Fusion of multiple classifiers using self supervised learning for satellite image change detection

Alexandros Oikonomidis[1 [0000−0003−4803−6419]], Maria Pegia[1 [0000−0003−2643−0028]], Anastasia Moumtzidou[1 [0000−0001−7615−8400]], Ilias Gialampoukidis[1 [0000−0002−5234−9795]], Stefanos Vrochidis[1 [0000−0002−2505−9178]], and Ioannis Kompatsiaris[1 [0000−0001−6447−9020]]

Information Technologies Institute / Centre for Research & Technology Hellas, Thessaloniki, Greece
{aleoikon, mpegia, moumtzid, heliasgj, stefanos, ikom}@iti.gr

Abstract. Deep learning methods are widely used in the domain of change detection in remote sensing images. While datasets of that kind are abundant, annotated images, specific for the task at hand, are still scarce. Neural networks trained with Self supervised learning aim to harness large volumes of unlabeled satellite high resolution images to help in finding better solutions for the change detection problem. In this paper we experiment with this approach by presenting 4 different change detection methodologies. We propose a fusion method that under specific parameters can provide better results. We evaluate our results using two openly available datasets with Sentinel-2 satellite images, S2MTCP and OSCD, and we investigate the impact of using 2 different Sentinel 2 band combinations on our final predictions. Finally we conclude by summarizing the benefits of this approach as well as we propose future areas of interest that could be of value in enhancing the change detection task's outcomes.

Keywords: Change Detection · Self Supervised Learning · Satellite images

1 Introduction

The history of change detection can be traced back to the early days of remote sensing with one of the first examples being the use of aerial photography to identify agricultural land use changes [13]. Change detection is the process of comparing two images of the same area at different times to identify changes, such as urban growth and development, deforestation events and vegetation evolution. Those changes are depicted onto a change mask where usually a white pixel represents a change and a black pixel that nothing has changed.

Despite the large amount of data that is now available from programs like Copernicus and Landsat, there is still a lack of open labelled datasets using these images. This makes it difficult to compare and evaluate new proposed change detection algorithms. Sufficient labelled datasets are essential for developing supervised learning methods with deep neural networks.

A promising candidate in tackling this problem is Self Supervised Learning(SSL), a subset of Unsupervised Learning. The term supervision means that labels do exist

for the data on which a model is trained. However, as opposed to Supervised Learning where the data is annotated with the help of a human, in SSL the labels are extracted from the data itself. For this reason, SSL is able to use large amounts of data that are not annotated. Generally, computer vision pipelines that employ self-supervised learning involve performing two tasks, a pretext and a downstream task. The downstream task can be anything like classification or detection task, with insufficient annotated data samples. The pretext task is the self-supervised learning task solved to learn visual representations, with the aim of using the trained model weights obtained in the process, for the downstream one. In this work, we aim to leverage SSL in exploiting significant amounts of earth observation images to enhance change detection. We experiment with already proposed methodologies with some alterations of our own while also putting forward a fusion technique that aims to aid in the solution of said problem.

The remainder of the paper is organised as follows. First, in Section 2 we discuss past work that has been done in the Change Detection domain, focusing on the deep learning approaches and SSL techniques. Section 3 goes with great detail into the methodology we use to tackle this problem as well as it describes the proposed fusion approach. Section 4 presents the results of our experiments and last but not least, in Section 5 we propose some future areas of interest to be explored in regards to change detection in satellite images.

2 Related Work

Convolutional Neural Networks (CNNs) are the de facto architecture used when dealing with images of any kind. Daudt et al. [5] presented three fully convolutional siamese networks trained end-to-end using the Onera Satellite Change Detection [6] and the Air Change datasets [1]. Their first network is based on the U-net architecture [10], named Fully Convolutional Early Fusion (FC-EF), with the other two being its siamese extensions. Their goal was to train these networks solely with change detection datasets without any sort of pre-training. Despite achieving good results, the authors remarked that the unavailability of larger annotated datasets, for this specific task, was a limiting factor.

Trying to combat this, a lot of methods use various techniques based on transfer learning. While this is a valid option, most pre-trained models are not trained on remote sensing data, but usually RGB images where complete datasets are more abundant. This results in most of the satellite image's information (e.g. Sentinel 2 images have 13 bands) being unusable [5,6] for any given task.

Leenstra et al. [8] use Self Supervised Learning techniques to solve most of the limitations mentioned before. They defined two pretext tasks that were trained on the S2MTCP dataset [7]. One was made to discriminate between overlapping and non-overlapping patches, while the second was trained to minimize the distance between overlapping patches while maximizing the distance between non overlapping ones based on a modified triplet loss function.

Their network architecture, in both cases, is a siamese convolutional network that despite its simplicity produced great results. The Onera dataset again was used for evaluation on the pretext tasks and training for the downstream. For change detection they

used a discriminative model to extract features from bi-temporal images, and they fine-tuned the network to detect changes using either Change Vector Analysis (CVA) [2,3] with thresholding techniques like the Otsu and Triangle methods [11], or a simple linear classifier.

Chen et al. [4] on the other hand, chose to use a contrastive loss for the pretext task. The main difference between these two approaches is that triplet loss tries to ensure a margin between distances of negative pairs and distances of positive pairs while contrastive loss takes into account the margin value only when comparing dissimilar pairs, and it does not care at all where similar pairs are at that moment [12]. As a result, contrastive loss may reach a local minimum sooner, while triplet loss may continue to better organize the vector space.

For their network architecture, they chose a Siamese ResUnet [14] to obtain pixel-wise representations of the temporal different and spacial similar image patch pairs. For the downstream task they used the Otsu and the Rosin thresholding method [11] on the difference between the features of the two network branches to obtain the binary change mask.

In this paper we chose to experiment with Leenstra et al's approach because despite its simplicity it produced competitive results. We explored how 2 different band combinations affect the final's task results and we examined the impact of the network's size in regards to the change detection task. We also demonstrate a fusion technique that is able to use the predicted outputs from all the change detection classifiers to generate a better result.

3 Methodology

In this section we describe how Self Supervised learning, in this specific case, is used to help solve the problem of change detection in satellite images.

As it has been said in section 2 there are not many large annotated datasets specifically build for change detection in remote sensing data. Using SSL, large amounts of earth observation images can be used to train a pretext task on an unrelated job and then use those trained weights to solve a downstream task, which in our case is change detection. By doing so, we can pre-train the majority of the weights and then teach the model to recognize changes with a minimal amount of labels.

3.1 Network Architecture

The network's configuration changes in each task, either that being a pretext or the downstream one. In general the models are based on a Siamese Network's arrangement where the encoder part of the network consists of branches of convolutional layers while the decoder part is specific for each task. According to [8] two pretext tasks and one downstream task are defined. In Figure 1 there is an overview of the proposed change detection pipeline where each prediction is fused with the aim to produce a better one.

Pretext Task 1 This task asks the network to do similarity detection between overlapping and non-overlapping patches. Overlapping patches have been given the label

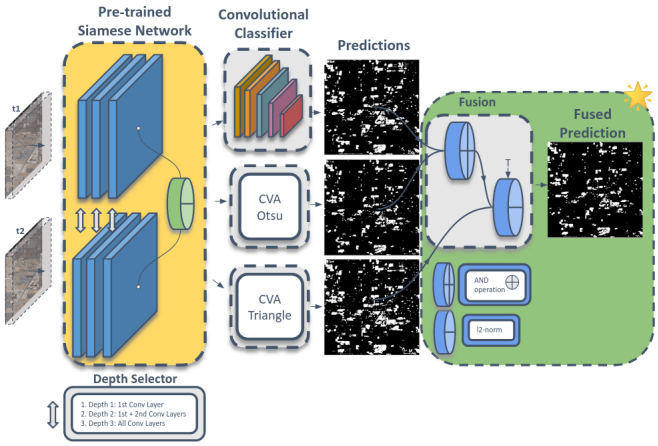


Fig. 1: Methodology Pipeline. The pre-trained siamese network uses convolutional brances trained on an unrelated task, while the Fusion technique exploits the change detection classifiers' results to produce a fused prediction

0 while non-ovelapping one the label 1. The proposed approach has a Siamese Network with two branches, where two images, spatially similar but temporally different, are used as inputs in each branch. Each branch consists of 3 convolutional layers with 32 filters, 3x3 size kernels each and a Relu activation function. After each convolution there is a dropout of 0.1. No pooling operations are implemented between each layer.

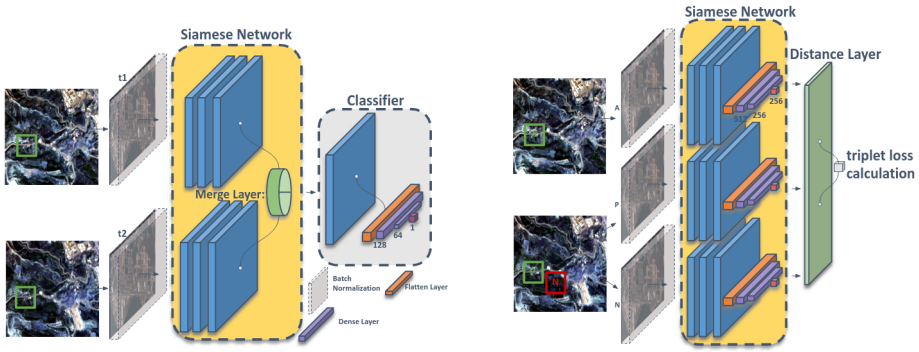
Features learned from each of the two embedding networks then fuse together passing through a layer that calculates their absolute difference (Merge Layer). Subsequently the resulting feature map is then passed through a classifier that tries to predict if the patches are overlapping or not. Figure 2a shows the network's architecture.

The loss function we use is Binary cross entropy1, as this is a binary classification problem. It compares each of the predicted probabilities to actual class output which can be either 0 or 1. It then calculates the score that penalizes the probabilities based on the distance from the expected value.

$$Loss = -\sum_{i=1}^{N}(y_i \log(\hat{y_i}) + (1 - y_i) \log(1 - \hat{y_i})) \tag{1}$$

where $\hat{y}$ the predicted label, y the true label and $i = (1...N)$ where $N =$ number of training samples.

Pretext Task 2 In this task the network is trying to minimize the triplet loss between 3 patches, a technique mainly used in face recognition algorithms [12]. These triplets are usually named Anchor (A), the Positive (P) and the Negative (N) patch. The Anchor and the Positive are overlaping patches while the Anchor and the Negative are non overlapping ones. We want the encodings (distance) of the Anchor and Positive images to



(a) Pretext task 1 (b) Pretext task 2

Fig. 2: The Pretext tasks' network architecture. Patches of size 96x96 pixels were taken from the two images.

be quite similar while the encodings of the Anchor and Negative images to be different. This can be seen in Equation 2,

$$Loss(A, P, N) = max(\|F(A) - F(P)\|^2 - \|F(A) - F(N)\|^2 + margin, 0) \leq 0 \tag{2}$$

where $F(X)$ is the encoding of the patch $X = (A, P, N)$ and the $margin$ is a value to keep negative samples far apart.

The Siamese network for this task consists of 3 branches, one for each patch, that have the same architecture as in the pretext task 1. A distance layer then is introduced, that calculates $\|F(A) - F(P)\|^2$ and $\|F(A) - F(N)\|^2$, and then the triplet loss is calculated. The networks architecture can be seen in Figure 2b.

Downstream Task Two methods were defined to solve the change detection problem, as mentioned also in Section 2. A linear classifier and Change Vector Analysis (CVA) with Otsu and Triangle Thresholding. We also add a third fusion method that takes the predictions of the other two and aims to produce a change mask that is better than the rest.

Change Vector Analysis (CVA) As suggested by Leenstra et al [8], CVA can be used to identify spectral changes between two identical images which were acquired at different times. In this case, CVA calculates the distance map, using the euclidean distance, between the two features produced by the pre-trained convolutional branches as shown in equation 3.

$$distance\,(F_1, F_2) = \sqrt{\sum_{i=1}^{N} (F_{1i} - F_{2i})^2} \tag{3}$$

F_1 and F_2 are the corresponding feature maps produced by the two branches of the pre-trained Siamese network and $i = (1...N)$. $N =$ number of pixels in each map.

Thresholding techniques can then be employed on this distance map, like the Otsu or the Triangle method, to produce the final predicted change mask.

Convolutional Classifier In place of the linear classifier, we propose a different approach that employs convolutional layers with 1x1 filters to produce a change mask. Those layers can also be called "Networks in Networks" as it's defined in [9]. Although a 1x1 filter does not learn any spatial patterns that occur within the image, it does learn patterns across the depth of the image. Therefore filters like that provide a method of dimensionality reduction as well as the benefit of enabling the network to learn more. We call this approach a convolutional classifier.

Five of such convolutional layers were used with 32, 16, 8, 4 and 2 filters respectively. The latter uses a softmax activation function to produce a 1x2 vector of probabilities. Each probability corresponds to the chance of one pixel being of the class change or no change. The first one is depicted by a white pixel in the change mask while the second one by a black one.

This convolutional classifier was trained in a Supervised manner on a small annotated satellite imagery dataset. Because the non-changes are more frequent in the ground truth change masks we use a weighted Categorical Cross Entropy as a loss function as seen in 4 to counteract this class imbalance.

$$Loss = -\sum_{i=1}^{N}(y_i \cdot \log(\hat{y_i}) \cdot weights) \tag{4}$$

The parameter $\hat{y}$ refers to the predicted label, y to the true label and $i = (1...N)$. $N =$ number of pixels in the training image.

The pre-trained weights from the convolutional branch in the pretext tasks were transferred to a Siamese configuration for the encoding part of the network. A depth selector was also introduced to be able to select exactly how many of the 3 pre-trained convolutional layers we need for the change detection task. The reason behind this is to determine if choosing earlier feature representations, versus later ones, has any effect on the performance of the downstream problem. This selection is done manually.

Fusion This approach aims to unify all the predicted change masks, produced from the previous methods, to generate a mask with the best results (Equation 5).

Let $\Omega_{cm} = \{0,1\}^{n \times m}$, where n = number of rows , m = number of columns, be the change map produced from each aforementioned methods, where $cm =$(Conv, Otsu, Triangle). From our experiments we discovered that the change mask generated by CVA with Triangle Thresholding ($\Omega_{triangle}$) usually was the best, but still it was failing to detect specific pixel changes that the other two methods (Ω_{conv} and Ω_{otsu}) could. Therefore, first, Fusion creates a similarities map (Ω_{sim}) between Ω_{conv} and Ω_{otsu} using the binary operator $\oplus$. Subsequently a threshold row-specific method is used between the $\Omega_{triangle}$ and the Ω_{sim} maps to keep only these specific pixels' areas.

In particular, for each row i, the method computes the $l2$-norm of the respective rows from the $\Omega_{triangle}$ and the Ω_{sim} method and it either keeps the one from the $\Omega_{triangle}$,

if the distance is greater than the given threshold T, or the other from the Ω_{sim}. This way we can enhance the "good" predictions of the CVA with Triangle Thresholding technique by exploiting the areas where the Otsu and the Convolutional Classifier methods are better, using a user defined parameter.

$$\begin{aligned} \Omega_{sim} &= \Omega_{conv} \oplus \Omega_{otsu} \\ \Omega_{Fused,i} &= \begin{cases} \Omega_{triangle,i}, & \text{if } l_2(\Omega_{triangle,i}, \Omega_{sim,i}) > T \\ \Omega_{sim,i}, & \text{otherwise} \end{cases} \\ &\text{for } i = 1, \ldots, n \end{aligned} \tag{5}$$

This algorithm assumes that one method always produces the overall better results and uses the others to complement those.

3.2 Datasets

To evaluate the proposed methods we employed two openly available change detection datasets. For the pretext tasks the Sentinel-2 Multitemporal Cities Pairs dataset (*S2MTCP* [7]) was used and the downstream task was applied on the Onera Satellite Change Detection dataset (*OSCD*[1]). The latter was also used for evaluation purposes on the pretext task 1.

The S2MTCP dataset is an urban change detection collection of 1,520 Sentinel-2 Level 1C (L1C) image pairs. It was created by Leenstra et al [8]. It does not contain change masks and its purpose is to teach the network feature representations in regards to the aforementioned pretext tasks. The images are roughly 600x600 in shape and contains all Sentinel-2 bands of the Level 1C product resampled to 10m.

Despite the fact that the S2MTCP dataset contains images with less than one percent cloud cover, some randomly taken patches contained mostly clouds. To avoid the performance loss, those cloudy images were discarded before training.

The OSCD dataset(Onera) contains 24 pairs of Sentinel-2 Level 2 multispectral images with pixel-level change ground truth maps for each pair. The annotated changes focus on urban changes. The images contain all 13 Sentinel-2 bands and it varies in spatial resolution between 10m, 20m and 60m. For the downstream task, we split the data in train and test groups as recommended [6]: 14 image pairs were used for training and 10 image pairs were used for testing. On the other hand for the pretext task 1 all 48 images were used. The reason for using the OSCD dataset, on this task, is to test the network's ability to discriminate between overlapping and non overlapping patches on images it has never seen before.

3.3 Settings

To augment the available data, patches were taken from each image, from both datasets, of size 96x96 pixels per patch. Random rotations, vertical and horizontal flips where also applied to each patch at the pre-processing step and not during training.

[1] `https://rcdaudt.github.io/oscd/`

Table 1: The dataset use in regards to its specific task. Patch pairs from the S2MTCP and Onera were employed both for the pretext task 1, with the latter used only for evaluation purposes. Patch triplets were used for the pretext task 2. Training on the downstream task was executed with the Onera patch pairs.

	S2MTCP			Onera	
Set	**% Splits**	**Patch Pairs**	**Patch Triplets**	**% Splits**	**Patch Pairs**
Train	85%	12776	6389	58.33%	1400
Validation	10%	1510	755	-	-
Test	5%	744	372	41.66%	1000

For the S2MTCP dataset, 10 patch pairs, either overlapping or non overlapping where randomly selected from each image pair, resulting in 15200 pairs, for the pretext task 1. For the second task, 8116 patch triplets where taken, as described in the Section 3.

For the Onera dataset, 480 patch pairs were generated with the same augmentations as in the S2MTCP for testing purposes on the pretext task 1. For the downstream task, 100 patches per image were extracted, resulting into 2400 patch pairs. Table 1 provides an overview of the data splits of the aforementioned datasets.

All the aforementioned methods were executed using Tensorflow on a RTX 3060 12gb GPU. Also the Adam optimizer was used with a learning rate of 0.001, $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 1e-7$ on all the networks. Lastly we applied a small weight decay of 0.0001 and we experimented with multiple band combinations from the Sentinel 2 images.

4 Results

Table 2 provides a quantitative evaluation of different approaches applied on the Onera dataset on 3 channel images. Leenstra's method uses the pretext task 1 methodology, as mention in Section 3 as well as CVA with Triangle Thresholding. The 3 Fully Convolutional Classifiers (FCCs) presented in this table 2 are solely trained on the Onera dataset (i.e. no pre-training). It is very notable that our Fusion method was able to produce the best F1 score, even when compared with state of the art methods as the FC-EF, while also having a Recall and Precision that is very competitive. Following, we present the different metrics for each task specifically (i.e pretexts and downstream).

Pretext task 1 For this task, where we predict if two patches are spatially overlapping, we present the loss and the accuracy of the band combinations: B2(blue), B3(green) and B4(red) and B2,B3,B4,B8(VNIR) in the Table 3.

The confusion matrices that were produced by the predictions versus the actual label values can be seen in Table 4a 4b for B2B3B4 and Table 4c 4d for B2B3B4B8.

These results show that despite the band combination, the network managed to learn each task. Notable is though that it produced better metrics and better predictions when using Bands 2, 3 and 4. The Accuracy achieved is very high despite the data splits and

Table 2: Comparison between the quantitative results of our best methods with a similar SSL methodology and 3 FCCs architectures

	Method	F1	Recall	Precision
SSL	Pretext task 1 & Fusion	**48.91**	43.40	56.03
	Pretext task 1 & Triangle	48.82	42.51	**57.33**
	Leenstra [8]	38.71	41.14	36.55
FCCs	EF [5]	34.15	82.14	21.56
	FC-EF [5]	48.89	**53.92**	44.72
	FC-Siam-diff [5]	48.86	47.94	49.81

Table 3: Pretext Task 1 results for different band combinations

Dataset	Data Split	Loss		Accuracy	
		Bands 2,3,4	Bands 2,3,4,8	Bands 2,3,4	Bands 2,3,4,8
S2MTCP	Validation	0.0790	0.0880	98.27%	97.22%
S2MTCP	Test	0.0478	0.0920	98.65%	98.12%
Onera	All	0.1644	0.1845	94.16%	93.75%

Table 4: Confusion matrices on the pretext task 1 ((a),(b) for bands 2,3,4 and (c),(d) for bands 2,3,4,8). The 0 label means an overlapping patch pair while 1 a non overlapping.

(a) S2MTCP, bands 2,3 and 4

Test Set		Predicted	
		(0)	(1)
True	(0)	352	7
	(1)	3	382

(b) Onera, bands 2,3 and 4

Full Set		Predicted	
		(0)	(1)
True	(0)	108	12
	(1)	2	118

(c) S2MTCP, bands 2,3,4 and 8

Test Set		Predicted	
		(0)	(1)
True	(0)	367	14
	(1)	0	362

(d) Onera, bands 2,3,4 and 8

Full Set		Predicted	
		(0)	(1)
True	(0)	106	14
	(1)	1	119

the dataset used for validation. This might be due to the fact that the network is trying to solve a simple task.

Pretext task 2 This task tries to minimize the Triplet loss. In other words, it is trying to produce encodings between the Anchor and Positive patches that are quite similar and encodings between the Anchor and Negative patches that are not, as described in the Methodology 3. A way to evaluate if this has been achieved is by calculating the cosine similarity between the encodings of patch pairs. We should expect the similarity between the Anchor and Positive patches to be larger than the similarity between the Anchor and the Negative ones. In Table 5 we present the mean values of the positive and negative similarities for the test and validation sets of the S2MTCP dataset.

In both cases, of different band combinations, we see that indeed the positive similarity is greater than the negative one, despite the dataset split, with the difference on the second one (B2, B3, B4, B8) being slightly higher.

Table 5: Evaluation of the Pretext task 2 using Cosine Similarity

(a) For Bands 2, 3, 4

Cosine similarities		
Data Split	A - P	A - N
Test	-0.9931156	-0.97266644
validation	-0.9931706	-0.97263557

(b) For Bands 2, 3, 4, 8

Cosine similarities		
Data Split	A - P	A - N
Test	-0.99086726	-0.96263564
validation	-0.9901459	-0.96268386

Change Detection Table 6 contains the quantitative evaluation of the proposed downstream tasks when their weights are pre-trained on the Pretext Task 1 (Table 6a) and Pretext Task 2 (Table 6b). Experiments where enacted using two different band combinations, as well as either using the full pre-trained Siamese network from the pretext task (Depth: 3) or only the 2 pre-trained convolutional layers (Depth: 2). Notable is the fact that the latter, in all the cases produced better results. This happens due to the fact that earlier layers in the Siamese network produce more general image encodings while later ones are more task specific.

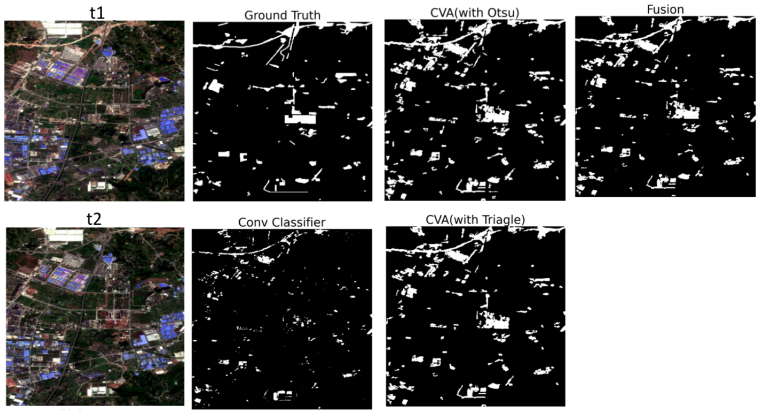


Fig. 3: Predicted change masks on the Chongqing city, an image pair from the Onera Test Set. The Ground Truth mask shows the actual changes that occur between the image pair

The convolutional classifier also is generating better results when trained with bands 2,3,4 and 8 rather than just bands 2,3 and 4, a testament to the fact that neural networks will perform better when given more informantion. Classic techniques though as, in our case, Change Vector Analysis with the Triangle Thresholding overall produced the best results even when using bands 2,3 and 4.

Fusion was able to keep the different metrics as close as it is possible to the best downstream method (Cva with Triangle) and it was able to improve the predictions in

regards to some metrics as the F1 score and the Specificity. The thresholding parameter T, as mentioned in Methodology 3, for every case, was set to 3.5.

Table 6: Change detection results using all the downstream methods with Pretext tasks' pre-trained weights, Pretext task 1 (a), Pretext task 2 (b), for two different network depths and with two different band combinations.

(a)

Pretext Task 1		Conv classifier		Cva + Otsu		Cva + Triangle		Fusion	
		Bands 2,3,4	Bands 2,3,4,8	Bands 2,3,4	Bands 2,3,4,8	Bands 2,3,4	Bands 2,3,4,8	Bands 2,3,4	Bands 2,3,4,8
Depth: 2	Sensitivity	44.50	50.35	**75.39**	70.50	42.51	62.79	43.40	62.24
	Specificity	93.89	96.05	86.50	87.66	**98.56**	94.14	98.45	94.21
	Precision	24.89	36.72	20.25	20.63	**57.33**	32.78	56.03	32.85
	F1	31.92	42.47	31.93	31.92	48.82	43.08	**48.91**	43.01
	Accuracy	91.74	94.07	86.02	86.92	**96.12**	92.78	96.06	92.82
Depth: 3	Sensitivity	37.62	43.90	**71.03**	70.56	50.97	62.76	49.84	61.61
	Specificity	95.14	96.78	87.97	86.97	97.26	94.22	**97.26**	94.36
	Precision	26.05	38.25	21.18	19.77	**45.79**	33.05	45.32	33.18
	F1	30.78	40.88	32.63	30.88	**48.24**	43.30	47.47	43.13
	Accuracy	92.64	94.48	87.24	86.26	**95.24**	92.85	95.20	92.93

(b)

Pretext Task 2		Conv classifier		Cva + Otsu		Cva + Triangle		Fusion	
		Bands 2,3,4	Bands 2,3,4,8	Bands 2,3,4	Bands 2,3,4,8	Bands 2,3,4	Bands 2,3,4,8	Bands 2,3,4	Bands 2,3,4,8
Depth: 2	Sensitivity	35.33	37.26	**72.12**	74.52	44.97	45.35	45.27	45.41
	Specificity	95.12	97.51	85.05	85.45	**98.26**	98.00	98.19	98.00
	Precision	24.77	40.52	18.00	18.89	**53.97**	50.82	53.18	50.79
	F1	29.13	38.82	28.80	30.14	**49.06**	47.93	48.91	47.95
	Accuracy	92.52	95.70	84.50	84.97	**95.94**	95.71	95.89	95.71
Depth: 3	Sensitivity	26.78	35.13	66.64	63.28	**82.31**	63.67	82.30	62.97
	Specificity	**96.52**	97.80	83.99	89.97	64.46	92.75	64.47	92.88
	Precision	25.95	**42.11**	15.92	22.30	9.53	28.53	9.53	28.69
	F1	26.36	38.30	25.70	32.98	17.08	39.41	17.08	**39.42**
	Accuracy	93.49	**95.08**	83.23	88.81	65.23	91.48	65.25	91.58

These results were produced, when evaluating the change detection methods on the Onera Test set, using the full size images. Figure 3 presents the outputs from all the classifiers, and the fusion, when the network is given an image pair.

5 Conclusion

Self Supervised learning (SSL) allows for the use of large unlabelled earth observation datasets i.e S2MTCP, in aiding the performance of change detection networks. This process can be split into two tasks, the pretext task and the downstream, where by using insufficient annotated data samples i.e Onera Dataset, the second is able to enhance its performance by leveraging the model weights of the first.

In this work, we presented 4 SSL pipelines for change detection, 3 of them were inspired by [8]. We experimented with two different Sentinel-2 band combinations, as well as with using different amounts of pre-trained convolutional layers for the downstream network. We proposed a Fusion technique that, given the right parameters, could

be able to use predictions from different methodologies to create a better one. And last but not least, we showed that SSL on a small network was able to produce competitive results.

Future work will focus on tackling the question of which bands should be used specifically for change detection as well as what pre-processing techniques should be employed for satellite images. Moreover, it is of great interest to discover new pretext tasks that will be more difficult to solve, so we can explore if the difficulty of the task impacts the discovery of changes. And last but not least, fusion methodologies that could work online while training neural networks is an area of interest worth exploring.

Acknowledgements This work was supported by the EU's Horizon 2020 research and innovation programme under grant agreements H2020-883484 PathoCERT and H2020-101004152 CALLISTO.

References

1. Benedek, C., Sziranyi, T.: Change detection in optical aerial images by a multilayer conditional mixed markov model. IEEE Transactions on Geoscience and Remote Sensing **47**(10), 3416–3430 (2009)
2. Bovolo, F., Bruzzone, L.: The time variable in data fusion: A change detection perspective. IEEE Geoscience and Remote Sensing Magazine **3**(3), 8–26 (2015)
3. Bruzzone, L., Bovolo, F.: A novel framework for the design of change-detection systems for very-high-resolution remote sensing images. Proceedings of the IEEE **101**(3), 609–630 (2013)
4. Chen, Y., Bruzzone, L.: Self-supervised remote sensing images change detection at pixel-level (2021)
5. Daudt, R.C., Saux, B.L., Boulch, A.: Fully convolutional siamese networks for change detection (2018)
6. Daudt, R.C., Saux, B.L., Boulch, A., Gousseau, Y.: Urban change detection for multispectral earth observation using convolutional neural networks (2018)
7. Leenstra, M., Marcos, D., Bovolo, F., Tuia, D.: Sentinel-2 multitemporal cities pairs (Nov 2020), `https://doi.org/10.5281/zenodo.4280482`
8. Leenstra, M., Marcos, D., Bovolo, F., Tuia, D.: Self-supervised pre-training enhances change detection in sentinel-2 imagery (2021)
9. Lin, M., Chen, Q., Yan, S.: Network in network (2013)
10. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation (2015)
11. Rosin, P.L.: Unimodal thresholding. Pattern Recognition **34**(11), 2083–2096 (2001)
12. Schroff, F., Kalenichenko, D., Philbin, J.: Facenet: A unified embedding for face recognition and clustering. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 815–823 (2015)
13. SINGH, A.: Review article digital change detection techniques using remotely-sensed data. International Journal of Remote Sensing **10**(6), 989–1003 (1989)
14. Zhang, Z., Liu, Q., Wang, Y.: Road extraction by deep residual u-net. IEEE Geoscience and Remote Sensing Letters **15**(5), 749–753 (2018)