Object-Based Change Detection in Satellite Images Combined with Neural Network Autoencoder Feature Extraction

Conference Paper · November 2019

DOI: 10.1109/IPTA.2019.8936085

Ekaterina Kalinicheva
Institut national de l'information géographique et forestière
10 PUBLICATIONS **80** CITATIONS
SEE PROFILE

Jeremie Sublime
Institut Supérieur d'Electronique de Paris
43 PUBLICATIONS **237** CITATIONS
SEE PROFILE

Maria Trocan
Institut Supérieur d'Electronique de Paris
97 PUBLICATIONS **524** CITATIONS
SEE PROFILE

Object-Based Change Detection in Satellite Images Combined with Neural Network Autoencoder Feature Extraction

Ekaterina Kalinicheva*, Jérémie Sublime * and Maria Trocan*
* ISEP - LISITE laboratory, DaSSIP team, 10 rue de Vanves, 92130 Issy-Les-Moulineaux, France
e-mail: e.kalinicheva@isep.fr, jeremie.sublime@isep.fr, maria.trocan@isep.fr

***Abstract*—Change detection is a challenging task in the field of remote sensing. While most applications use supervised change detection and classification techniques, it still remains a difficult task to create a training database for generic applications. In this paper, we propose an end-to-end unsupervised change detection and clustering baseline. The presented baseline firstly deploys a neural network autoencoder for feature extraction and comparison to detect some meaningful changes. These changes are further segmented with 3D graph-based techniques and clustered. The presented algorithm gives promising results and is fully unsupervised.**

***Index Terms*—unsupervised change detection, satellite images, autoencoder, segmentation, clustering, feature extraction.**

I. Introduction

Change detection between two bi-temporal satellite images has been a challenging task for many researchers over the years. However, most of the change detection algorithms are developed for a specific application or area and use classified training data that cannot easily be re-used. For example, in [1] the authors have developed a siamese neural network model for supervised change detection in urban areas with open source multispectral images. In [2], the authors propose an automatic landslides detection algorithm based on the comparison of the features extracted using a convolutional network pre-trained on the DeepSat dataset [3]. Meanwhile, due to the variety of satellite data (different resolution, number of bands, etc), it might be too costly and complicated to create a universal database that can be used for a generic algorithm. To tackle this issue, different unsupervised change detection approaches have been developed.

The main difficulty with unsupervised methods is that their accuracy is usually much lower than for supervised ones. For this reason, researchers often fuse several unsupervised algorithms results to obtain more robust change detection results [4]. In another type of change detection approach, the authors propose the automatic detection of changed and unchanged pixels [5] that are further used as training samples for a chosen classification algorithm. At the same time, traditional object-based methods with the comparison of the segment borders and their features for two bi-temporal images still remain one of the most popular [6]. This approach is performed directly at object level which facilitates the interpretation of the detected changes.

Nevertheless, most of the classic change detection approaches do not separate trivial (seasonal) changes from non-trivial ones (permanent changes and changes that do not follow seasonal tendency). This issue can seriously complicate the interpretation of detected change areas in images that contain a high vegetation ratio, as all seasonal changes might be classified as changes. Finally, deep learning algorithms have proved to be effective at modeling the complex and non-linear transformations between multi-temporal bands and classes or clusters of changes in change detection applications [7]. Following this problematic, in [7] the authors introduce a Restricted-Boltzmann Machines-based (RBM) method to model a translation between two images in order to detect non-trivial changes.

In this paper, we propose an end-to-end approach to detect and cluster non-trivial changes between two bi-temporal images. This method combines patch-wise analysis of image features with convolutional autoencoders and object extraction. Contrarily to many object-based change detection approaches where the authors compare features extracted from the potential change segments, we firstly perform the change detection based on the patch-wise feature translation and then deploy the graph-based methods to define the change objects that are further clustered.

In the second section of this paper, we describe the proposed change detection and clustering baseline. Then, in the third section, we present the data and some experimental results. Finally, section 4 ends this paper with a conclusion and future perspectives.

II. Proposed algorithm

The proposed method aims to detect non-trivial change areas between two bi-temporal images in an unsupervised manner. Thereafter, it defines the change objects within these areas that are further clustered and associated to different change behavior.

The algorithm baseline is presented in Figure 1. Let Im_1 and Im_2 be a couple of co-registered bi-temporal images for

978-1-7281-3975-3/19/$31.00 ©2019 IEEE

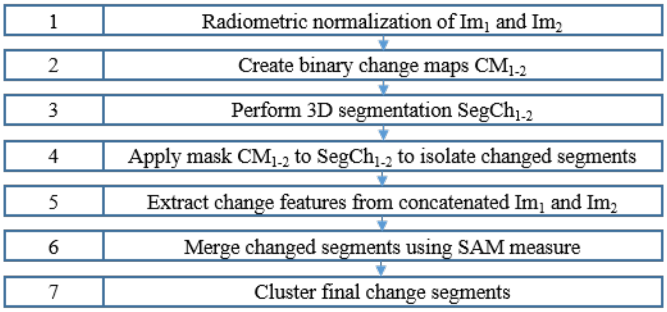
1 Radiometric normalization of Im1 and Im2
2 Create binary change maps CM1-2
3 Perform 3D segmentation SegCh1-2
4 Apply mask CM1-2 to SegCh1-2 to isolate changed segments
5 Extract change features from concatenated Im1 and Im2
6 Merge changed segments using SAM measure
7 Cluster final change segments

Fig. 1. Proposed baseline.

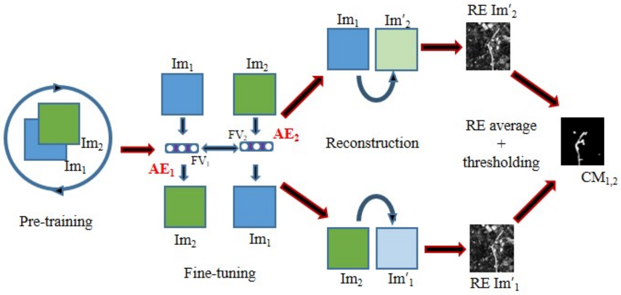

Fig. 2. Change detection algorithm.

change detection. First of all, we start with a relative radiometric normalization of Im_1 and Im_2 [8] so the pixel values of the non-changed areas have the same range for both images. Then, we proceed with the detection of non-trivial change areas using a joint AutoEncoders (AEs) transformation model to generate a binary change map CM_{1-2}. As every isolated change area from CM_{1-2} might potentially contain several change classes, we use a 3D graph-based segmentation algorithm to detect different objects within these areas. Once the non-trivial change areas are defined, we perform a feature extraction for these areas using a convolutional AE. The extracted features will further be used for the segments analysis and clustering. As most of the segmentation algorithms produce over-segmented results, we merge the segments within each isolated change area using the Spectral Angle Mapper (SAM) dissimilarity criteria [9]. Finally, during the last step, the final segments are regrouped in clusters with classical clustering algorithms and associated to different change behavior.

A. Change detection

The principle of change detection between two bi-temporal images is based on a joint AEs model that is able to reconstruct Im_1 from Im_2 and vice versa by learning the translation of image features [10]. Areas with no changes or trivial changes will be easily translated by the model and will have a small reconstruction error. At the same time, as non-trivial changes are less numerous and unique, they will be considered as outliers and will have high reconstruction errors that can be used to map areas of non-trivial changes.

The steps for our change detection algorithm are the followings, see Figure 2:

- The first step consists in pre-training the model on both images Im_1 and Im_2 for self-encoding.
- During the second step, we fine-tune the joint AE model by mapping Im_1 to Im_2 and Im_2 to Im_1.
- Once the model is trained, we perform patch-wise reconstruction of Im'_2 from Im_1 and vise versa and simultaneously compute patch reconstruction errors (RE). In other words, the reconstruction error of every patch is associated to the position of its central pixel on the image.
- In the last step, we identify the areas of high reconstruction error using automatic Otsu's thresholding method [11] in order to create a binary change map CM_{1-2} of the non-trivial changes.

Our feature learning and translation model is based on convolutional AE. An encoding part of the convolutional AE is composed of a set of convolution filters for feature extraction. They are followed by fully-connected layers that compress extracted features maps in feature vectors (FV). The decoding part is symmetrical to the encoding one and aims to reconstruct the input data from the encoded feature vector.

We start the change detection process with the model pre-training. The pre-training is performed in a patch-wise manner on patches of size $p \times p$ extracted for every pixel of the images Im_1 and Im_2. During the pre-training, we aim to encode all i, j, m-patchs of chosen samples into feature vectors, and then during the decoding pass, transform them back to the initial i, j, m-patch ($i \in [1, H]$, $j \in [1, W]$, $m \in [1, 2]$, where H is the images height, W is the images width). The patch reconstruction mean squared error (MSE) is chosen as the loss function for the model optimization.

Once the model is stabilized, we move on to the fine-tuning of the joint AE model (Figure 2). The model is composed of two joint autoencoders - AE1 and AE2 - that perform the transformation of Im_1 to Im_2 and of Im_2 to Im_1 respectively. Both AEs have the same structure as the pre-trained model, and are initialized with the parameters of the pretraining phase. While training, the joint model aims to minimize the patch reconstruction errors for AE1 and AE2 and the MSE between the outputs of their bottlenecks (Figure 2, Fine-tuning). After the model is fine-tuned, we perform the reconstruction of Im'_2 from Im_1 and Im'_1 from Im_2 and calculate the reconstruction error (MSE) for every patch. Two images with reconstruction error values $RE\_Im'_1$ and $RE\_Im'_2$ are then obtained. Finally, we apply Otsu's thresholding on average of these images to create a binary change map CM_{1-2}. Before applying the thresholding, we exclude 0.5% of the highest RE values under the hypothesis that they correspond to some noise and extreme outliers.

B. 3D segmentation

To cluster the obtained changes, we chose an object-based approach that uses 3D segmentation based on graph analysis and the tree-merging algorithm. The principle of 3D segmentation is slightly different from a single image segmentation. For a single image, the image pixels are presented as a graph of nodes that are connected to their neighbors. Some similarity measure is associated to the edge weights between these nodes (Figure 3 (a)). The nodes with the highest connectivity values are then regrouped as segments using a chosen graph-cut algorithm. For simplicity reasons, in Figure 3 (a), we present a 4-connected neighborhood, although most of algorithms use full 8-connected neighborhood.

In the case of 3D segmentation, the graph grid becomes more complex. Every i, j-pixel of the first image is connected to its 4 or 8 local neighbors and also to the i, j-pixel of the second image (Figure 3 (b)) which gives us 5- or 9-connected neighborhood depending on the considered connectivity degree. Additionally, the i, j-pixel of the first image can be connected to its 8 neighbors both locally and in the second image, which gives us the maximum possible connectivity of 17.

After performing the segmentation of the 3D graph, we obtain the segments that may belong to only one image, or span on both images. Obviously, if i, j-pixel from the first and the second images belongs to the same segment, it corresponds to a non-changed object (Figure 3 (c)).

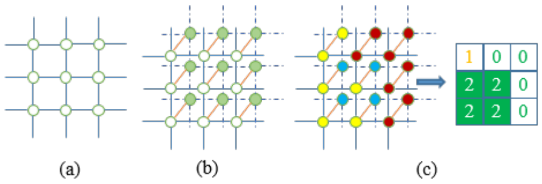

Fig. 3. Graphs examples. Circles and lines represent nodes (pixels) and edges (similarity measure). (a)- graph constructed for a single image, (b)- 3D graph, white nodes correspond to the first image and filled to the second one, (c)- the results of 3D graph segmentation and corresponding change objects, the colors of nodes correspond to different segments, if i, j-pixel of both images correspond to the same segment, clearly, it belongs to a non-change area. Indices 0 and 1 correspond to non-change objects and index 2 corresponds to a change object at the resulting change segmentation. For the simplicity of representation, we do not show the diagonal links between nodes.

For the proposed baseline, we chose a tree-merging bottom-up segmentation algorithm [12] due to its relatively fast performance and good results comparatively to other graph-based algorithms [13]. The principle of the tree-merging algorithm is the following: initially, every pixel belongs to a separate segment, then, in each step of the algorithm, the adjacent segments are merged if the difference between them is not much less than the internal variability of these segments. Contrary to the original algorithm presented in [12] that uses the Euclidean distance, we use Mahalanobis distance between two pixels as the edge weight (1):

$$d\left(\vec{x}, \vec{y}\right) = \sqrt{\left(\vec{x} - \vec{y}\right)^{\top} S^{-1} \left(\vec{x} - \vec{y}\right)}, \qquad (1)$$

where $\vec{x}$, $\vec{y}$ are two pixels values and S is the covariance matrix of these images. The proposed algorithm requires three parameters which are σ- the standard deviation for Gaussian kernel that is used for image smoothing during the preprocessing phase, k- the constant for the thresholding function used for segment merging, and min_size that defines the minimum component size for the post-processing stage.

After the segmentation is applied to the couple of images, we obtain the segmentation results Seg_1 and Seg_2 for Im_1 and Im_2 respectively, their intersection as in Figure 3 (c) gives us the change segments $SegCh_{1-2}$. Clearly, this 3D segmentation algorithm cannot be used as a standalone approach for non-trivial change detection for most of the satellite image couples as it may detect numerous false positive changes caused by the variations in image luminosity and trivial changes. At the same time, this method gives us a high ratio of true positive changes and as well as segments for the different types of changes (or lack of change). To isolate the non-trivial change objects from the others, we apply a change mask CM_{1-2} to the results of 3D segmentation $SegCh_{1-2}$.

Obviously, the borders of binary change areas from CM_{1-2} do not completely superpose with change objects borders from $SegCh_{1-2}$. Therefore, change areas will contain some pixels that are detected by the 3D segmentation algorithm as unchanged. We consider the change areas borders CM_{1-2} as the references as this algorithm has a high precision of the change detection and low ratio of false positive changes. The pixels that do not belong to any change segments have to be attached to an existing neighbor segments or group the new ones. The strategy is the following: if the size of a group of adjacent pixels from $SegCh_{1-2}$ that do not belong to any change segment does not exceed 3 pixels, they are merged with the largest neighbor segment; otherwise, these pixels are regrouped in the same segments as for the non-changed areas from Seg_1 (or Seg_2 as their segment borders are identical).

C. Feature extraction

To continue with the analysis of detected change areas, we extract their features. The features are extracted with deep convolutional AE in a patch-wise manner for all corresponding change pixels of the concatenated images Im_1 and Im_2. The training of the feature extraction model resembles to the pre-training of the change detection model and consists in the reconstruction of input patches at the end of encoding-decoding cycle. Once the model is trained and stabilized, we extract its encoding part to encode every i, j-patch in its corresponding feature vector.

D. Segment fusion and clustering

After defining the change segments within the change areas, we proceed to the fusion of adjacent segments as they often belong to the same class. For this purpose, we use the SAM dissimilarity criteria (2) calculated for the median feature values within a change segment:

$$\theta(x,y) = \cos^{-1}\left(\frac{\sum_{i=1}^{n} x_i y_i}{\left(\sum_{i=1}^{n} x_i^2\right)^{\frac{1}{2}} \times \left(\sum_{i=1}^{n} y_i^2\right)^{\frac{1}{2}}}\right), \quad (2)$$

where x_i and y_i are the median value of $i-th$ feature of the first and second adjacent segments correspondingly, n is the number of features. The SAM value always vary between 0 and $\pi/2$. Once the SAM criteria is obtained for all adjacent segments, we apply Otsu's thresholding to its values and then all adjacent segments with SAM inferior to this value are merged.

At the last step, we cluster the obtained change segments by using their median feature values as segment descriptors.

III. Experimental framework

A. Dataset

Our algorithm was applied to a SPOT-5 dataset of the Montpellier area, France, taken between 2002 and 2008. The dataset belongs to the archive Spot World Heritage (available on www.theia-landsat.cnes.fr). This particular dataset was chosen due to its high ratio of agricultural zones and progressive construction of new areas. The preprocessing level of the images is 1C (orthorectified images, reflectance of the top of atmosphere). The images are composed of green, red, NIR and SWIR bands with 10 meters resolution. The original images are clipped to rectangular shapes of 1600×1700 pixels and transformed to UTM zone 31N: EPSG Projection (transformation from geographic coordinate system in degree coordinates to a projected coordinate system in meter coordinates).

B. Results

We assess the algorithms performances on two extracts from the dataset: an image couple taken in May 2004 and April 2005 for the first extract, and in February 2006 and August 2008 for the second couple. To evaluate the proposed approach, we compare the obtained results with ground truth classification change maps. These ground truths were created for an extract of the image of size 800×600 pixels (48 km^2) for the first couple and for 320×270 pixels (8,64 km^2) for the second one. However, the change detection and clustering was performed on the full images of 1600×1700 pixels (272 km^2).

The convolutional AE model for our change detection algorithm is presented in Table I, where B is the number of spectral bands and p is the patch size. The kernel size of 3 and stride of 1 were used for all the convolution layers. To speed up the training and to obtain more robust results, we kept only green, red and NIR bands (B=3) as they contain the essential information that is sufficient for change detection. The patch size of 5×5 pixels was chosen [10]. Note that as we have two couples of images from the same dataset, we pre-train the model only once with the patched extracted from four images.

TABLE I
Models architecture.

	Change detection	Feature extraction
encoder	Conv(B,32)+ReLU Conv(32,32)+ReLU Conv(32,64)+ReLU Conv(64,64)+ReLU Linear(64×p^2, 12×p^2)+ReLU Linear(12×p^2, 2×p^2)+ℓ_2-$norm$	Conv(B,32)+ReLU Conv(32,32)+ReLU Conv(32,64)+ReLU Conv(64,64)+ReLU MaxPooling(kernel=3, stride=3) Conv(64,128)+ReLU Conv(128,128)+ReLU MaxPooling(kernel=3) Linear(128,64)+ReLU Linear(64,32)+ReLU Linear(32,n)+ℓ_2-$norm$
decoder	Linear(2×p^2, 12×p^2)+ReLU Linear(12×p^2, 64×p^2)+ReLU Conv(64,64)+ReLU Conv(64,32)+ReLU Conv(32,32)+ReLU Conv(32,B)+Sigmoid	Linear(n,32)+ReLU Linear(32,64)+ReLU Linear(64,128)+ReLU UnPooling(kernel=3) Conv(128,128)+ReLU Conv(128,64)+ReLU UnPooling(kernel=3, stride=3) Conv(64,64)+ReLU Conv(64,32)+ReLU Conv(32,32)+ReLU Conv(32,B)+ReLU

We compute the precision, recall, accuracy, and Cohen's kappa score [14] and compare our results with these of [7] as, to our knowledge, it is the only state-of-art algorithm specialized on the detection of non-trivial changes. This algorithm is initialized with the same parameters as in the original article. The comparison between two algorithms is presented in Table II.

TABLE II
Performance of change detection algorithms based on convolutional joint AEs model.

		Classification performance		
Images	**Method**	***Precision***	***Recall***	***Kappa***
2004	Conv. AE	0.68	0.79	0.71
2005	RBM	0.48	0.64	0.52
2006	Conv. AE	0.76	0.79	0.75
2008	RBM	0.40	0.61	0.43

Some visual results are presented in Figures 4 and 5. All images are represented in false colors, where red corresponds to the vegetation and green to empty fields or bare soil. As the first couple of images was taken during the same period of the year, the vegetation does not change much between the images. However, the second couple of images was taken in different seasons of the year - on February 2006 and August 2008 - which gives us different overall seasonal tendency: the fields that are empty in February have vegetation in August, and vice versa, the forest vegetation has some minor changes too.

Figure 4 shows the construction of a new road. The road limits were correctly identified by both models, although we can see that that the RBM method is very sensitive to luminosity that causes most of the false positive changes in urban areas.

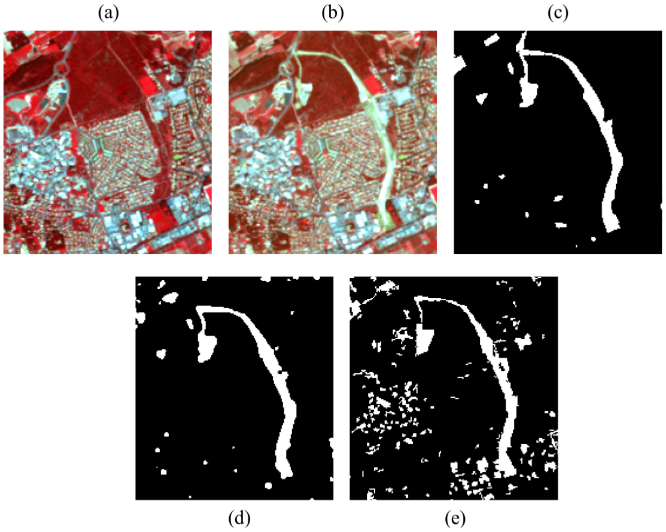

Fig. 4. Change detection results. Image extract 180×190 pixels. (a)- image taken on May 2004, (b) - image taken on April 2005, (c)- ground truth, (d)- convolutional AE, (e)- RBM.

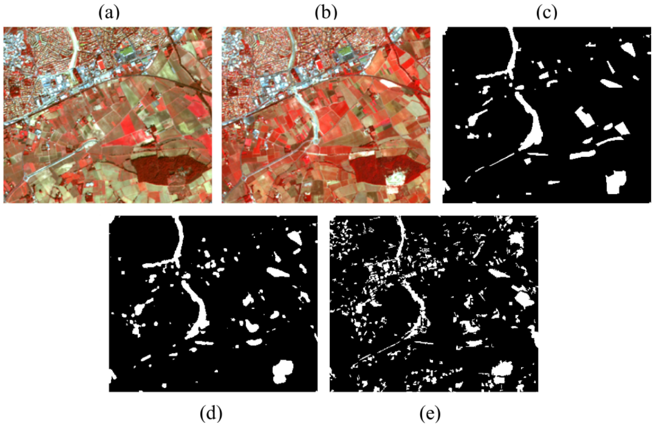

Fig. 5. Change detection results. Image extract 320×270 pixels. (a)- image taken on February 2006, (b)- image taken on August 2008, (c)- ground truth, (d)- convolutional AE, (e)- RBM.

Figure 5 represents changes in an agricultural area between February 2006 and August 2008 as well as some constructions. We can see that the ratio of false positive changes is elevated. Moreover, in many cases, only a part of a field is incorrectly labeled as a change. It can be explained by non-homogeneous vegetation densities, and a further morphological analysis might improve the results.

For the next step, we continue with the detection of change objects using the 3D segmentation tree-merging algorithm. We use the following parameters values that were chosen empirically and give the best segment shapes for the test data: σ=0.3, k=3 min_size=5. To lower the graph complexity, 9-connected neighborhood was chosen.

When the change maps are obtained for both couples of images, we extract the change features. Firstly, we concatenate two images, then we apply the change mask CM_{1-2} to obtain feature vectors for every patch of the corresponding pixel within the change areas. The scheme of convolutional AEs for feature extraction is presented in Table I. As the extracted features will be further used for the SAM measure that was originally developed for hyperspectral images, it was decided to encode the patches in 10 features. We use all four image bands (B=4) to obtain more robust change features for further clustering. The patch size of p=9 was chosen for the proposed model. The proposed patch size is large enough to get all the information about neighborhood patterns, but at the same time, the border pixel effect does not influence the results too much.

The extracted features are used to merge the preliminary change segments obtained with 3D segmentation into the final change objects basing on SAM criteria.

Different clustering algorithms combined with the proposed baseline were tested in order to find the most efficient one. These algorithms are k-means, hierarchical and spectral clustering. Our baseline is compared to pixel-wise k-means clustering of change areas and patch-wise DEC clustering [15] that uses the same AE model as for feature extraction. We also highlight the advantages of the proposed baseline comparing it to a baseline without region merging (called *Baseline w/o merging*, steps 1-2-3-4-5-7, see Figure 1) combined with different clustering algorithms, and baseline without feature extraction combined with different clustering algorithms applied directly to raw image values (called *Baseline w/o features*, steps 1-2-3-5-6-7). We use normalized mutual information (NMI) and adjusted rand index (ARI) to evaluate the clustering results.

In our dataset we detect different global classes of changes. For the ground truth extract of the couple of images taken in 2004 and 2005 years these classes are:

1) vegetation → construction,
2) construction in a dense area,
3) vegetation → empty field,
4) empty field → vegetation.

For the couple of images taken in 2006 and 2008:

1) sparse vegetation → beginning of the construction/bare soil,
2) construction field → end of the construction,
3) construction field/empty field → sparse vegetation.

For the first couple of images, we perform the clustering with 6 classes that are associated to the four previously indicated classes plus the class ”changes in water” that is not presented in the ground truth extract, and the class ”others” that corresponds to the false positive changes that are mostly changes in vegetation density. For the second couple of images, we chose 5 clusters that correspond to the 3 previously mentioned classes plus the classes ”changes in water” and ”others” that correspond to the false positive changes that are mostly due to changes in crop cultures. Our proposed baseline is evaluated only for the areas that were correctly detected as changes to avoid error accumulation. The performance of our

TABLE III
CHANGE CLUSTERING PERFORMANCE FOR DIFFERENT APPROACHES.

	Methods	Clustering performance					
		Hierarchical		k-means		Spectral	
2004-2005	K-means	-	-	0.38	0.40	-	-
	DEC	-	-	0.39	0.30	-	-
	Baseline w/o merging	0.50	0.44	0.51	0.50	0.46	0.48
	Baseline w/o features	0.51	0.46	0.48	0.45	-	-
	Proposed baseline	**0.60**	**0.62**	0.58	0.53	0.39	0.33
2006-2008	K-means	-	-	0.41	0.39	-	-
	DEC	-	-	0.32	0.30	-	-
	Baseline w/o merging	0.44	0.41	0.44	0.44	0.47	0.44
	Baseline w/o features	0.41	0.47	0.40	0.46	-	-
	Proposed baseline	**0.64**	**0.61**	0.36	0.31	0.59	0.56

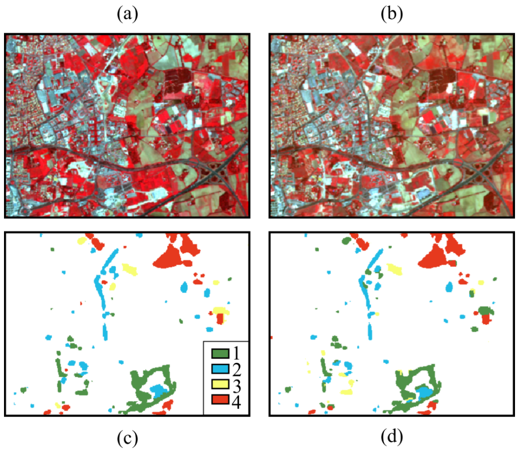
(a)
(b)
(c)
(d)

Fig. 6. Clustering results. Image extract 320×270 pixels.

proposed approach and concurrent methods are presented in Table III.

By observing the results, we can see that the proposed baseline with hierarchical clustering performed better than the concurrent methods. We can also confirm the necessity of segment merging and feature extraction steps as they drastically improve the results.

In order to evaluate the classes accuracy, we compute the confusion matrix for our baseline with hierarchical clustering presented in Table IV. An example of clustering results for the first couple of images is presented in Figure 6. We can see that all the primary change classes have been clustered with high precision, although further analysis might be needed to identify their sub-classes.

IV. CONCLUSION

In this paper, we have presented an end-to-end approach for the detection and clustering of non-trivial changes between two bi-temporal high resolution satellite images. The presented approach is based on change detection with joint convolutional AEs and object-based change clustering using 3D graph-based segmentation and feature extraction with convolutional AE. Our approach has shown promising results on the experimental dataset and outperformed concurrent approaches.

TABLE IV
CONFUSION MATRIX FOR THE PROPOSED BASELINE WITH HIERARCHICAL CLUSTERING.

		2004-2005			
		Predicted			
True		**cl 1**	**cl 2**	**cl 3**	**cl 4**
	cl 1	**0.91**	0.01	0.06	0.02
	cl 2	0.02	**0.81**	0.15	0.02
	cl 3	0	0.12	**0.83**	0.04
	cl 4	0.02	0.06	0.17	**0.75**

		2006-2008		
		Predicted		
True		**cl 1**	**cl 2**	**cl 3**
	cl 1	**1**	0	0
	cl 2	0.25	**0.75**	0
	cl 3	0.27	0	**0.73**

In future works, we plan on adapting the proposed approach for a full time series, as the temporal constraint will help us to distinguish more classes.

REFERENCES

[1] R. Caye Daudt, B. Le Saux, A. Boulch, and Y. Gousseau, "Urban change detection for multispectral earth observation using convolutional neural networks," *CoRR*, vol. abs/1810.08468, 2018.

[2] T. Lei, Q. Zhang, D. Xue, T. Chen, H. Meng, and A. K. Nandi, "End-to-end change detection using a symmetric fully convolutional network for landslide mapping," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2019, pp. 3027–3031.

[3] S. Basu, S. Ganguly, S. Mukhopadhyay, R. DiBiano, M. Karki, and R. Nemani, "Deepsat: a learning framework for satellite imagery," 11 2015, pp. 1–10.

[4] P. Du, S. Liu, P. Gamba, K. Tan, and J. Xia, "Fusion of difference images for change detection over urban areas," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 5, no. 4, pp. 1076–1086, Aug 2012.

[5] K. Tan, X. Jin, A. Plaza, X. Wang, L. Xiao, and P. Du, "Automatic change detection in high-resolution remote sensing images by using a multiple classifier system and spectral-spatial features," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 9, no. 8, pp. 3439–3451, Aug 2016.

[6] D. Peng and Y. Zhang, "Object-based change detection from satellite imagery by segmentation optimization and multi-features fusion," *International Journal of Remote Sensing*, vol. 38, pp. 3886–3905, 07 2017.

[7] Y. Xu, S. Xiang, C. Huo, and C. Pan, "Change detection based on auto-encoder model for VHR images," *Proc SPIE*, vol. 8919, pp. 02–, 10 2013.

[8] M. El Hajj, A. Bégué, B. Lafrance, O. Hagolle, G. Dedieu, and M. Rumeau, "Relative radiometric normalization and atmospheric correction of a SPOT 5 time series," *Sensors*, vol. 8, no. 4, pp. 2774–2791, 2008.

[9] F. Kruse, A. Lefkoff, J. Boardman, K. Heidebrecht, A. Shapiro, P. Barloon, and A. Goetz, "The spectral image processing system (SIPS)-interactive visualization and analysis of imaging spectrometer data," *Remote Sensing of Environment*, vol. 44, pp. 145–163, 07 1993.

[10] E.Kalinicheva, J.Sublime, and M.Trocan, "Change detection in satellite images using reconstruction errors of joint autoencoders, *Springer-Verlag Lecture Notes in Computer Science (LNCS)*, to appear.

[11] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 1, pp. 62–66, Jan 1979.

[12] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," *International Journal of Computer Vision*, vol. 59, no. 2, pp. 167–181, Sep 2004.

[13] B. Dezső, R. Giachetta, I. László, and I. Fekete, "Experimental study on graph-based image segmentation methods in the classification of satellite images," 05 2019.

[14] J. Cohen, "A coefficient of agreement for nominal scales," *Educational and Psychological Measurement*, vol. 20, no. 1, pp. 37–46, 1960.

[15] X. Guo, L. Gao, X. Liu, and J. Yin, "Improved deep embedded clustering with local structure preservation," in *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, 2017, pp. 1753–1759.