Change Detection with Masked Image Modeling

April 4, 2023

Arnaud Gardille, Sébastien Meyer
École Normale Supérieure Paris-Saclay, France

Abstract

The task of change detection on satellite images has been addressed both in the supervised setting, with convolutional neural networks and more recently Transfomers, as well as in the unsupervised setting, with methods such as Change Vector Analysis. However, models developed in both approaches still hardly take into account the complexity of objects within a given scene. Between two images, not only the changes of interest occur, but also changes in illumination, atmospheric conditions or ageing of buildings, to name but a few. In this study, we elaborate on OmniMAE, a masked autoencoder model which takes as input a sequence of images and we propose both an unsupervised and a supervised application of OmniMAE to change detection on satellite images. The unsupervised method relies on both the reconstruction and the prediction losses, while the supervised method consists in a simple convolutional head appended to the Transformer decoder. While our methods do not improve baseline results on SZTAKI nor OSCD datasets, they introduce efficient ways of reconstructing satellite images and finetuning models, as OmniMAE only requires a few patches from the original images in order to be trained. Our code is available on GitHub[1] and our experiments are reproducible by using the provided parameters.

Introduction and Contributions

Change detection on satellite images has been an important research area due to its numerous potential applications. The rise of deep learning method for conditional images generation like U-Net [1] makes it quite tempting to learn changes in a supervised manner. However such methods require quite massive labeled dataset, which are very expansive to create. Moreover, labels are quite task-specific as one might be only interested in a tiny subset of the numerous changes really happening on the images. On the other hand, vast amount of unlabeled satellite images are freely available, so that unsupervised methods become quite attractive. They also have the benefit to allow for the discovery of unanticipated events.

Nevertheless, change detection based on high resolution images remains a challenging task for several reasons. On the one hand, the objects present in a scene will vary from a dataset to another, and even inside a location. For instance, an image of Paris will include several buildings, cars, and shapes with many edges, while an image of fields will show flat surfaces and differences will mainly occur in terms of vegetation. On the other hand, many changes might occur between two images which are taken within an interval of months, while not being of interest to the researcher. In such case, the aspect of buildings might change due to illumination variations or appearance alteration such as ageing. Also, fields might change due to harvesting or weather conditions.

In this study, we investigate the use of a recently released masked autoencoder model for change detection. Related work is presented in **Section 1**, while we detail our method in **Section 2**. The explored method consists in predicting a last image based on some of its "patches" plus patches of a first co-registered image. The main idea behind our method is that the first image might contain information about the global structure of both images, while the patches from the last image brings some information about qualitative changes such as illumination changes which occur only in the last image. If the new image cannot be correctly reconstructed, we make the assumption that it is due to significant modifications between the two images and thus detect a change in an unsupervised fashion. In a more general framework, masked autoencoder can be used to extract semantic information about images. Based on this knowledge, we introduce OmniMAECNN, a model using the studied masked autoencoder as a base model, with a convolutional head predicting change maps in a supervised fashion. Our experiments and results regarding both unsupervised and supervised methods are discussed in **Section 3**.

[1] https://github.com/sebastienmeyer2/masked-change-detection

1. Related Work

Change detection on satellite images can be addressed both in the supervised and unsupervised settings. In this section, we present different methods which consist in predicting a complete change map from a given pair of multi-temporal satellite images. These methods will, in part, serve as baselines for our qualitative as well as quantitative experiments.

1.1. Supervised Change Detection

Given the advances in computer vision, the use of machine learning for change detection on satellite images is not a recent idea. For instance, the first use of convolutional neural networks for change detection on satellite images dates back to as early as 2015. In [2], the authors define a convolutional network working on a set of two small input patches of shape 64×64 representing an area at two different points in time. Then, a final decision network outputs a similarity prediction which allows to perform change detection. However, this type of models is limited to only detecting whether if a change occured, but not where. As we are interested in predicting where changes occur, we focus our study on models which output predictions at the pixel level.

CNNs. The first models that were proposed for performing change detection on satellite images were inspired from [2]. In [3], the authors present two CNN-based models, namely EF and Siam. The main idea behind Siam is to build a siamese network, that is, a network which extract semantic information from both images using a CNN architecture with shared weights. On the contrary, EF simply takes as input a concatenated version of both images. Both models receive small patches as inputs, of shape $2 \times 15 \times 15 \times C$ for Siam and of shape $15 \times 15 \times 2C$ for EF and they output a binary class stating whether if a change occurred for the given patch. Complete change maps are produced by classifying all patches within the image and smoothing the results using a 2D gaussian vote centered at the central pixels of classified patches.

FCNNs. More recently, fully convolutional neural networks (FCNNs) have been proposed in order to perform pixel-level classification [4]. As shown **Figure 1**, the authors proposed three new fully connected architectures, based on both the architectures from [3] and the principles of U-Net, that is, using skip connections between the downsampling and upsampling operations. The upsampling operation outputs a binary image with the same shape as the original image, allowing for pixel-level classification. Firstly, FC-Siam-conc and FC-Siam-diff are two variants which build upon Siam. While FC-Siam-conc concatenates feature maps from both images during upsampling, FC-Siam-diff only concatenates the difference between them. The last proposed architecture, namely FC-EF, simply extends EC with similar ideas. Finally, an extended version of FC-EF is proposed under FresUNet. While not detailed in the original paper from the authors, the architecture of this model can be found in their github repository[2]. Compared to FC-EF, FresUNet implements slightly more sophisticated skip connections after each downsampling and upsampling operations.

Transformers. Transformer [5] has shown to be a very powerful architecture in terms of transfer learning for various sub-tasks. Thus, its application to change detection on satellite images has been recently tested out. A Transformer-based architecture has been proposed in [6], where the tokens from both images are concatenated before encoding and decoding, thus allowing the Transformer to use information from both images during decoding. The illustration of the proposed BIT-based model is shown **Figure 2**. As this architecture is much more complex than FCNNs, we did not use this model as a baseline. However, as we will highlight in **Section 2**, we used the proposed BIT architecture to design our own supervised change detection method.

1.2. Change Detection Datasets

FCNNs and Transformer-based models allow to predict a full change map given two images, therefore allowing for fine-grained change detection. They can also be dedicated to the detection of specific changes (vegetation, buildings, etc.). To that end, supervised change detection require annotated change maps for training. The lack of annotated change detection data has been a real issue for the change detection community, especially for bigger models such as Transformers, which require even more data in order to output meaningful results. Still, more and more change detection datasets are publicly available. Below, we present some of these.

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

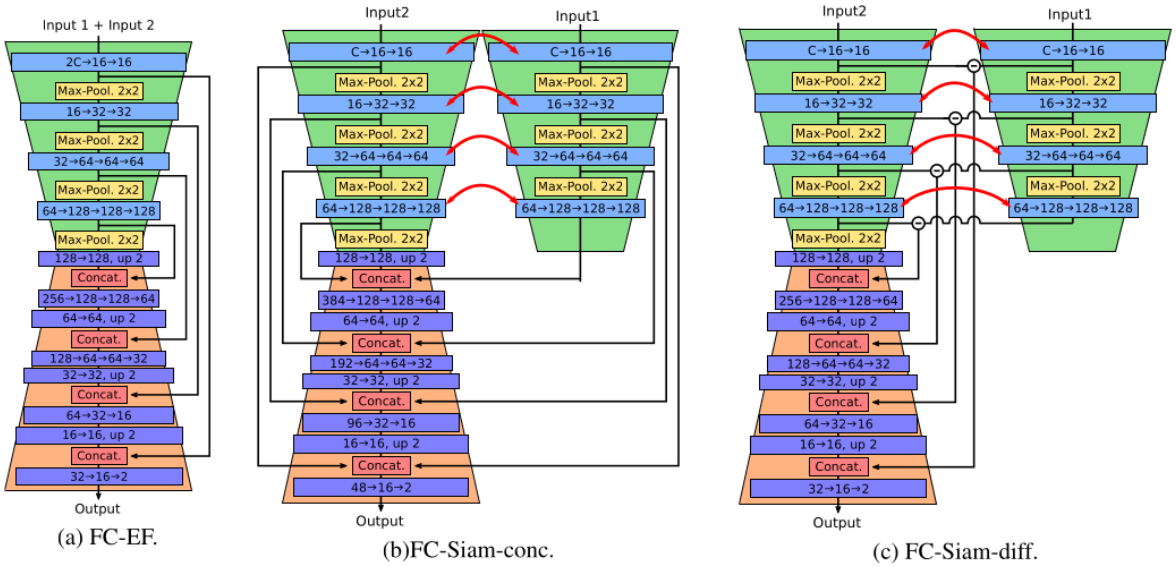


Figure 1. Illustration of the three architectures for change detection as proposed in [4]. Red arrows illustrate shared weights.

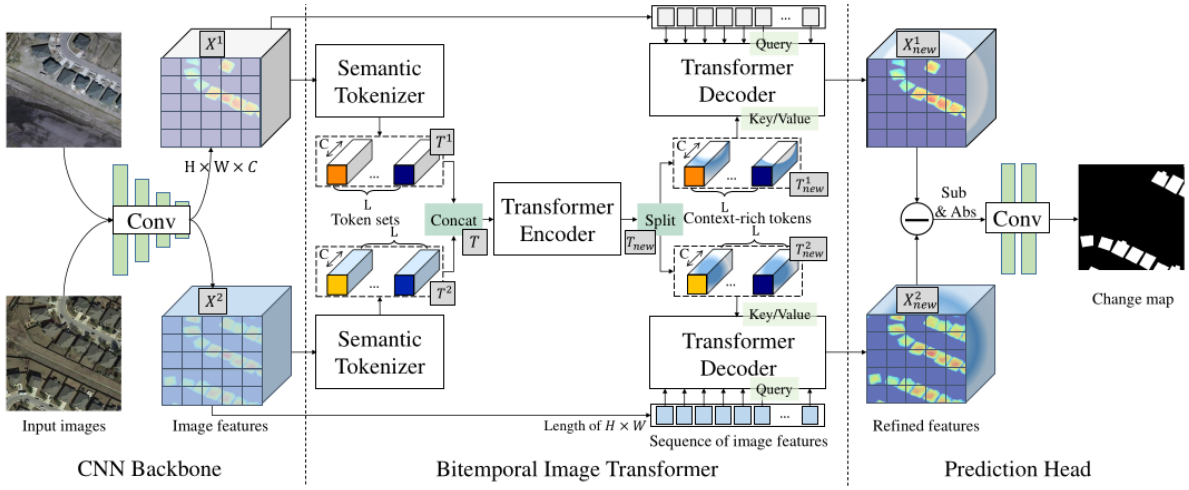


Figure 2. Illustration of the BIT-based model as proposed in [6]. Input images are transformed into feature maps, which are then converted to tokens fed to a Transformer encoder. Both images are reconstructed based on these tokens, while a final convolutional head outputs the change map.

SZTAKI. The SZTAKI dataset [7, 8] is a public change detection dataset introduced in 2008 and comprising 13 pairs of 952×640 medium-resolution ($1.5m$) RGB aerial images. The images are divided into three locations: Archieve (1 pair), Szada (7 pairs) and Tiszadob (5 pairs). This dataset is widely used for the evaluation of change detection architectures, as in [4]. However, it contains only RGB images and not multispectral ones. Consequently, we used this dataset mainly for the evaluation of unsupervised change detection methods.

OSCD. The Onera Satellite Change Detection dataset [3] is a public change detection dataset introduced in 2018, comprising 24 pairs of multispectral images from the Sentinel-2 satellites and spanning between 2015 and 2018. For each location, registered pairs of 13-band multispectral satellite images vary in spatial resolution between $10m$, $20m$ and $60m$. The annotated changes focus on urban changes, such as new buildings or new roads. We used this dataset for the evaluation of both unsupervised and supervised change detection methods.

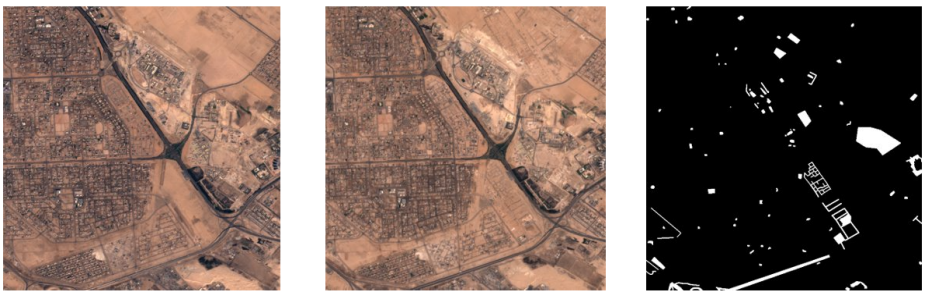


Figure 3. Abu Dhabi images from the OSCD dataset. Some buildings are under construction both at the center of the image (correctly annotated) and at the upper right of the image (not annotated).

LEVIR-CD. The LEVIR-CD dataset [9] is a public change detection dataset introduced in 2020 and comprising 637 pairs of 1024×1024 high-resolution ($50cm$) remote sensing images. The annotations were made by the Learning, Vision and Remote Sensing Laboratory from Beijing, China. This dataset is one of the datasets used to train and evaluate the BIT model presented in [6].

Let us make several remarks about datasets here. Firstly, as we can observe, there is no clear dataset which can be used for a general change detection purpose. Indeed, researchers or companies are interested in detecting changes from a specific nature (buildings, vegetation, etc.) and therefore require very precise annotation. As for general machine learning tasks, companies or laboratories who have built good quality datasets might be tempted to keep them for themselves, this is the reason why we do not have publicly available large datasets. In addition, authors who evaluate supervised change detection architectures are often the ones who produced the dataset used for evaluation, thus are more likely to know how to tweak their models to perform better on the chosen dataset. Finally, annotations are most of the time human-based. So, the choice of selecting what to annotate as a change boils down to deciding what defines a change from the annotator perspective. As shown **Figure 3**, some changes might not be annotated in the ground truth labels, leading to questionable datasets.

1.3. Unsupervised Change Detection

On the contrary, unsupervised change detection methods do not make any assumption about having precise labels for training. These methods rely solely on the input images in order to output change maps. However, we can make at least two remarks concerning those methods. On the one hand, these methods are making no assumption about the input images. So, their parameters need to be chosen wisely in order to provide meaningful predictions. Again, some supervised learning can be implemented in order to choose the best hyperparameters, coming back to the limitations of supervised change detection. On the other hand, these methods have been proven to work mainly on preprocessed, clean images. Therefore, some prior work on the input images is required before doing prediction.

CVA. Change Vector Analysis (CVA) is a simple analysis technique based on the L2 distance between input images. If we denote by I_1 the first image and by I_2 the second image, CVA starts by normalizing the images, denoted $\widetilde{I_1}$ and $\widetilde{I_2}$. The change map is computed as follows:

$$L = ||\widetilde{I_1} - \widetilde{I_2}||_2.$$

Then, the normalized change map $\widetilde{L}$ is computed and changes are detected according to a threshold. This threshold can either be selected manually (e.g. through empirical observations) or through automated selection for minimizing some criterion such as false positive rate. In our study, we decided to perform thresholding in two steps. First, we apply a median filtering to filter out isolated pixels. Second, we apply a percentile thresholding which allows to keep pixels only above the chosen percentile value.

DeepCVA. DeepCVA [10] is a CNN-based Change Vector Analysis technique which was introduced in 2019. In DeepCVA, Change Vector Analysis is computed between features extracted by a Convolutional Neural Network. By using deep features, DeepCVA gets rid of local or global changes that are not of interest between images (angle of acquisition, atmospheric conditions, etc.). The proposed model is based on a model which was pretrained for the task of semantic labeling on remote sensing aerial images. As discussed in the Appendix, DeepCVA works better on high-resolution images. When using

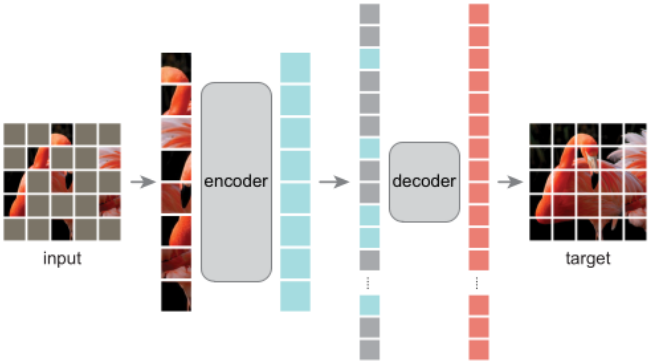


Figure 4. MAE architecture as proposed in [12]. Only visible patches are sent through the encoder.

224×224 crops, we found out that the model was unstable, so we did not use it for our comparison.

Clustering. Clustering is a simple method which consists in doing a PCA on the absolute difference between the two input images and to apply k-means clustering on some of its principal components. Finally, the resulting change map has to be resized back to the image shape and a fixed thresholding is applied to assign labels. On top of that, we added closing and opening operations which we discuss in more detail in the Appendix.

1.4. Masked Image Modeling

Since the introduction of the Transformer architecture in [5], there have been many other developments dedicated to the application of Transformers to other tasks than NLP. More specifically, Dosovitskiy et al. introduced the Vision Transformer in [11]. Vision Transformers can be used to extract semantic information from images, such as convolutional neural networks. However, they work in a different manner. For instance, Vision Transformers take as inputs tokens, which we will call "patches" from now on, of shape 16×16. The output of the Vision Transformer also corresponds to a combination of these patches in order to reconstruct a representation of the input image.

In addition, it is possible to use Vision Transformers to perform auto-encoding. Auto-encoding consists in sending the input image to a latent space before reconstructing the original image. In the case of Vision Transformers, a simple linear layer is appended to the end of a first Vision Transformer architecture, the *encoder*, while the *decoder* consists in a second Vision Transformer architecture. In general, the decoder architecture is shallower than the encoder architecture.

MAE. Masked Autoencoders [12] were introduced in 2021. The concept of Masked Autoencoders builds upon the Vision Transformer architecture from a self-supervised setting. As shown **Figure 4**, a random proportion of patches are "masked", i.e. discarded, before encoding. Then, the model learns to reconstruct the full image by only using the latent representation and mask patches. This technique yields improvements in both the training time (depending on the proportion of masked patches, of 75% in the original paper) and the accuracy. The encoder can then be used for downstream tasks such as image classification.

OmniMAE. Since the introduction of MAE, there has been several trials to extend it to videos, from which we can cite VideoMAE [13]. However, the most flexible MAE model for videos, namely OmniMAE [14], was released by Meta AI in 2022. The main difference between MAE and OmniMAE is that, in order to support sequences as inputs, OmniMAE divides videos using patches of shape $2 \times 16 \times 16$, where 2 represents the time dimension. Therefore, the positional encoding is also adapted in order to fit a fixed input length. The base length of input videos is 16 frames, thus yielding 8 patches in terms of temporality. Simple images are duplicated in order to be viewed as 2 frames and to be patchable. OmniMAE allows to perform self-supervised learning with a drastically reduced proportion of patches. Indeed, OmniMAE supports masking proportions of approximately 90% for images and of 95% for videos during training.

2. Method

Since OmniMAE can take videos as inputs, the main objective of our study was to try to feed the model with some sequence of satellite images in order to perform change detection. Indeed, by passing patches from both images to the model, it should reconstruct the images by taking into account:

1. Global information (atmospheric conditions, etc.) thanks to the visible patches of the last sequence (of two frames).
2. Local information (buildings, roads, etc.) thanks to the patches of the first sequence (of two frames) when the patches of the last sequence (of two frames) are masked.

As the available code on GitHub[3] was not working for reconstructing videos, we had to make some changes to the Vision Transformer architecture. Then, we designed an easy-to-use strategy for passing two satellite images through the model. Finally, this allowed us to design both supervised and unsupervised change detection methods based on OmniMAE.

2.1. General Changes

On the one hand, the code as provided by the authors would not work with videos. We found out that this was due to the positional encoding method inside the Vision Transformer architecture. Thus, we modified the method for supporting videos. Moreover, we adapted the positional encoding method to make OmniMAE support any number of frames. To that end, we implemented the following changes:

1. The original positional encoding values are given for a video of 16 frames.
2. For videos longer than 16 frames, repeat the positional encoding as many times as needed.
3. For odd number of frames (for example, simple images), simply duplicate the last frame to fit the patch shape.

Thanks to these simple yet powerful modifications, we were then able to use OmniMAE with different sequences of satellite images. More specifically, we created fake "videos" of 4 frames by putting images one after the other. If we denote by $\mathbf{1}$ the first image and by $\mathbf{2}$ the last image, what we define by video ordering denotes the sequence of images fed to the model. The two main video orderings that we tested out were $\mathbf{1122}$ and $\mathbf{1221}$. Interestingly, we know that OmniMAE was trained on videos, so it expects frames within a patch to be very similar. Consequently, the model outputs frames that are very similar when taken two-by-two. By setting the video ordering to $\mathbf{1221}$, we get a first sequence (of two frames) that are very similar and a second sequence (of two frames) that are also very similar, while still showing differences when we take the difference of both sequences.

On the other hand, we also had to design the masking strategy. OmniMAE supports any type of masking, as long as the proportion of masked patches is the same within a given batch of images (which will not impact our usage except during training). We designed three different masking strategy as follows:

- **none**: all patches are passed through the model,
- **complementary**: half of the patches are masked out in the first sequence (of two frames), while the complementary patches are masked out in the second sequence (of two frames).
- **random**: a given proportion of patches are masked out in each of the sequences (of two frames).

Let us denote by $\mathcal{M}$ the model and for a given sequence $\mathbf{ij}$, we denote by $\mathcal{M}(\mathbf{ij})$ the effect of the model on sequence $\mathbf{ij}$. The output sequence for the video ordering $\mathbf{1221}$ can therefore be written as $\mathcal{M}(\mathbf{12})\mathcal{M}(\mathbf{21})$.

2.2. Supervised Change Detection

The method that we designed for supervised change detection using OmniMAE is inspired from [6]. Indeed, we can notice that, as the BIT-based Transformer architecture shown **Figure 2**, OmniMAE outputs a sequence of images (in our case, 4 images). Based on this output, we compute the absolute difference between the first reconstructed image of the first sequence $\mathcal{M}(\mathbf{12})_1$ and the first reconstructed image of the second sequence $\mathcal{M}(\mathbf{21})_1$, i.e. $|\mathcal{M}(\mathbf{12})_1 - \mathcal{M}(\mathbf{21})_1|$ the difference between the first and the third reconstructed images. Finally, the resulting difference image is fed to a shallow convolutional head which outputs pixel-level logits indicating the presence or not of a change. For instance, this convolutional head needs to be trained in a supervised setting in order to be able to output meaningful change maps. We call this model OmniMAECNN.

[3] https://github.com/facebookresearch/omnivore

2.3. Unsupervised Change Detection

In the unsupervised setting, we define *reconstruction loss* as the sum of all L2 losses between each input image and its reconstructed counterpart. We also define the *prediction loss* as the contiguous loss between reconstructed images, that is, the sum of all L2 losses between reconstructed image at index i and reconstructed image at index $i+1$ (combination of three losses). Finally, we define *both losses* as the sum of both the *reconstruction loss* and the *prediction loss*. In addition, we define the *baseline loss* as the simple L2 loss between image $\mathbf{1}$ and image $\mathbf{2}$. From these loss images, the procedure to output a change map is as follows:

1. Sum over channels.
2. Rescale the tensor between 0 and 1.
3. Perform a median filtering.
4. Apply a percentile thresholding to set the top percentile values to 1, the rest to 0.

3. Experiments

In this section, we present the results obtained for both the supervised change detection methods and the unsupervised change detection methods. Firstly, we present the implementation details as well as the considered datasets. Then, we show some visualization of change maps together with quantitative results.

3.1. Supervised Change Detection

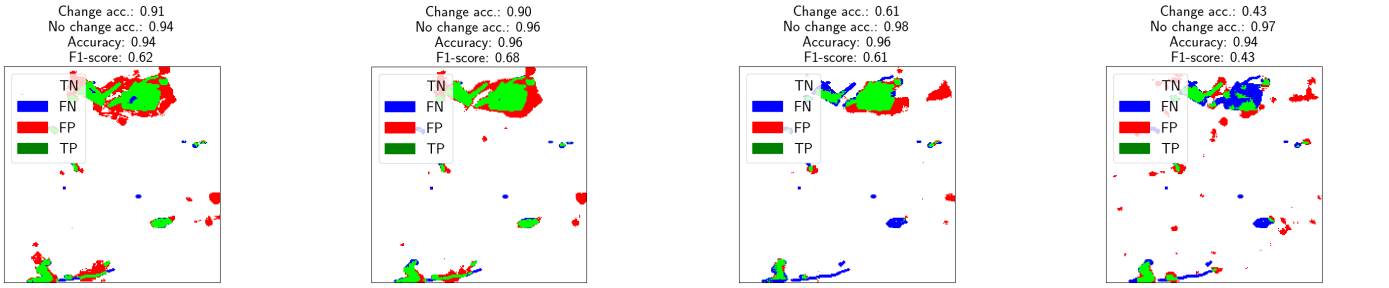
Implementation details. For FresUNet, we use the default implementation from [4]. For OmniMAECNN, we finetune only the weights from the first and last layers, together with the convolutional head. Indeed, the convolutional head only contains a few hundred parameters, which is not enough to extract meaningful information from the reconstructed images. Allowing the first and last layers of OmniMAE to be finetuned increases the number of trainable parameters to a few millions. In addition, we modified the architecture of OmniMAE to support different number of channels. For the first layer, the weights of new channels are initialized to the weights of the Red channel and finetuned. For the last layer, we append again the first third of weights as many times as needed. Contrary to the unsupervised setting, there is no other post-processing than predicting the class with the highest logit value.

Data. For our study, we used the OSCD dataset with a similar preprocessing as in [4]. For each city, we concatenate the desired channels: Red, Green and Blue for **RGB** or Red, Green, Blue, NIR, Vegetation Red Edge (3 channels), Narrow Nir, SWIR (2 channels) for **Res20**. The full city images are then normalized to zero mean and unit variance. In order to create enough training samples, smaller images of shape 224×224 are created from the city images. We take all such small images with a stride of 112. This allows to create a training set containing 321 images and a test set containing 124 images. Compared to [4], where smaller images were of size 96×96, we produce much less training data.

Qualitative analysis. For the qualitative analysis of our supervised change detection method OmniMAECNN, we rely on a 224×224 crop from the Montpellier city image from OSCD dataset (see **Figure 5**). For FresUNet, increasing the number of channels improves the results in terms of both accuracy and F-score. With more bands, the model is able to detect more changes and to avoid more false positives. Overall, we observe a better detection of changes in the image. Regarding OmniMAECNN, using only RGB bands already produces good results, with a F-score being very close to the one of FresUNet (RGB). However, we notice that even though the F-score is large, the recall of the model is quite low. Surprisingly, our application of OmniMAECNN trained on Res20 bands on this crop yields poorer results, with a quite low F-score of 0.43 against 0.61 in the RGB case.

Quantitative analysis. The overall results of supervised change detection baselines as well as OmniMAECNN are presented in **Table 1**. If we compare both models only for the RGB bands, we can observe that the accuracy and precision of OmniMAECNN and FresUNet are very close, while the recall of OmniMAECNN is much smaller. On the contrary, when we use OmniMAECNN with Res20 bands, the recall starts increasing very rapidly. Nevertheless, the other scores such as accuracy and precision do not improve and even tend to decrease. Overall, the best model is FresUNet with the Res20 bands, while OmniMAECNN with Res20 bands barely beats FresUNet with RGB bands. However, it is still interesting to note that the video ordering **1221** reaches a F-score of more than 50%.

Figure 5. Crop of size 224×224 from the Montpellier city in OSCD dataset used to evaluate supervised change detection methods.



(a) **FresUNet** (RGB) (b) **FresUNet** (Res20) (c) **OmniMAECNN** (RGB) (d) **OmniMAECNN** (Res20)

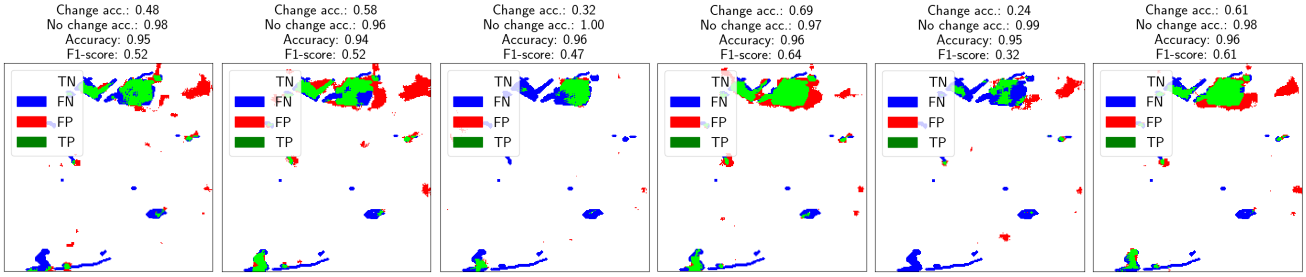
Figure 6. Change maps for supervised change detection baselines and $\mathbf{OmniMAECNN - 1221 + random\ (80\%)}$.

Method	Bands	# Params	Global acc.	Precision	Recall	F-score
FresUNet	RGB	1,103,874	93.66	42.20	50.30	45.89
FresUNet	Res20	1,104,994	**95.05**	**53.61**	**54.89**	**54.25**
$\mathbf{OmniMAECNN - 1221 + random\ (20\%)}$	RGB	1,769,855	94.21	44.30	32.20	37.29
$\mathbf{OmniMAECNN - 1221 + random\ (20\%)}$	Res20	5,907,150	93.67	42.46	51.87	46.70

Table 1. Supervised change detection results on OSCD dataset.

Influence of parameters As shown in both **Figure 7** and **Table 2**, we perform a study of the influence of the different parameters of OmniMAECNN. Firstly, we can notice that in all cases, using the video ordering **1221** yields better results than using the video ordering **1122** in terms of F-score. Recall that using the video ordering **1221** with OmniMAE will bring reconstructed images that are quite similar, while still carrying information about changes between the two images. This method also enforces some time invariance inside our OmniMAECNN model. However, using the video ordering **1122** with the masking method **none** gives better results in terms of recall, which is where FresUNet also performs better than our chosen parameters. Finding a good trade-off between precision and recall for OmniMAECNN requires further research.

Furthermore, we can observe that adding randomness improves the results in the case of the video ordering **1221**, while it is not the case for the video ordering **1122**. We are not able to provide an explanation for this phenomenon. Nevertheless, as we know that the video ordering **1221** is already better in the case of the masking method **none**, we can comment on the improvements observed for other masking strategies. Indeed, the less patches, the better the results. This can be linked to how OmniMAE was trained originally, with only 10% of the image patches and only 5% of the video patches. Consequently, we chose the video ordering **1221** and the masking method **random** with 20% of visible patches.



(a) **1122 − none** (b) **1221 − none** (c) **1122 − compl.** (d) **1221 − compl.** (e) **1122 − rdm** (f) **1221 − rdm**

Figure 7. Comparison of OmniMAECNN with different parameters (RGB bands) on a single crop.

Figure 8. Crop of size 224×224 from the Szada city in SZTAKI dataset used to evaluate unsupervised change detection methods.

Video ordering	Masking method	Global acc.	Precision	Recall	F-score
1122	**none**	87.85	21.12	**46.59**	29.07
1221	**none**	92.60	31.76	33.54	32.62
1122	**complementary**	93.75	15.80	3.91	6.27
1221	**complementary**	93.61	38.64	33.42	35.84
1122	**random (20%)**	93.61	33.44	19.84	24.90
1221	**random (20%)**	**94.21**	**44.30**	32.20	**37.29**

Table 2. Comparison of OmniMAECNN with different parameters (RGB bands).

3.2. Unsupervised Change Detection

Implementation details. The implementation of both CVA and DeepCVA were retrieved from TP9[4], while the implementation of the Clustering method has been adapted from GitHub[5]. For CVA, *baseline loss*, *reconstruction loss*, *prediction loss* and *both losses*, the window size of the median filtering was set to 4 and the percentile thresholding to 95%. For the Clustering method, we use 3 principal components and a fixed threshold of 100 and we assign pixels above to 1 and pixels below to 0.

Data. We evaluated the unsupervised change detection methods on both OSCD and SZTAKI datasets. We perform a simple equalization for a percentile value of 2 for both images. Then, we perform histogram matching of the last image with respect to the first one. Moreover, we randomly crop out small images of size 224×224.

Qualitative analysis. In order to perform a first analysis of the different unsupervised change detection methods, it is useful to produce visualization examples. In **Figure 8**, we show a cropped sample from the SZTAKI dataset together with its ground truth change map. Note that only changes related to buildings or private roads are labeled, even though some other roads have changed between both images. In **Figure 9**, we show the images output by OmniMAE when the video ordering is **1122** and only 20% of the patches from all images are sent through the model. As we can observe, the reconstruction is still quite close to the true images.

In **Figure 10**, we evaluate the different unsupervised change detection baselines against OmniMAE with video ordering **1122** and masking method **none**. Firstly, we can observe that the baseline loss is already performing relatively well. CVA

[4] https://mvaisat.wp.imt.fr/

[5] https://github.com/ChaymaBouzaidii/Change-detection-in-multitemporal-satellite-images

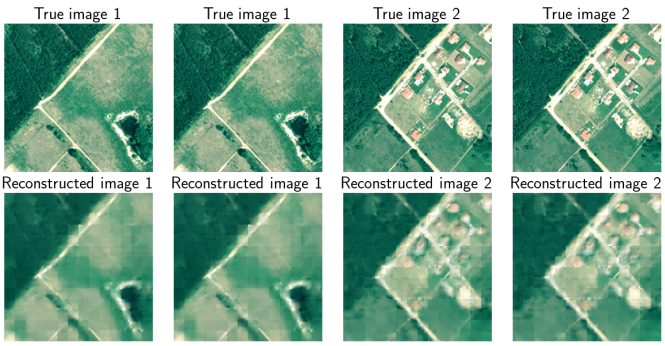
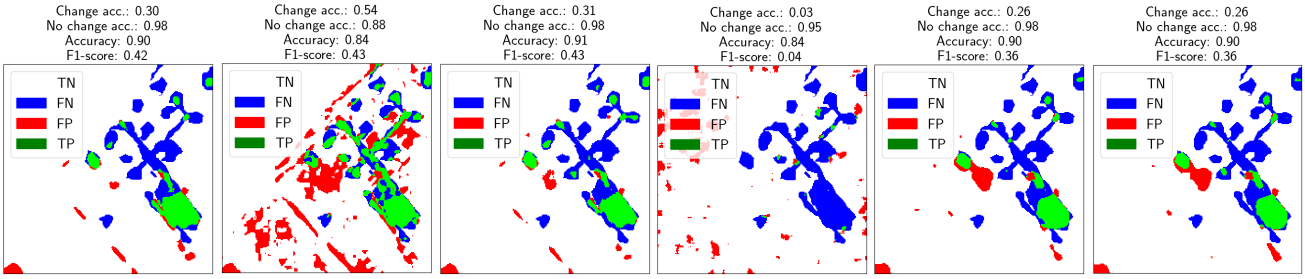


Figure 9. Reconstruction of the original images with $\mathbf{OmniMAE - 1122 + random}$ (20%).



(a) **CVA** (b) **Clustering** (c) **Baseline** (d) **Reconstr.** (e) **Prediction** (f) **Both**

Figure 10. Change maps for unsupervised change detection baselines and $\mathbf{OmniMAE - 1122 + none}$.

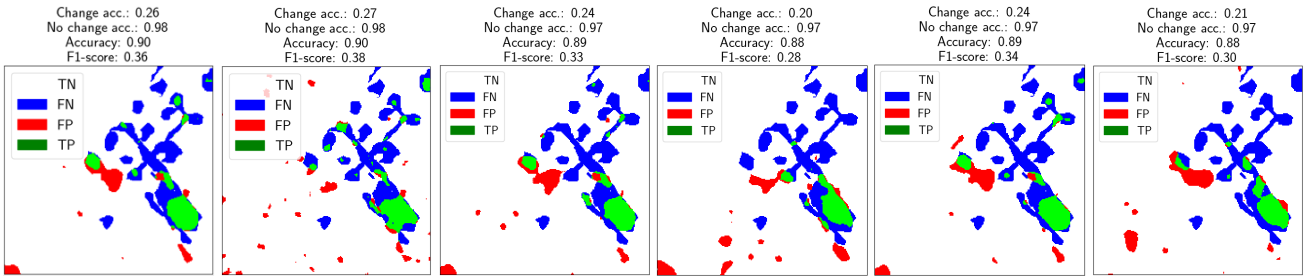
yields very similar results, while, interestingly, the Clustering method yields a similar F-score but a much larger accuracy for changes (at the cost of a smaller global accuracy). Against these baselines, OmniMAE does not perform better. We can notice that combining both losses is not very meaningful. For instance, the large construction area at the bottom right of the image is detected in the *prediction loss*, while only some remnants of the central buildings are detected in the *reconstruction loss*. Overall, the change accuracy and F-score of OmniMAE are not better than the accuracy and F-score of unsupervised change detection baselines.

Quantitative analysis. For the quantitative analysis, we performed the same Data pre-processing as for supervised change detection for OmniMAE, that is, we use the images from OSCD dataset translated to zero mean and unit variance. Only CVA and Clustering methods use simple equalization and histogram matching as preprocessing. Although results might be slightly different from what we would expect from the qualitative analysis, we give the overall scores of unsupervised methods in **Table 3**. It appears that both CVA and *baseline loss* perform relatively well in terms of F-score. Again, we observe that OmniMAE cannot improve the results obtained by simply using the original images.

Method	Global acc.	Precision	Recall	F-score
CVA	93.58	39.20	36.66	37.88
Clustering	92.52	30.82	32.03	31.41
Baseline loss	**93.69**	**40.32**	**37.71**	**38.98**
OmniMAE − 1122 + none	93.28	36.24	33.90	35.03

Table 3. Unsupervised change detection results on OSCD dataset.

Influence of parameters. As shown **Figure 11**, we perform a study of the different parameters of OmniMAE. Indeed, the results presented **Figure 10** stem from using the video ordering **1122** and the masking method **none**. In order to qualitatively evaluate the other ordering and masking strategies, we can compare them to the one we selected. In general, feeding the model with less patches than in the original images yields worse results. As a conclusion, the model is still able to recon-



(a) **1122 − none** (b) **1221 − none** (c) **1122 − compl.** (d) **1221 − compl.** (e) **1122 − rdm** (f) **1221 − rdm**

Figure 11. Comparison of unsupervised change detection methods and different parameters for OmniMAE on a single crop.

struct the image, but with decreasing quality. Interestingly, we can notice that using the sequence **1221** sometimes leads to better results. The internal processing of these images by OmniMAE might enforce some time invariance, while the sequence **1122** does not. However, the video ordering **1122** yields in general better results than the video ordering **1221**.

Conclusion

All in all, we have illustrated our modifications of OmniMAE on simple examples. However, the overall results of our unsupervised and supervised methods remain lower than the results of the baselines, *baseline loss* in the unsupervised setting and *FresUNet* in the supervised setting. The main observation is that, although being quite complex and sophisticated, the processing of satellite images by OmniMAE does not take more information into account than the simple baselines such as simple equalization or histogram matching. Also, OmniMAE was trained using a non-perceptual loss, so it cannot be used off-the-shelf for advanced change detection. Finally, we can notice that OmniMAE and our methods currently support input images of size 224×224, while most if not all of the baselines we studied support inputs of any size.

Further research could be conducted on semi-supervised learning and performing change detection in the latent space. More specifically, we have shown that we can successfully finetune a masked autoencoder comprising Vision Transformers to perform change detection on satellite images. Therefore, it is clear that the latent space of such model contains meaningful information about the overall scene and the difference between both images. This might be expected to give more information to a more sophisticated model than our simple convolutional head.

References

[1] Olaf Ronneberger, Philipp Fischer and Thomas Brox. *U-Net: Convolutional Networks for Biomedical Image Segmentation.* In International Conference on Medical image computing and computer-assisted intervention. Springer, 2015, pp. 234–241.

[2] Sergey Zagoruyko and Nikos Komodakis. *Learning to Compare Image Patches via Convolutional Neural Networks.* In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 4353–4361.

[3] Rodrigo Caye Daudt, Bertrand Le Saux, Alexandre Boulch and Yann Gousseau. *Urban Change Detection for Multispectral Earth Observation using Convolutional Neural Networks.* In International Geoscience and Remote Sensing Symposium (IGARSS). IEEE, 2018.

[4] Rodrigo Caye Daudt, Bertrand Le Saux and Alexandre Boulch. *Fully Convolutional Siamese Networks for Change Detection.* In IEEE International Geoscience and Remote Sensing Symposium (IGARSS) 2018, pp. 2115-2118.

[5] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser and Illia Polosukhin. *Attention Is All You Need.* In Advances in Neural Information Processing Systems. December 2017.

[6] Hao Chen, Zipeng Qi and Zhenwei Shi. *Remote Sensing Image Change Detection with Transformers.* In IEEE Transactions on Geoscience and Remote Sensing, pp. 1-14. July 2021.

[7] Csaba Benedek and Tamás Szirányi. *Change Detection in Optical Aerial Images by a Multilayer Conditional Mixed Markov Model.* In IEEE Transactions on Geoscience and Remote Sensing, vol. 47, no. 10, pp. 3416-3430. 2009.

[8] Csaba Benedek and Tamás Szirányi. *A Mixed Markov Model for Change Detection in Aerial Photos with Large Time Differences.* In International Conference on Pattern Recognition (ICPR), Tampa, Florida, USA. December 2008.

[9] Hao Chen and Zhenwei Shi. *A Spatial-Temporal Attention-Based Method and a New Dataset for Remote Sensing Image Change Detection.* In Remote. Sens., vol. 12, no. 10, p. 1662, 2020.

[10] Sudipan Saha, Francesca Bovolo and Lorenzo Bruzzone. *Unsupervised Deep Change Vector Analysis for Multiple-Change Detection in VHR Images.* In IEEE Transactions on Geoscience and Remote Sensing, Vol. 57, No. 6, pp. 3677-3693. 2019.

[11] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit and Neil Houlsby. *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale.* In International Conference on Learning Representations (ICLR), 2021.

[12] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár and Ross Girshick. *Masked Autoencoders Are Scalable Vision Learners.* In Conference on Computer Vision and Pattern Recognition (CVPR), 2022.

[13] Zhan Tong, Yibing Song, Jue Wang and Limin Wan. *VideoMAE: Masked Autoencoders are Data-Efficient Learners for Self-Supervised Video Pre-Training.* October 2022.

[14] Rohit Girdhar, Alaaeldin El-Nouby, Mannat Singh, Kalyan Vasudev Alwala, Armand Joulin and Ishan Misra. *OmniMAE: Single Model Masked Pretraining on Images and Videos.* 2022.

Appendix

DeepCVA

In this subsection, we explain more in detail why we were not able to use DeepCVA for our study. More precisely, let us consider the Szada city image from SZTAKI dataset as well as the corresponding 224×224 crop shown **Figure 8**. DeepCVA is an unsupervised change detection method which yields state-of-the-art results. As shown **Figure 12**, DeepCVA is able to reach a F-score of at least 50 for the whole image of Szada.

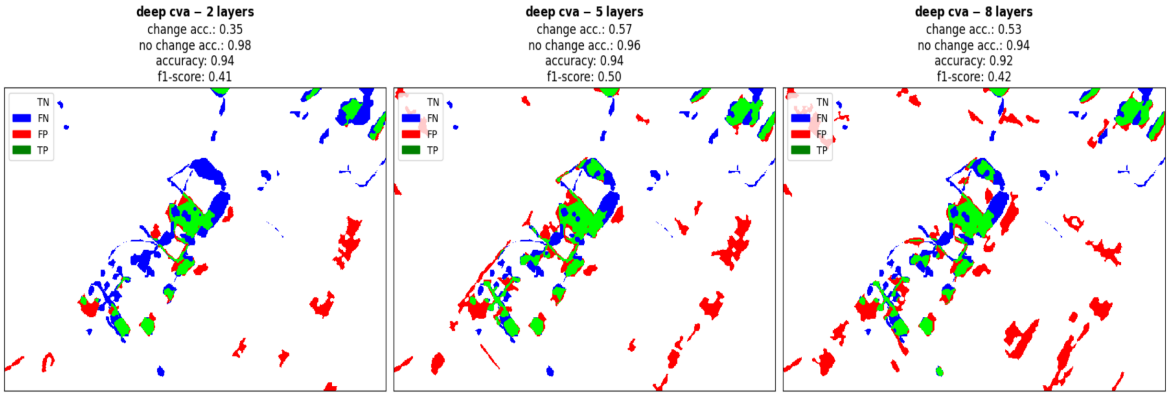


Figure 12. DeepCVA predictions on Szada images from SZTAKI dataset.

Nevertheless, this method is not adapted to smaller images. As shown **Figure 13**, for a crop of size 224×224, DeepCVA collapses regardless of the chosen threshold. In the original paper from Saha et al. [10], the authors indeed used DeepCVA on images of size between 800×800 and $1,400 \times 1,400$. Consequently, DeepCVA might require some adaptation in order to be used on images of smaller size.

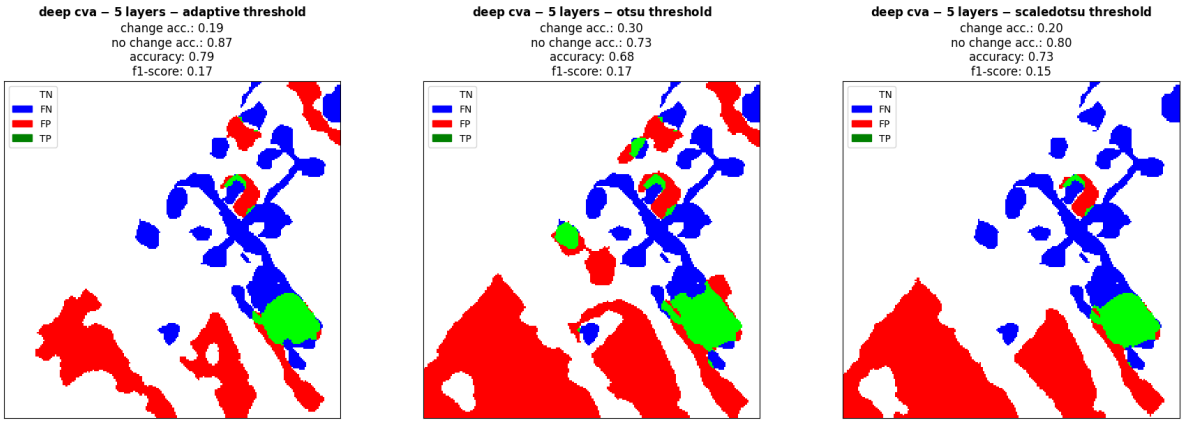


Figure 13. DeepCVA predictions on a 224×224 crop of Szada images from SZTAKI dataset.

Clustering

In this subsection, we explain more in detail our choice of hyperparameters for the Clustering method. Recall that for our experiments, we selected a number of principal components of 3 and a threshold of 100 for the final assignment of labels. In **Figure 14**, we show the results when using a number of principal components of 4. We can observe that the selected number of principal components has a huge impact on the change detection made by the Clustering method.

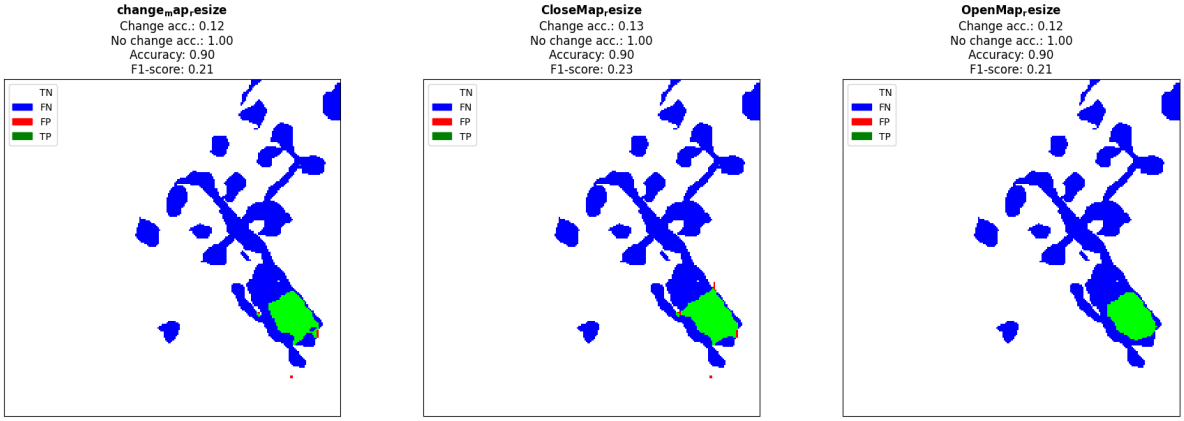


Figure 14. Clustering predictions on a 224 × 224 crop of Szada images from SZTAKI dataset with 4 principal components.

Finally, we show in both **Figure 14** and **Figure 15** the impact of closing and opening when applied to the predicted change map. In general, closing the change map largely increases the change accuracy while diminishing slightly the overall accuracy. In the end, the F-score when closing the change map is very similar to the original one. Lastly, opening the change map again does not largely improve the results. So, for all our experiments, we decided to use the original change map as predicted by the Clustering method.

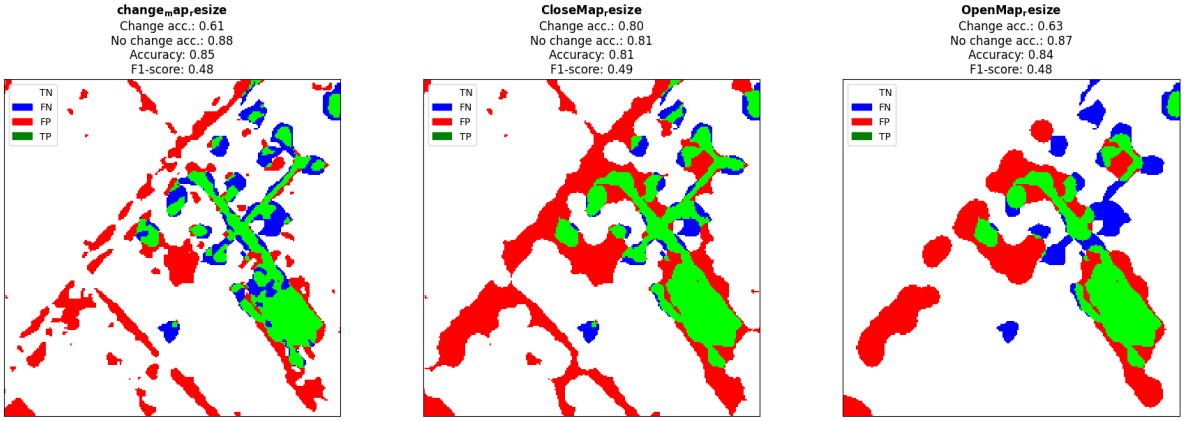


Figure 15. Clustering predictions on a 224 × 224 crop of Szada images from SZTAKI dataset with 3 principal components.