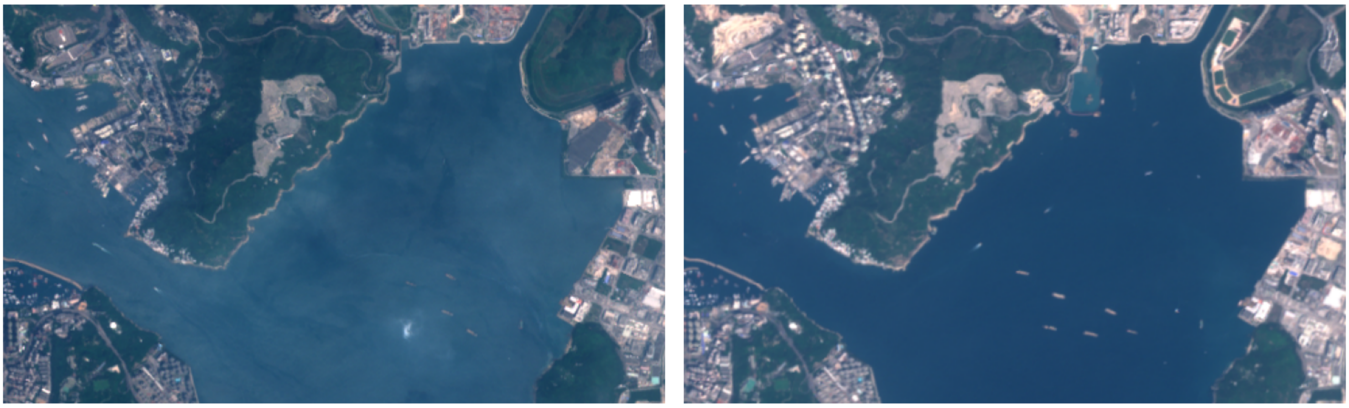
In the case of 3 bands, the FC-EF-res architecture clearly improves the results with the best scores in precision and global accuracy. FC-EF-res does not reach the best recall but the Siam and EF architectures have in return a precision twice as low.

Besides that, we can see that with FC-EF-res 2 we do not manage to reproduce exactly the same scores as FC-EF-res. This is probably due to a difference in the training parameters. The improvements visible in the results of FC-EF-res are still visible in the results of FC-EF-res 2. In the case of 13 bands FC-EF-res 2 has the best accuracy after FC-EF, and a better recall than FC-EF, just like FC-EF-res. In the 3-bands case FC-EF-res 2 has the second best accuracy and the second best global accuracy.

5 Experiments

To analyze the results and limitations of the algorithm presented here, we perform several tests on the OSCD dataset. The experiments presented here have been performed on the images using the 3 visible bands.

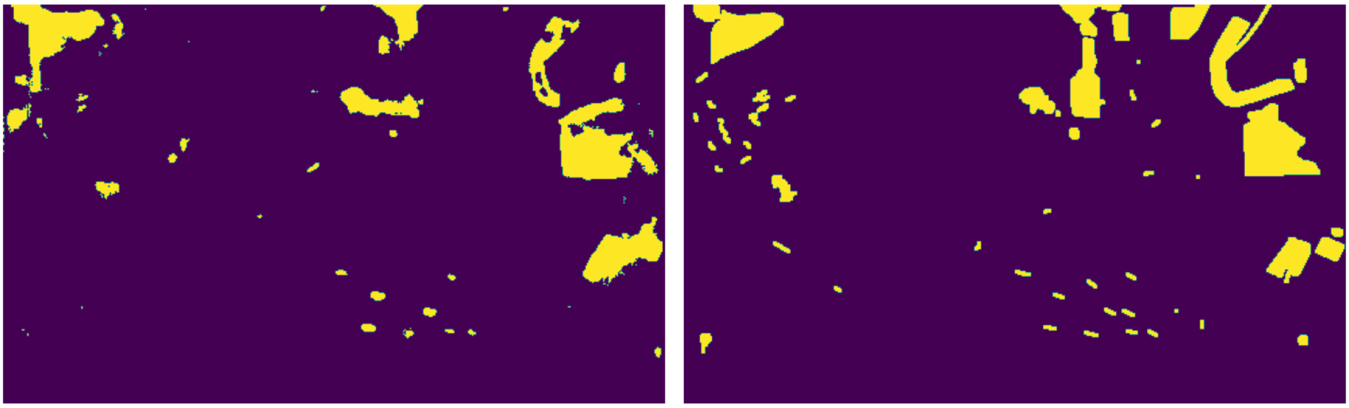
In Figure 3 we can observe an example of a pair of images of the dataset OSCD. Here, the images were acquired over Hong Kong. We can observe several types of changes on this image pair: for example the appearance of roads and buildings in the upper right corner of the second image. But



(a) First image

(b) Second image

Figure 3: A pair of image of the OSCD dataset over Hong Kong. Many urban changes are visible, in particular in the upper left and upper right corners. However, other types of changes are present such as the appearance of boats.



(a) Change map computed by the network

(b) Ground truth

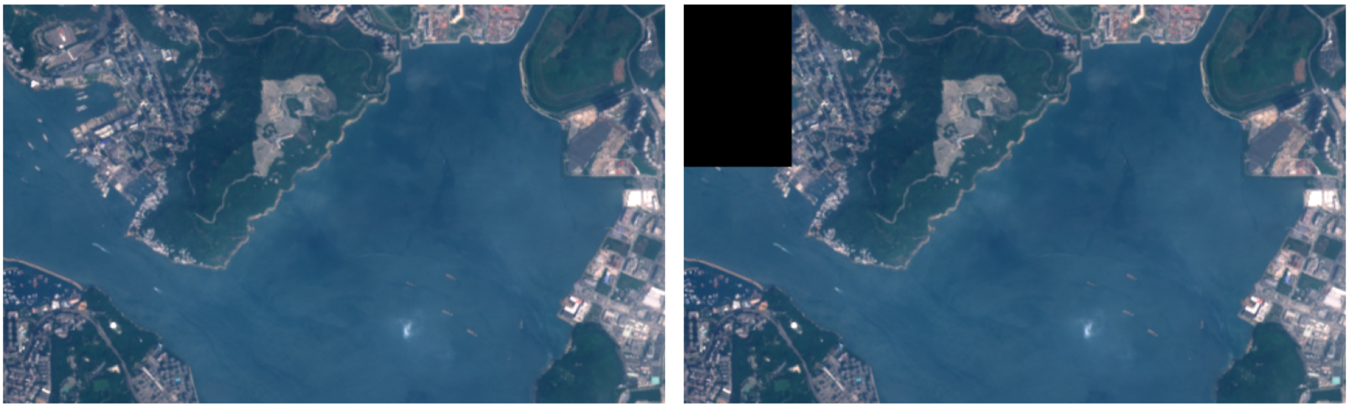
Figure 4: Comparison between the output of the network and the ground truth for the Hong Kong image pair.

there are also temporary changes between the two images such as the boats, which are visible at the bottom and left parts of the two images.

Figure 4 shows the change map obtained by the network, together with the ground truth. It can be seen that an important part of the changes are detected (63% recall), in particular on the right side of the image most of the changes are detected. It is the smallest changes that are the least well detected, such as the boats. Among the large changes that are not detected we can especially note the development of the harbor at the top of the image. This element is not detected but this can be explained by the fact that, even if the port is well distinguished on the second image, the colors of this area have hardly changed between the first and second images.

We can also see that the detection mask given by the network is much less sharp than the one given by the ground truth, which results in a significant proportion of false positives (28% precision). In particular, on the right side of the image we can see that in the ground truth we can distinguish several areas of change, however the output of the network presents enlarged areas that correspond to merged regions present in the ground truth.

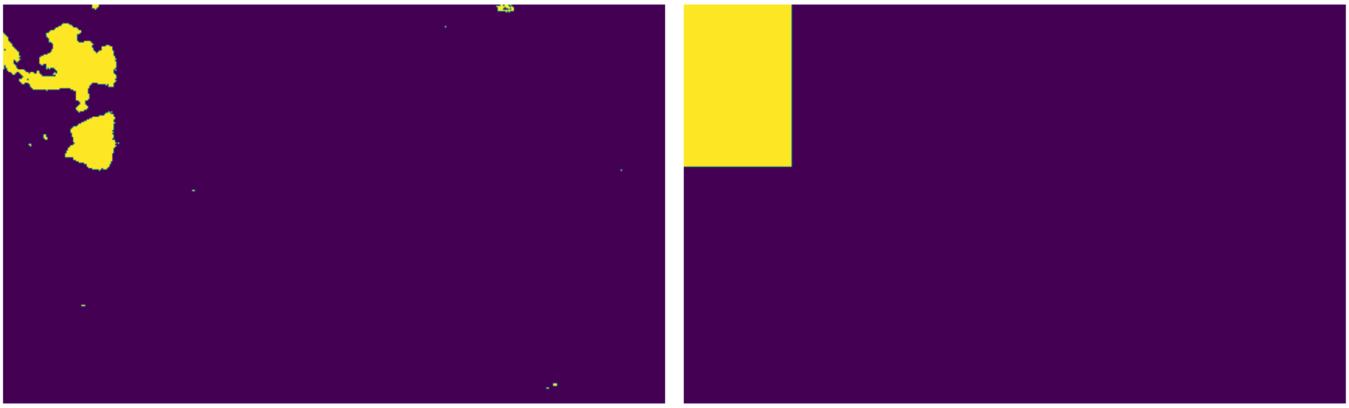
To illustrate the shortcomings of the architecture presented here, we will perform a very simple experiment by synthesizing a fake image with changes. We take the Honk Hong image presented earlier and set all the values in the upper left corner of the image to 0. This gives us the pair of images shown in Figure 5. Although this pair of images is not a real test, it is reasonable to expect the change to be relatively well detected because it is extremely simple.



(a) Original image

(b) Modified image

Figure 5: Pair of images over Hong Kong. Images are exactly the same except for the top left corner.



(a) Change map computed by the network

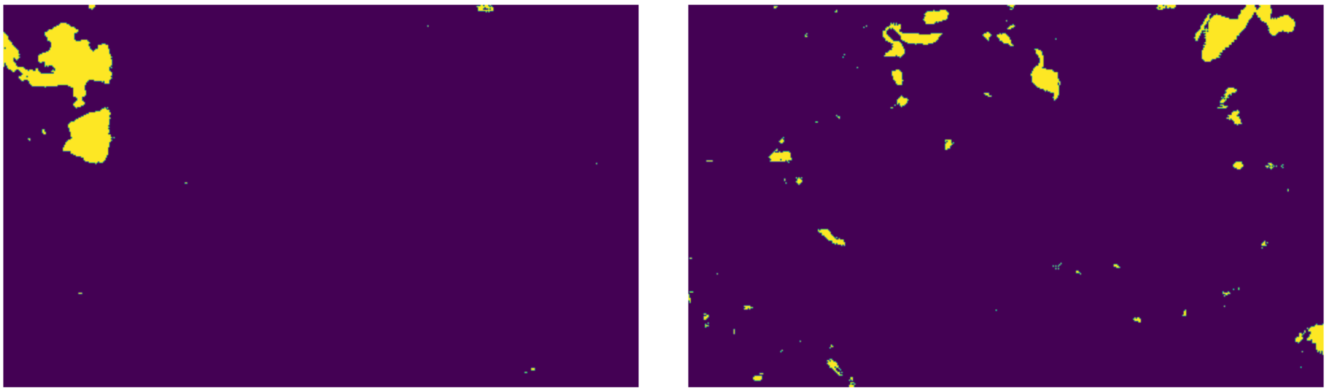
(b) Ground truth

Figure 6: Result of the network on the synthetic change over Hong Kong. Only a part of the black rectangle has been detected. Some of the detections are located in areas where the two images compared are identical.

Figure 6 depicts the output of the network. Note that only a part of the black rectangle is detected as a change. The detected area has a precise shape. This shape seems to be correlated to the shape of the coastline. In particular, the detected region is the one where the image has the most variance in terms of colors and patterns. On the other hand, on the uniform areas, here the forest and the sea, no change was detected. It is reasonable to assume that the network does not detect intensity changes on uniform areas, to be robust to lighting variations, for example.

A more problematic flaw is the fact that there are false positives elsewhere in the image. This means that the network can detect anomalies on pixels that are identical. One could argue that the image parameters are not identical because of the change in the top left. However, the experiment was also performed using the same image twice and we also got several pixels detected as a change. The fact that the network detects anomalies while observing two identical images can be explained by its architecture. The two images are treated as independent channels by the network so the weights learned can vary from one channel to another. In particular, this means that the network will not compute the same features for the two images. By comparing two different features from the same patch, it is not impossible to detect a change.

One observation that follows from the fact that the network learns different weights for image 1 and for image 2 is that we may not obtain the same result depending on the order in which we give the images to the network. This can be seen in Figure 7. In Figure 7a we gave the original image then the modified image, and in Figure 7b we reversed the order. We can see that the detected pixels are completely different. In particular in Figure 7b we have almost exclusively false positives. The artificially modified area is not detected at all by the network.



(a) Change map when using the original image before the modified one.

(b) Change map when using the modified image before the original image

Figure 7: Result of the network on the synthetic change over Hong Kong when switching the order of the images. The detections obtained in the two images are almost all distinct.

We can assume that these problems could be solved by changing the training of the network. For example by training the network by including images without changes, or by randomly swapping the order in the image pair. Another possibility would be to change the architecture of the network to make it invariant to permutations. A first step to achieve this is to use a Siamese network architecture, as presented by [1, 3]. In a Siamese network both images pass through the same encoder separately. To ensure invariance by permutation the decoder must also be the same for both images. In the Siamese network, the images are concatenated before passing each layer of the decoder. There is therefore no guarantee of symmetry.

6 Conclusion

We observe that the network architecture proposed by [4] allows to improve the existing results on the OSCD dataset. In particular the architecture succeeds in standing out when testing the dataset on color images. However, we have also shown some defects of the network presented here. For example, the network is not robust to input permutations and can in some cases give many false positives. Several possibilities can be explored to improve the results. We can modify the training step of the network with a different data augmentation. We can also change the architecture of the network to use the same encoder for both images. This would mean moving towards a Siamese network architecture as presented by [1, 3].

7 Demo Tutorial

The demo linked to this paper takes as input two color images in PNG format. Both images should be satellites images of the same area, and co-registered. The demo works best on images from the Sentinel-2 satellite. If both images do not have the same size, the images will be automatically cropped. The demo expects values stored on 8 bits. If the input images do not match this requirement, the images will be converted to 8 bit.

The output image is a change map. For each pixel in the input images, the value of the change map is 1 if a change is detected and 0 otherwise.

Pairs of images from the OSCD test set are provided with the demo. For those images, the ground truth is available in the original dataset[3].

Image Credits

All images are produced by the author and come from the processing of Sentinel-2 images.

References

[1] R. Caye Daudt, B. Le Saux, and A. Boulch, *Fully convolutional Siamese networks for change detection*, in IEEE International Conference on Image Processing (ICIP), October 2018. https://doi.org/10.1109/ICIP.2018.8451652.

[2] R. Caye Daudt, B. Le Saux, A. Boulch, and Y. Gousseau, *Urban change detection for multispectral earth observation using convolutional neural networks*, in IEEE International Geoscience and Remote Sensing Symposium (IGARSS), July 2018. http://dx.doi.org/10.1109/IGARSS.2018.8518015.

[3] T. Chen, Z. Lu, Y. Yang, Y. Zhang, B. Du, and A. Plaza, *A Siamese network based U-Net for change detection in high resolution remote sensing images*, IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 15 (2022), pp. 2357–2369. https://doi.org/10.1109/JSTARS.2022.3157648.

[4] R. Caye Daudt, B. Le Saux, A. Boulch, and Y. Gousseau, *Multitask learning for large-scale semantic change detection*, Computer Vision and Image Understanding, 187 (2019), p. 102783. https://doi.org/10.1016/j.cviu.2019.07.003.

[5] K. He, X. Zhang, S. Ren, and J. Sun, *Deep residual learning for image recognition*, 2015. https://doi.org/10.48550/arXiv.1512.03385.

[6] O. Ronneberger, P. Fischer, and T. Brox, *U-net: Convolutional networks for biomedical image segmentation*, 2015. https://doi.org/10.48550/arXiv.1505.04597.

[3] https://ieee-dataport.org/open-access/oscd-onera-satellite-change-detection