Code-Aligned Autoencoders for Unsupervised Change Detection in Multimodal Remote Sensing Images

Luigi T. Luppino, Mads A. Hansen, Michael Kampffmeyer,
Filippo M. Bianchi, Gabriele Moser, Robert Jenssen, and Stian Normann Anfinsen

Abstract—Image translation with convolutional autoencoders has recently been used as an approach to multimodal change detection in bitemporal satellite images. A main challenge is the alignment of the code spaces by reducing the contribution of change pixels to the learning of the translation function. Many existing approaches train the networks by exploiting supervised information of the change areas, which, however, is not always available. We propose to extract relational pixel information captured by domain-specific affinity matrices at the input and use this to enforce alignment of the code spaces and reduce the impact of change pixels on the learning objective. A change prior is derived in an unsupervised fashion from pixel pair affinities that are comparable across domains. To achieve code space alignment we enforce that pixel with similar affinity relations in the input domains should be correlated also in code space. We demonstrate the utility of this procedure in combination with cycle consistency. The proposed approach are compared with state-of-the-art deep learning algorithms. Experiments conducted on four real datasets show the effectiveness of our methodology.

Index Terms—unsupervised change detection, multimodal image analysis, heterogeneous data, image regression, affinity matrix, deep learning, aligned autoencoder

I. INTRODUCTION

CHANGE detection (CD) methods in remote sensing aim at identifying changes happening on the Earth by comparing two or more images acquired at different times [1]. Multitemporal analyses with satellite data include land use mapping of urban and agricultural areas [2], [3], and monitoring of large scale changes such as deforestation [4], lake and glacier reduction [5], [6], urbanisation [7], etc. Bitemporal applications mainly concerned with the detection and assessment of natural disasters and sudden events, like earthquakes [8], floods [9], forest fires [10], and so forth.

Traditional CD methods rely on homogeneous data, namely a set of images acquired by the same sensor, under the same geometry, seasonal conditions, and recording settings. However, these restrictions are too strong for many practical examples. First of all, the satellite revisit period sets the upper limit to the temporal resolution when monitoring long-term trends, and the lower limit to the response time when assessing the damages of sudden events. Moreover, even when two images are collected with the same configurations, they might be not homogeneous because of other factors, for example light conditions for optical data or humidity and precipitation for synthetic aperture radar (SAR).

Heterogeneous CD methods overcome these limitations, but at the cost of having to handle more complicated issues; Heterogeneous data imply different domains, diverse statistical distributions and inconsistent class signatures across the two images, especially when different sensors are involved, which makes a direct comparison infeasible [11]. These problems have been tackled by use of many different techniques: copula theory [1], marginal densities transformations [12], evidence theory [13], [14], graph theory [15], manifold learning [16], kernelised or deep canonical correlation analysis [10], [17], [18], dictionary learning [19], scale-invariant local descriptors [20], [21], superpixel segmentation [22], clustering [23], minimum energy [24], multidimensional scaling [25], nonlinear regression [26], [9], and deep learning (especially autoencoders) [27], [28], [29], [30], [31], [32].

A common solution in heterogeneous CD is to apply highly nonlinear transformations to transfer the data from one domain to the other and vice versa [30], [33], [34]. Alternatively, all the data are mapped to a common domain where they can be compared [12], [27], [28], [32]. Nonetheless, this crucial step often requires iterative fine-tuning of the transformation functions starting from unreliable preliminary results, e.g. random initialisation [28], [32] and clustering [30], or from manually selected training samples [1], [10], [16] that are not always available.

One contemporary way to map data across two domains is image-to-image (I2I) translation using a conditional generative adversarial network (cGAN) [35], which was extended by enforcing cyclic consistency in the cycleGAN architecture [36]. These approaches have inspired many recent heterogeneous CD methods [33], [34], [37]. A notable difference between the cGAN and the cycleGAN is that training of the former requires paired images that contain the same objects imaged with different styles or sensor modes, whereas the cycleGAN does not. Paired I2I translation can only be applied in heterogeneous CD if change pixels are censored, as these will otherwise distort the training process and promote a transformation between different objects.

When generative adversarial frameworks are used in heterogeneous CD, the translated (or cyclically translated) images take the role as fake or generated data, and the network is

L.T. Luppino, M.A. Hansen, M. Kampffmeyer, R. Jenssen and S.N. Anfinsen are with the Machine Learning Group, Department of Physics and Technology, UiT The Arctic University of Norway, e-mail: luigi.t.luppino@uit.no.
F.M. Bianchi is with NORCE Norwegian Research Center, Norway.
G. Moser is with DITEN Department, University of Genoa, Italy.
Manuscript received -; revised -.

trained to make them indistinguishable from true images from the relevant domain. The cGAN and cycleGAN may succeed to align the distributions of translated data and true data, but they are also seen to suffer from inherent drawbacks: They rely on large training sets, the iterative training of generator and discriminator must be judiciously balanced, training is prone to mode collapse, and reasonable values of the hyperparameters can be difficult to find due to oscillating and unstable behaviour of the loss function. We therefore seek alternative training strategies to the adversarial ones.

In this work, we propose a simple unsupervised, heterogeneous CD method, inspired by the paradigm of I2I translation. The idea is to align the code layers of two autoencoders and treat them as a common latent space, so that the output of one encoder can be the input of both decoders, leading in one case to reconstruction of data in their original domain, and in the other case to their transformation into the other domain. Local information extracted directly from the input images is exploited to drive the code alignment in an unsupervised manner. Specifically, affinity matrices of the training patches are computed and compared, and the extracted information is used to ensure that pixel pairs that are similar in both input domains also have a high correlation in the common latent space. The implementation of this principle is inspired by the deep kernelised autoencoder of Kampffmeyer et al. [38], [39], where the inner product between the codes produced by two datapoints is forced to match their precomputed affinity.

To summarise, the contributions of this work are the following:

- We propose a simple, yet effective loss term, able to align the latent spaces of two autoencoders in an unsupervised manner.
- We implement a deep neural network for heterogeneous CD that incorporates this loss term.
- The well-documented TensorFlow 2.0 framework that we provide can be easily used for the development of other CD methods and for direct comparison with ours. Source code is made available at https://github.com/llu025/Heterogeneous_CD.

The remainder of this paper is organised as follows: The core ideas and the main contribution are presented in Sec. II; Experiments were conducted on four different real datasets, and Sec. III shows the results of the proposed approach against several state-of-the-art methods; Sec. IV concludes the paper.

II. METHODOLOGY

Assume that we have two different sensors (or sensor modes) whose single-pixel measurements lie in the domains $\mathcal{X}$ and $\mathcal{Y}$. These could be e.g. $\mathbb{R}_{\geq 0}$ (nonnegative real numbers) for a single-channel SAR sensor, $\overline{\mathbb{R}}_{\geq 0}^{C}$ for a multispectral radiometer with C bands, or $\mathbb{C}_{\succeq 0}^{C \times C}$ for a polarimetric SAR sensor with C polarisations that records a complex and semipositive definite covariance matrix for each pixel.

Further assume that these sensors are scanning the same geographical region at separate times and we obtain an image $\mathcal{I}_{\mathcal{X}} \in \mathcal{X}^{H \times W}$ recorded at time t_1 and an image $\mathcal{I}_{\mathcal{Y}} \in \mathcal{Y}^{H \times W}$ recorded at $t_2 > t_1$. The images and their domains have common dimensions, the shared height H and width W, which are obtained after coregistration and resampling. They will in general have different numbers of channels, denoted as $|\mathcal{X}|$ and $|\mathcal{Y}|$. The two images can be thought of as realisations of stochastic processes that generate data tensors from domain $\mathcal{X}$ and $\mathcal{Y}$.

An underlying assumption is that a limited part of the image has changed between t_1 and t_2. The final goal is to detect all changes in the scene. However, given the heterogeneity of $\mathcal{X}$ and $\mathcal{Y}$, direct comparison is meaningless, if not unfeasible, without any preprocessing step. Let $\boldsymbol{X} \in \mathcal{X}^{h \times w}$ and $\boldsymbol{Y} \in \mathcal{Y}^{h \times w}$ be data tensors holding size $h \times w$ patches of the full images $\mathcal{I}_{\mathcal{X}}$ and $\mathcal{I}_{\mathcal{Y}}$. We are interested in implementing the two transformations: $\hat{\boldsymbol{Y}} = F(\boldsymbol{X})$ and $\hat{\boldsymbol{X}} = G(\boldsymbol{Y})$, defined as $F : \mathcal{X}^{h \times w} \to \mathcal{Y}^{h \times w}$ and $G : \mathcal{Y}^{h \times w} \to \mathcal{X}^{h \times w}$, to map data between the image domains. In this way, the input images can be transferred to the opposite domain, and the changes can be detected by computing the difference image as the weighted average:

$$\boldsymbol{\Delta} = W_{\mathcal{X}} \cdot d^{\mathcal{X}}(\boldsymbol{X}, \hat{\boldsymbol{X}}) + W_{\mathcal{Y}} \cdot d^{\mathcal{Y}}(\boldsymbol{Y}, \hat{\boldsymbol{Y}}), \qquad (1)$$

where $d^{\mathcal{X}}(\cdot, \cdot)$ and $d^{\mathcal{Y}}(\cdot, \cdot)$ are sensor-specific distances, chosen according to the statistical distribution of the data, which operate pixel-wise. The generic weights $W_{\mathcal{X}}$ and $W_{\mathcal{X}}$ can be used to balance the contribution of the domain-specific distances. We may want to use $W_{\mathcal{X}} = 1/|\mathcal{X}|$ and $W_{\mathcal{Y}} = 1/|\mathcal{Y}|$ in order to remove undue influence of the number of channels if $d^{\mathcal{X}}$ and $d^{\mathcal{Y}}$ involve summations on the corresponding channels. Alternatively, it may be appropriate to compensate for different noise levels of the sensors that affect the magnitude of the distances, for instance by boosting the contribution of optical data with respect to highly speckled radar data. The weights can be set heuristically or according to empirical optimisation and theoretical considerations. We prefer to use L_2 distances to limit the computational cost.

To implement $F(\boldsymbol{X})$ and $G(\boldsymbol{Y})$, we use a framework that consists of two autoencoders, each associated with one of the two image domains $\mathcal{X}$ and $\mathcal{Y}$ (We will from now suppress the superscripting with image patch dimensions $h \times w$). Specifically, they consist of two encoder-decoder pairs implemented as deep neural networks: the encoder $E_{\mathcal{X}}(\boldsymbol{X}) : \mathcal{X} \to \mathcal{Z}_{\mathcal{X}}$ and decoder $D_{\mathcal{X}}(\boldsymbol{Z}) : \mathcal{Z}_{\mathcal{X}} \to \mathcal{X}$; the encoder $E_{\mathcal{Y}}(\boldsymbol{Y}) : \mathcal{Y} \to \mathcal{Z}_{\mathcal{Y}}$ and decoder $D_{\mathcal{Y}}(\boldsymbol{Z}) : \mathcal{Z}_{\mathcal{Y}} \to \mathcal{X}$. Here, $\mathcal{Z}_{\mathcal{X}}$ and $\mathcal{Z}_{\mathcal{Y}}$ denote the code layer or latent space domains of the respective autoencoders. These are implemented with common dimensions, such that the code layer representation $\boldsymbol{Z}$ (also known as the *code*) can denote data tensors in both $\mathcal{Z}_{\mathcal{X}}$ and $\mathcal{Z}_{\mathcal{Y}}$. When we need to specify which input space the codes originate from, they will be written as $\boldsymbol{Z}^{\mathcal{X}}$ and $\boldsymbol{Z}^{\mathcal{Y}}$.

When trained separately and under the appropriate regularisation, the autoencoders will learn to encode their inputs and reconstruct them with high fidelity in output. Without any external forcing, the distributions of the codes in $\mathcal{Z}_{\mathcal{X}}$ and $\mathcal{Z}_{\mathcal{Y}}$ will in general not be close (see Fig. 1a for a visual example). However, we will introduce loss terms that enforce their alignment, both in distribution and in the location of land
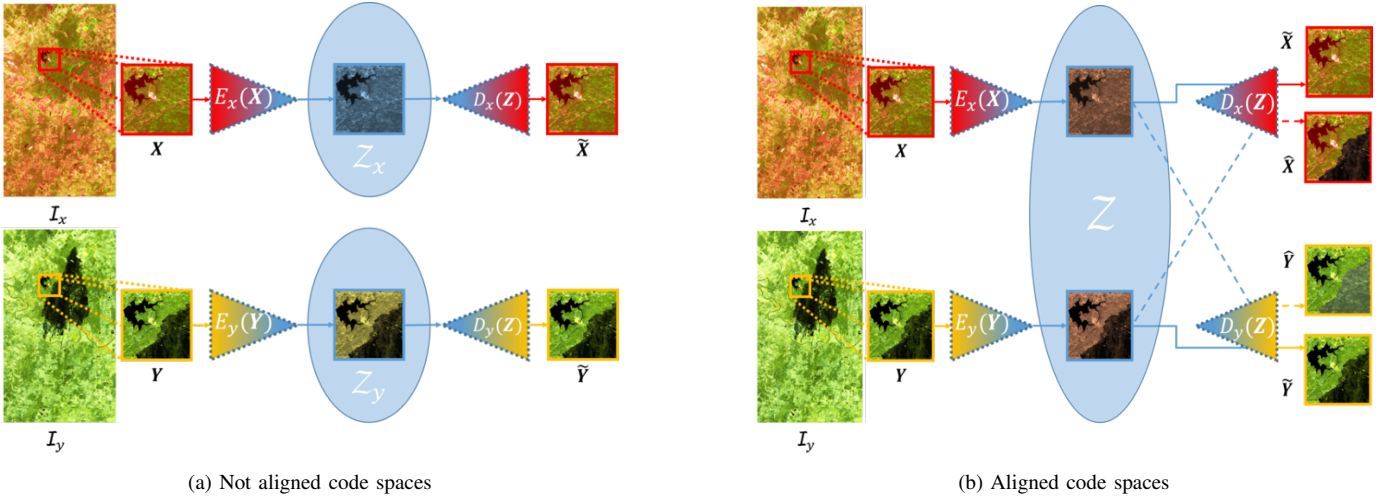
(a) Not aligned code spaces

(b) Aligned code spaces

Fig. 1: Two autoencoders without (a) and with (b) code space alignment.

covers within the distributions[1]. If the code distributions in $\mathcal{Z}_{\mathcal{X}}$ and $\mathcal{Z}_{\mathcal{Y}}$ align successfully, the encoders can be cascaded with the adjacent decoders to map the latent domain codes back to their original domains, or with the opposite decoders to map data across domains, leading to the sought transformations:

$$
\begin{aligned}
\hat{\boldsymbol{Y}} &= F(\boldsymbol{X}) = D_{\mathcal{Y}}(\boldsymbol{Z}^{\mathcal{X}}) = D_{\mathcal{Y}}\left(E_{\mathcal{X}}\left(\boldsymbol{X}\right)\right), \\
\hat{\boldsymbol{X}} &= G(\boldsymbol{Y}) = D_{\mathcal{X}}(\boldsymbol{Z}^{\mathcal{Y}}) = D_{\mathcal{X}}\left(E_{\mathcal{Y}}\left(\boldsymbol{Y}\right)\right),
\end{aligned}
\quad (2)
$$

as depicted in Fig. 1b.

Autoencoders require regularisation in order to avoid learning an identity mapping. This is commonly implemented as sparsity constraints or compression at the code layer by dimensionality reduction, with the latter measure known as a bottleneck. In our implementation, we retain the image patch dimensions (h and w) throughout the hidden layers of the autoencoder and do not resort to bottlenecking, as this is seen to produce the best results. The additional constraints associated with code alignment and crossdomain mapping are seen to enforce the required regularisation.

In the following, we define the terms of the loss function $\mathcal{L}\left(\boldsymbol{\vartheta}\right)$. The loss function is minimised with respect to the parameters of the networks, $\boldsymbol{\vartheta}$, to train the two autoencoders with the goal of obtaining the desired $F(\boldsymbol{X})$ and $G(\boldsymbol{Y})$. In order to compare input patches and translated ones, a weighted distance between patches is defined. Let $\boldsymbol{A}$ and $\boldsymbol{B}$ be two equal-sized $h \times w$ patches, then $\delta(\boldsymbol{A}, \boldsymbol{B}|\boldsymbol{\pi})$ denotes a general weighted distance between patches, where $\boldsymbol{\pi}$ is a vector of weights, each associated with a pixel $i \in \{1, \ldots, n\}$ of the patches, with $n = h \cdot w$. In particular, $\delta(\boldsymbol{A}, \boldsymbol{B}|\boldsymbol{1}) = \delta(\boldsymbol{A}, \boldsymbol{B})$, being $\boldsymbol{1}$ a vector of ones. When the pixel measurements $\boldsymbol{a}_i \in \boldsymbol{A}$ and $\boldsymbol{b}_i \in \boldsymbol{B}$ are vectors, the mean squared L_2 norm can be used:

$$
\delta(\boldsymbol{A}, \boldsymbol{B}|\boldsymbol{\pi}) = \frac{1}{n} \sum_{i=1}^{n} \pi_i \|\boldsymbol{a}_i - \boldsymbol{b}_i\|_2^2. \quad (3)
$$

[1]Alignment in distribution is not sufficient, since the arrangement of land covers within the distributions may have changed, for instance by mode swapping.

A. Reconstruction Loss

Consider two training patches of $h \times w$ pixels extracted at the same location from $\mathcal{I}_{\mathcal{X}}$ and $\mathcal{I}_{\mathcal{Y}}$. The first requirement for the autoencoders is to reproduce their input as faithfully as possible in output, which means that for the reconstructed image patches $\tilde{\boldsymbol{X}}$ and $\tilde{\boldsymbol{Y}}$,

$$
\begin{aligned}
\tilde{\boldsymbol{X}} &= D_{\mathcal{X}}\left(E_{\mathcal{X}}\left(\boldsymbol{X}\right)\right) \simeq \boldsymbol{X} \\
\tilde{\boldsymbol{Y}} &= D_{\mathcal{Y}}\left(E_{\mathcal{Y}}\left(\boldsymbol{Y}\right)\right) \simeq \boldsymbol{Y}
\end{aligned}
\quad (4)
$$

must hold true. We introduce the mean squared error between the desired and the predicted output as the reconstruction loss term:

$$
\mathcal{L}_{\mathrm{r}}(\boldsymbol{\vartheta}) = \mathbb{E}_{\boldsymbol{X}}\left[\delta(\tilde{\boldsymbol{X}}, \boldsymbol{X})\right] + \mathbb{E}_{\boldsymbol{Y}}\left[\delta(\tilde{\boldsymbol{Y}}, \boldsymbol{Y})\right]. \quad (5)
$$

B. Cycle-consistency Loss

Cycle-consistency implies that data transformed from $\mathcal{X}$ to $\mathcal{Y}$ and back to $\mathcal{X}$ should match exactly the input data we started from. The same applies to the transformations from $\mathcal{Y}$ to $\mathcal{X}$ and back. If $F(\boldsymbol{X})$ and $G(\boldsymbol{Y})$ are close to be perfectly adapted, it must hold true that

$$
\begin{aligned}
\dot{\boldsymbol{X}} &= G(\hat{\boldsymbol{Y}}) = G\left(F(\boldsymbol{X})\right) \simeq \boldsymbol{X}, \\
\dot{\boldsymbol{Y}} &= F(\hat{\boldsymbol{X}}) = F\left(G(\boldsymbol{Y})\right) \simeq \boldsymbol{Y},
\end{aligned}
\quad (6)
$$

where $\dot{\boldsymbol{X}} = G(\hat{\boldsymbol{Y}})$ and $\dot{\boldsymbol{Y}} = F(\hat{\boldsymbol{X}})$ indicate the data cyclically transformed to the original domains. Hence, we define the cycle-consistency loss term as:

$$
\mathcal{L}_{\mathrm{c}}(\boldsymbol{\vartheta}) = \mathbb{E}_{\boldsymbol{X}}\left[\delta(\dot{\boldsymbol{X}}, \boldsymbol{X})\right] + \mathbb{E}_{\boldsymbol{Y}}\left[\delta(\dot{\boldsymbol{Y}}, \boldsymbol{Y})\right]. \quad (7)
$$

We note that cycle-consistency, like reconstruction, can be evaluated with unpaired data, since $\tilde{\boldsymbol{X}}$ and $\dot{\boldsymbol{X}}$ are computed from $\boldsymbol{X}$ while $\tilde{\boldsymbol{Y}}$ and $\dot{\boldsymbol{Y}}$ are computed from $\boldsymbol{Y}$.

C. Weighted Translation Loss

For those pixels not affected by changes, we require

$$
\begin{aligned}
\hat{\boldsymbol{Y}} &= F(\boldsymbol{X}) \simeq \boldsymbol{Y} \\
\hat{\boldsymbol{X}} &= G(\boldsymbol{Y}) \simeq \boldsymbol{X}.
\end{aligned}
\quad (8)
$$

From the opposite perspective, pixels that are likely to be changed shall not fulfil these same requirements. Thus, the weighted translation loss term is defined as follows:

$$\mathcal{L}_{\text{t}}(\boldsymbol{\vartheta}) = \mathbb{E}_{\boldsymbol{X},\boldsymbol{Y}}\left[\delta(\hat{\boldsymbol{X}}, \boldsymbol{X}|\boldsymbol{\pi})\right] + \mathbb{E}_{\boldsymbol{X},\boldsymbol{Y}}\left[\delta(\hat{\boldsymbol{Y}}, \boldsymbol{Y}|\boldsymbol{\pi})\right] , \quad (9)$$

where the contribution to the translation loss of the pixels is weighted by the prior $\boldsymbol{\pi}$, whose elements $\{\pi_i\}_{i=1}^n$ can be interpreted as the probability of pixel $i \in \{1, \ldots, n\}$ not being changed. The π_i for the entire image are stored in a matrix $\boldsymbol{\Pi} \in [0,1]^{H \times W}$, from which the patch corresponding to $\boldsymbol{X}$ and $\boldsymbol{Y}$ is extracted and flattened into the vector $\boldsymbol{\pi}$. These probabilities are not available at the beginning of training, so all entries of $\boldsymbol{\Pi}$ are initialised as 0. After several training epochs, a preliminary evaluation of the difference image $\boldsymbol{\Delta}$ is computed and scaled to fall into the range $[0,1]$, so that the prior can be updated as $\boldsymbol{\Pi} = 1 - \boldsymbol{\Delta}$. In this way, pixels associated with a large $\boldsymbol{\Delta}$ entry are penalised by a small weight, whereas the opposite happens to pixels more likely to be unchanged. The $\boldsymbol{\Pi}$ is updated iteratively at a rate that we can tune to accommodate both performance and computational cost. This form of self-supervision paradigm has already proven robust in other tasks such as deep clustering [40] and deep image recovery [41].

The translation loss must be evaluated with paired data, since $\hat{\boldsymbol{X}}$ is computed from $\boldsymbol{Y}$ and compared with $\boldsymbol{X}$, while $\hat{\boldsymbol{Y}}$ is computed from $\boldsymbol{X}$ and compared with $\boldsymbol{Y}$. The code correlation loss, presented in the next section, also requires paired data.

D. Code Correlation Loss

The main contribution of this work lies in the way the codes are aligned. It therefore rests on the design and definition of the specific loss term associated with code alignment, referred to as the code correlation loss.

The distances in the input spaces between all pixel pairs (i, j) in the co-located training patches are computed as $d_{i,j}^{\mathcal{X}} = d^{\mathcal{X}}(\boldsymbol{x}_i, \boldsymbol{x}_j)$ and $d_{i,j}^{\mathcal{Y}} = d^{\mathcal{Y}}(\boldsymbol{y}_i, \boldsymbol{y}_j)$ for $i, j \in \{1, \ldots, n\}$, where $\boldsymbol{x}_i$ and $\boldsymbol{y}_j$ denote the feature vectors of pixel $i \in \boldsymbol{X}$ and pixel $j \in \boldsymbol{Y}$, respectively. The appropriate choice of distance measure depends on the underlying data distribution, but should also consider complexity. The hypothesis of normality for imagery acquired by optical sensors is commonly assumed [42], [43]. Concerning SAR intensity data, a logarithmic transformation is sufficient to bring it to near-Gaussianity [23], [31]. This qualifies the use of the computationally efficient Euclidean distance for both these data sources.

Once computed, the distances between all pixel pairs can be converted to the affinities

$$A_{i,j}^{\ell} = \exp\left\{-\frac{\left(d_{i,j}^{\ell}\right)^2}{\sigma_{\ell}^2}\right\} \in (0,1], \quad i, j \in \{1, \ldots, n\}. \tag{10}$$

Here, $A_{i,j}^{\ell}$ are the entries of the affinity matrix $\boldsymbol{A}^{\ell} \in \mathbb{R}^{n \times n}$ for a given patch and modality $\ell \in \{\mathcal{X}, \mathcal{Y}\}$, and σ_{ℓ} is the kernel width, which must be automatically determined. Our choice is to set it equal to the average distance to the kth

nearest neighbour for all data points in the patch of modality ℓ, with $k = \frac{3}{4}n$. This heuristic, which can be traced back to [44], captures the scale of local affinities within the patch and is robust with respect to outliers. Other common approaches to determine the kernel width, such as the Silverman's rule of thumb [45], were discarded because they have not proven themselves as effective.

At this point, one can consider the rows

$$A_i^{\mathcal{X}} = \left[A_{i,1}^{\mathcal{X}}, \ldots, A_{i,n}^{\mathcal{X}}\right] \text{ and } A_j^{\mathcal{Y}} = \left[A_{j,1}^{\mathcal{Y}}, \ldots, A_{j,n}^{\mathcal{Y}}\right]$$

as representations of pixel i from patch $\boldsymbol{X}$ and pixel j from patch $\boldsymbol{Y}$, respectively, in a new affinity space with n features. Moreover, we can define a novel crossmodal distance between these pixels as

$$D_{i,j} = \frac{1}{\sqrt{n}}\|A_i^{\mathcal{X}} - A_j^{\mathcal{Y}}\|_2 \in [0,1], \ i, j \in \{1, \ldots, n\}, \quad (11)$$

noting that since the affinities are normalised to the range $[0,1]$, then so is $D_{i,j}$. This crossmodal distance allows to compare data across the two domains directly from their input space features. It further allows us to distinguish pixels that have consistent relations to other pixels in both domains from those that do not. This information can be interpreted in terms of probability of change.

The crossmodal input space distances $D_{i,j}$ for $i, j \in \{1, \ldots, n\}$ are stored in $\boldsymbol{D}$. We next want to make sure that these are maintained in the code layer. We do this by defining similarities $S_{ij} = 1 - D_{ij}$ and forcing them to be as similar as possible to correlations between the codes of corresponding pixels. Let $\boldsymbol{z}_i^{\mathcal{X}}$ and $\boldsymbol{z}_j^{\mathcal{Y}}$ denote the entry of code patch $\boldsymbol{Z}^{\mathcal{X}}$ corresponding to pixel i and the entry of code patch $\boldsymbol{Z}^{\mathcal{Y}}$ corresponding to pixel j, respectively. In mathematical terms, we enforce that

$$R_{i,j} \triangleq \frac{\left(\boldsymbol{z}_i^{\mathcal{X}}\right)^T \boldsymbol{z}_j^{\mathcal{Y}} + |\mathcal{Z}|}{2\,|\mathcal{Z}|} \simeq S_{i,j}, \quad i, j \in \{1, \ldots, n\}, \quad (12)$$

where the $S_{i,j}$ are elements of $\boldsymbol{S} = 1 - \boldsymbol{D}$. The normalisation of the codes, $\boldsymbol{z}_i^{\mathcal{X}}, \boldsymbol{z}_j^{\mathcal{Y}} \in [-1, 1]^{|\mathcal{Z}|}$, and their dimensionality $|\mathcal{Z}|$ is such that the code correlations $R_{i,j}$ falls in the range $[0, 1]$. Note that the elements on the diagonal of $\boldsymbol{S}$ represent the similarity between $\boldsymbol{x}_i$ and $\boldsymbol{y}_i$, that are not identical, so $S_{i,i}$ can be different from 1. Also observe that $\boldsymbol{S}$ is not symmetric, because the similarity between $\boldsymbol{x}_i$ and $\boldsymbol{y}_j$ is not necessarily the same as between $\boldsymbol{x}_j$ and $\boldsymbol{y}_i$.

Based on the above definitions and considerations, the code correlation loss term is defined as

$$\mathcal{L}_z(\boldsymbol{\vartheta}) = \mathbb{E}_{\boldsymbol{X},\boldsymbol{Y}}\left[\delta(\boldsymbol{R}, \boldsymbol{S})\right], \tag{13}$$

where the code correlation matrix $\boldsymbol{R}$ stores the $R_{i,j}$ from the left-hand side of Eq. (12). Note that only encoder parameters are adjusted with this loss term.

E. Total Loss Function

Finally, the loss function minimised in this framework is the following weighted sum:

$$\mathcal{L}(\boldsymbol{\vartheta}) = \lambda_{\text{r}}\,\mathcal{L}_{\text{r}}(\boldsymbol{\vartheta}) + \lambda_{\text{c}}\,\mathcal{L}_{\text{c}}(\boldsymbol{\vartheta}) + \lambda_{\text{t}}\,\mathcal{L}_{\text{t}}(\boldsymbol{\vartheta}) + \lambda_z\,\mathcal{L}_z(\boldsymbol{\vartheta}), \quad (14)$$

where the weights λ_r, λ_c, λ_t and λ_z are used to balance the loss terms and their impact on the optimisation result. Together, the cycle-consistency and the code correlation let us achieve the sought alignment, while at the same time the other two terms keep focus on a correct reconstruction and transformation of the input.

After the training and the computation of $\boldsymbol{\Delta}$, the CD workflow includes an optional step and a mandatory step. The former consists of spatial filtering of $\boldsymbol{\Delta}$ to reduce errors, based on the simple idea that spurious changed (unchanged) pixels surrounded by unchanged (changed) ones are most likely outliers that have been erroneously classified. For our method we selected the Gaussian filtering presented in [46], which uses spatial context to regularise $\boldsymbol{\Delta}$. The last step of a CD pipeline is to obtain the actual change map by thresholding $\boldsymbol{\Delta}$, and so all the pixels whose value is below the threshold are considered unchanged, vice versa for those with a larger value. The optimal threshold can be found by visual inspection or automatically by exploiting an algorithm such as [47], [48], [49]. We opted for the classical Otsu's method [50].

III. RESULTS

A. Implementation details

For the proposed framework we deploy fully convolutional neural networks designed as follows: Conv($3 \times 3 \times 100$)–ReLU–Conv($3\times3\times100$)–ReLU–Conv($3\times3\times C$)–Tanh. Conv($3\times3\times C$) indicates a convolutional layer with C filters of size 3×3, being $C = 3$ for the encoders, $C = |\mathcal{X}|$ for $D_\mathcal{X}$ and $C = |\mathcal{Y}|$ for $D_\mathcal{Y}$. All the layers are non-strided and we apply padding to preserve the input size. Leaky-ReLU [51] with slope of $\beta = 0.3$ for negative arguments is used. Tanh indicates the hyperbolic tangent [51], which normalises data between -1 and 1, as this has shown to speed up convergence [52]. Dropout [53] with a 20% rate is applied. A low number of features in the latent space allows to achieve the sought alignment more easily, whereas the number of layers and filters has been set to find a balance between flexibility of the network representations and the limited trainability of the networks, due to a small amount of training data. Concerning the latter, at every epoch 10 batches containing 10 random patches of 100×100 pixels are extracted and randomly augmented (90 degrees rotations and upside-down flips). As specified, the code correlation loss term $\mathcal{L}_z$ requires computation of a size $N \times N$ crossmodal distance matrix $\boldsymbol{D}$ when the training patch is $h \times w$. Due to memory constraints, only the inner 20×20 pixels of the training patches have been used to compute $\boldsymbol{D}$. For normalisation of the matrix $\boldsymbol{D}$ between 0 and 1, the framework responded better when applying contrast stretching between the empirical batch minimum and maximum values of $\boldsymbol{D}$. The four λ values controlling the weighted sum of $\mathcal{L}$ were all set to 1.

The Adam optimiser [54] was selected to perform the minimisation of $\mathcal{L}$ for 100 epochs with a learning rate of 10^{-4}, which experienced a stair-cased exponential decay with rate 0.96. Actually, we found it beneficial to reduce the learning rate associated with $\mathcal{L}_z$ more aggressively with rate 0.9. This was implemented because it turned out most beneficial to correlate the code spaces at the beginning, when the autoencoder just started to learn a meaningful representation of the latent spaces and a reasonable transformation of the data. After some updates of $\boldsymbol{\Pi}$, $\mathcal{L}_z$ was experienced to function more as a regulariser, whereas the translation loss $\mathcal{L}_t$ came more into play. These updates were made every 25 epochs, so at epoch 25, 50, and 75.

B. Evaluation criteria

The performance of the proposed approach is measured in terms of two metrics. The overall accuracy, $OA \in [0, 1]$, is the ratio between correctly classified pixels and the total amount of pixels. Cohen's kappa coefficient, $\kappa \in [-1, 1]$, indicates the agreement between two classifiers [55]. $\kappa = 1$ means total agreement, $\kappa = -1$ means total disagreement, $\kappa = 0$ means no correlation (random guess). When comparing against a ground truth dataset, Cohen's kappa is expressed as

$$\kappa = \frac{p_o - p_e}{1 - p_e}. \tag{15}$$

Here, p_o stands for the observed agreement between predictions and labels, i.e. the OA, while p_e is the probability of random agreement, which is estimated from the observed true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) as:

$$p_e = \left(\frac{\text{TP} + \text{FP}}{N} \cdot \frac{\text{FN} + \text{TN}}{N} \right) \\ + \left(\frac{\text{TP} + \text{FN}}{N} \cdot \frac{\text{FP} + \text{TN}}{N} \right). \tag{16}$$

In general, a high κ implies a high OA, but not vice versa. In any case, the papers presenting state-of-the-art methods do not always report both, so we compare algorithm performance dataset by dataset in terms of the available metrics.

C. Methods compared

We will in the following present four datasets that are used to test the proposed method and reference algorithms. On the first two datasets, the proposed method is compared to four similar deep learning approaches. The first two are the conditional adversarial network (CAN) of Niu et al. [33] and the symmetric convolutional coupling network (SCCN) of Liu et al. [28], which represent seminal work on unsupervised multimodal change detection with convolutional neural networks. The final two are are the ACE-Net and the X-Net recently proposed by the current authors in [37]. To be aware of the characteristics of the training strategies employed by these methods, it should be noted that the CAN and the ACE-Net apply adversarial training, the ACE-Net and the SCCN exploit code alignment, while the ACE-Net and the X-Net use similar weighted image-to-image translation schemes as the proposed method. The final two datasets have been used extensively by others in testing of methods whose source code we do not have access to. For these datasets we compare our results with the performance reported in Zhang et al. [27] for post-classification comparison (PCC) and a deep learning model based on stacked denoising autoencoders (SDAE). We also compare with several methods proposed by Touati et al.,

namely a method that obtains its result my filtering a textural gradient-based similarity map (TGSM) [56], a method using energy-based encoding of nonlocal pairwise pixel interactions (EENPPI) [24], a method based on modality invariant multidimensional scaling (MIMDS) [25], and a Markov model for multimodal change detection (M3CD) [57]. Finally, we compare with results obtained with the manifold learning-based statistical model (MLSM) of Prendes et al. [16], [58].
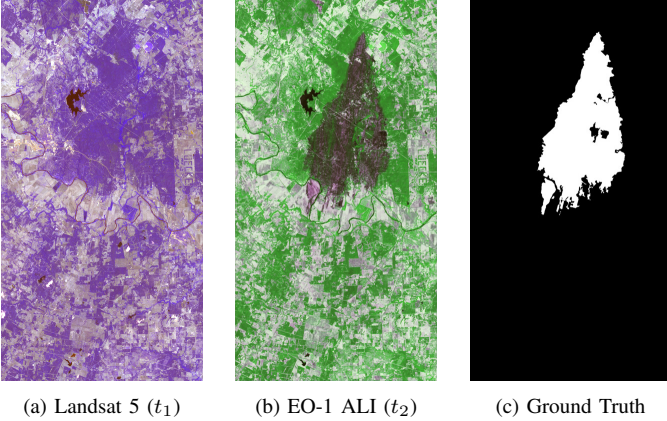
D. First dataset: Forest fire in Texas

(a) Landsat 5 (t_1)   (b) EO-1 ALI (t_2)   (c) Ground Truth

Fig. 2: Forest fire in Texas: Landsat 5 ($t1$), (b) EO-1 ALI ($t2$), (c) ground truth. RGB false color composites are shown for both images.

A Landsat 5 Thematic Mapper (TM) multispectral image (Fig. 2a) was acquired before a forest fire that took place in Bastrop County, Texas, during September-October 2011[2]. An Earth Observing-1 Advanced Land Imager (EO-1 ALI) multispectral acquisition after the event completes the dataset (Fig. 2b)[1]. Both images are optical, with 1534×808 pixels, and 7 and 10 channels respectively. The ground truth of the event (see Fig. 2c) is provided by Volpi *et al.* [10].

Fig. 3 displays the results obtained on this dataset by the proposed framework as compared to the reference methods. As one can notice, the proposed network produces consistently higher accuracy than the competitors and also maintains a low variance. We also report that Volpi *et al.* [10] and Luppino *et al.* [9] achieved a κ of 0.65 and 0.91 respectively with respect to the same ground truth. Concerning the training times, their averages are listed in Table I. These are comparable because the computation of the affinity matrices is time-consuming, but the proposed method is implemented with relatively small networks and trained for fewer iterations.

TABLE I: Average training time of the five methods on the Texas dataset.

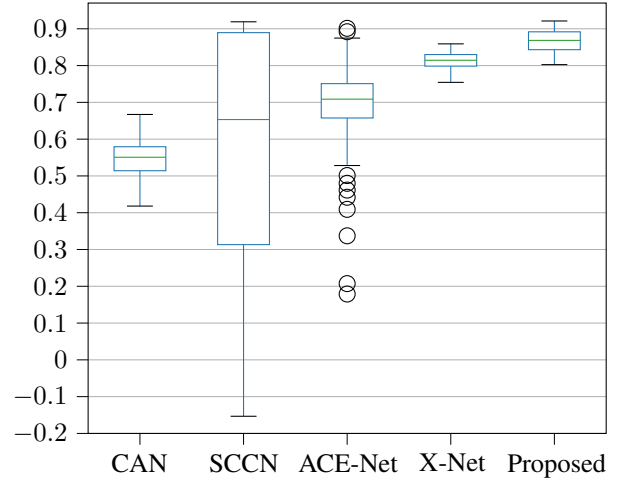| CAN	SCCN	ACE-Net	X-Net	Proposed
70 min	16 min	13 min	7 min	11 min

Fig. 3: κ obtained on the Texas dataset by the proposed approach and several state-of-the-art methods.

E. Second dataset: Flood in California

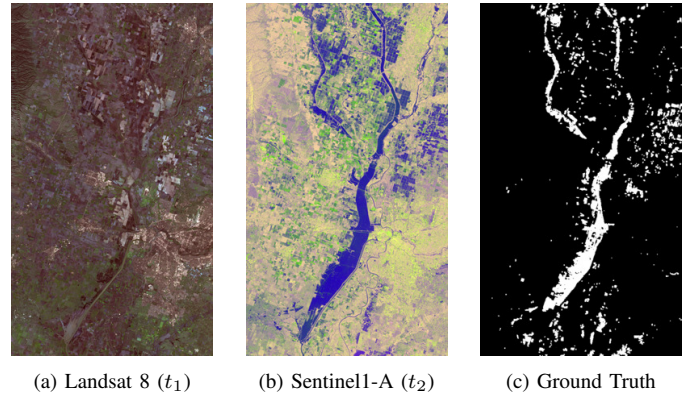(a) Landsat 8 (t_1)   (b) Sentinel1-A (t_2)   (c) Ground Truth

Fig. 4: Flood in California: Landsat 8 ($t1$), (b) Sentinel1-A ($t2$), (c) ground truth. RGB false color composites are shown for both images.

Fig. 4a shows the RGB channels of the Landsat 8 acquisition[1] covering Sacramento County, Yuba County and Sutter County, California, on 5 January 2017. In addition, the multispectral sensors mounted on Landsat 8 provides another 8 channels, going from deep blue to long-wave infrared. The same area was affected by a flood, as it can be noticed in Fig. 4b. This is a Sentinel-1A[3] acquisition, recorded in polarisations VV and VH on 18 February 2017 and augmented with the ratio between the two intensities as the third channel. The ground truth in Fig. 4c is provided by Luppino *et al.* [9]. Originally of 3500×2000 pixels, these images were resampled to 850×500 pixels as in [37] to compare the results.

The metrics obtained on this dataset are summarised in Fig. 5. Also in this case, the proposed framework outperforms the state-of-the-art counterparts, both in terms of high quality and low variance. For this dataset, $\kappa = 0.46$ was achieved in [9]. Table II contains the average training times on this dataset.

[2]Distributed by LP DAAC, http://lpdaac.usgs.gov

[3]Data processed by ESA, http://www.copernicus.eu/

TABLE II: Average training time of the five methods on the California dataset.

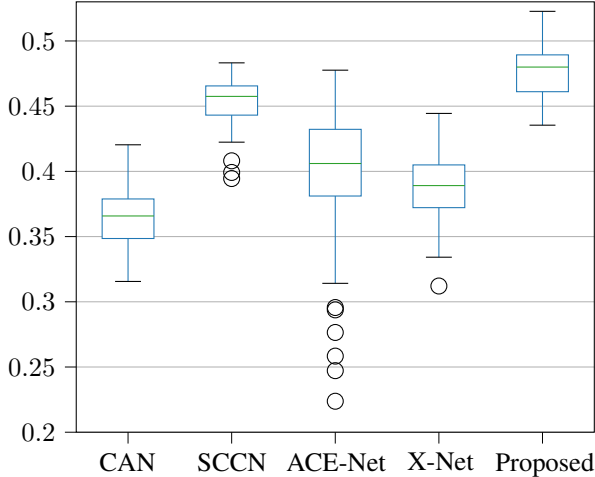| CAN	SCCN	ACE-Net	X-Net	Proposed
21 min	15 min	12 min	6 min	8 min

Fig. 5: κ obtained on the California dataset by the proposed approach and several state-of-the-art methods

Again, the proposed approach required a training time which is in line with the state-of-the-art algorithms.
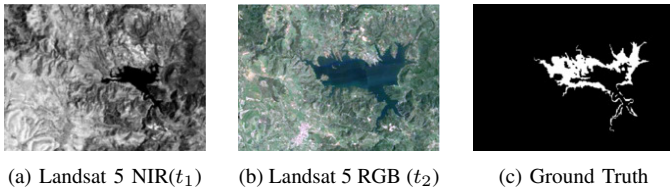
F. Third dataset: Lake overflow in Italy

(a) Landsat 5 NIR(t_1)   (b) Landsat 5 RGB (t_2)   (c) Ground Truth

Fig. 6: Lake overflow in Italy: Landsat 5 Near InfraRed (NIR) band ($t1$), (b) Landsat 5 red, green, and blue (RGB) bands ($t2$), (c) ground truth.

The next two datasets were provided by Touati *et al.* [57]. In Fig. 6a and Fig. 6b are two Landsat 5 images of 412×300 pixels: the first is the Near InfraRed (NIR) band of an image acquired in September 1995, the second represents the red, green, and blue (RGB) bands sensed on the same area in July 1996. These images were recorded before and after a lake overflow in Italy, whose profile is highlighted as ground truth in Fig. 6c. Table III presents the average overall accuracy for several methods. For the proposed method, the standard deviation is provided as well, and one may see that the results are very stable and close to the state-of-the-art. The small amount of data in terms of the number of pixels does not in general favour deep learning approaches, and the relative performance could potentially change with larger training samples. In this respect, Zhang *et al.* [27] proposed a method that seems to be an exception, as this deep learning approach produces the best

TABLE III: Average accuracy of several methods on the lake overflow dataset. Best on top, proposed method in bold.

Lake overflow dataset	OA
SDAE [27]	0.975
M3CD [57]	0.964
MIMDS [25]	0.942
Proposed	**0.922 ± 0.007**
PCC [27]	0.882

performance on this dataset. However, it must be pointed out that, unlike us, they adapt their architectures to the dataset, which is infeasible in a completely unsupervised setting. The average training time for the proposed framework on this dataset was a few seconds below 7 minutes.

G. Fourth dataset: Construction site in France

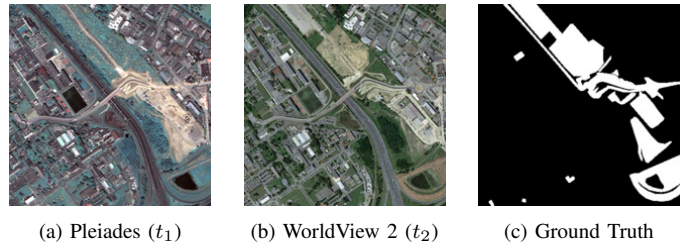(a) Pleiades (t_1)   (b) WorldView 2 (t_2)   (c) Ground Truth

Fig. 7: Constructions in France: Pleiades ($t1$), (b) WorldView 2 ($t2$), (c) ground truth.

The last dataset includes two RGB images captured by Pleiades (Fig. 7a) and WorldView 2 (Fig. 7b), showing the work progress of road constructions in Toulouse, France, during May 2012 and July 2013. The ground truth in Fig. 7c depicts such progress. For computational reasons, the images were reduced from 2000×2000 pixels to 500×500 as in [57], leading to an average training time of 7 minutes. The average accuracy obtained by several methods on this dataset is listed in Table IV. Again, the accuracy of the proposed method comes with a standard deviation, and also in this case it is very stable and close to the state-of-the-art.

Finally, in Fig. 8 we present a visual example of the transformations obtained with the proposed method on the datasets used in this section. As it can be seen, the data

TABLE IV: Average accuracy of several methods on the constructions dataset. Best on top, proposed method in bold.

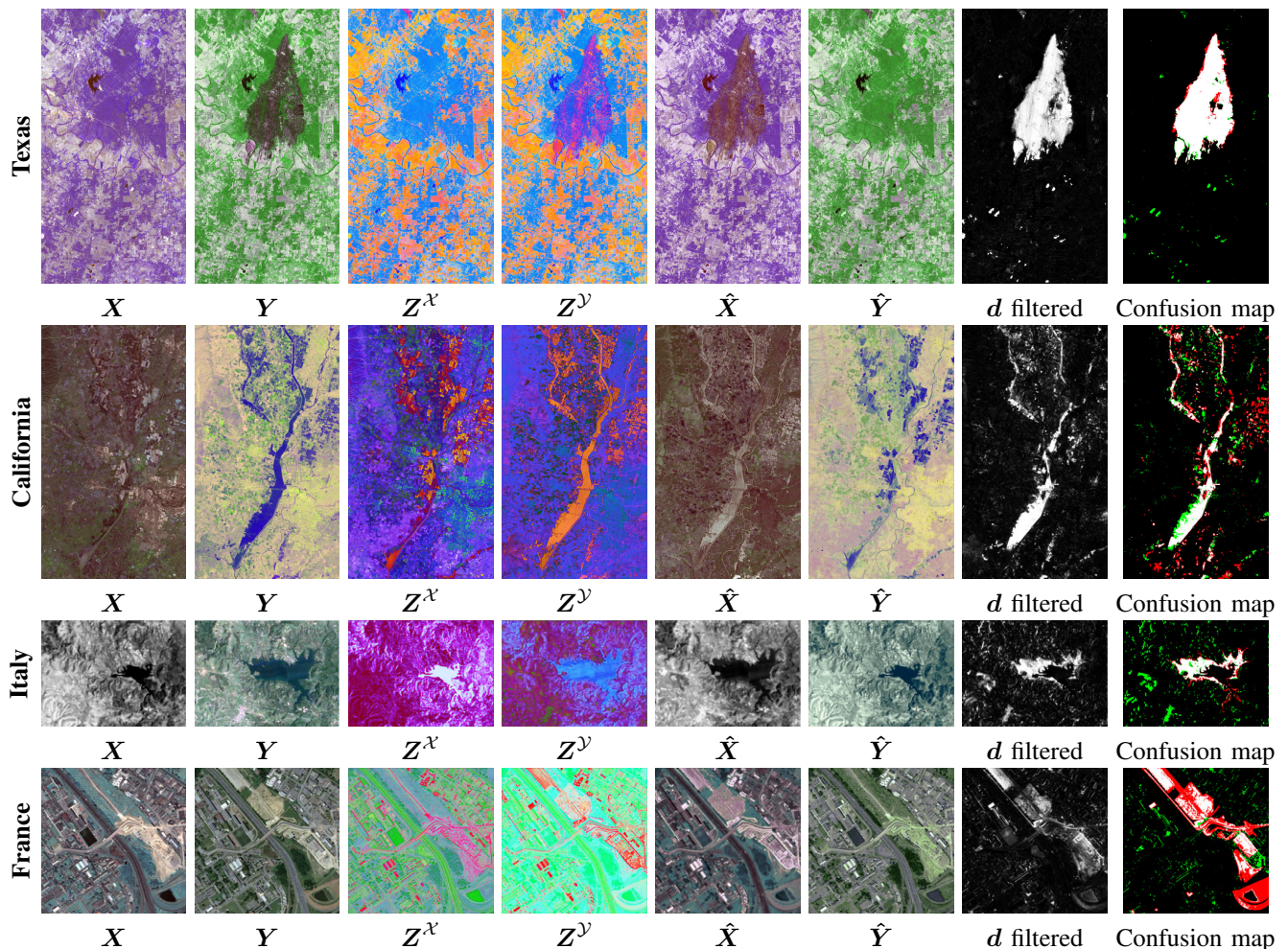| Constructions dataset	OA
MIMDS [25]	0.877
TGSM [56]	0.870
M3CD [57]	0.862
Proposed	**0.859 ± 0.003**
EENPPI [24]	0.853
MLSM [16]	0.844

Fig. 8: Examples of final results, organized in one row for each dataset. Col. 1: input image X; Col. 2: input image Y; Col. 3: transformations of X into the code space $Z^{\mathcal{X}} = E_{\mathcal{X}}(X)$; Col. 4: transformations of Y into the code space $Z^{\mathcal{Y}} = E_{\mathcal{Y}}(Y)$; Col. 5 transformations $\hat{Y} = F(X)$; Col. 6: transformations $\hat{X} = G(Y)$; Col. 7: d filtered; Col. 8: Confusion map (TP: white; TN: black; FP: green; FN: red) (g)

from one input domain are transformed into the other in a meaningful way, and the resemblance between the styles of the fake images and the original images is clear. In the two last datasets, one could speculate that the low amount of data and features (few pixels consisting of few channels) did not allow to achieve a proper alignments of the code spaces. This endorses the choice to compute d as a weighted sum of the difference images in the input spaces rather than just the difference image in the latent space, although it still remains a valid option.

IV. Conclusions

In this work, we presented a novel unsupervised methodology to align the code spaces of two autoencoders based on affinity information extracted from the input data. In particular, this is part of a heterogeneous CD framework that allows to achieve this latent space entanglement even when the input images contain changes, whose misleading contribution to the training is considerably reduced. The method proved to perform consistently on par with or better than the state-of-

the-art across four different datasets. Its performance worsen when handling a limited amount of features in input, especially when only one channel is available in one of the images, implying a regression from one variable to many, which is an ill-posed problem. On the other hand, it deals properly with multispectral and multipolarisation images, by being able to map data appropriately across domains in a meaningful manner.

V. Acknowledgement

References

[1] G. Mercier, G. Moser, and S. B. Serpico, "Conditional copulas for change detection in heterogeneous remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 46, no. 5, pp. 1428–1441, May 2008.
[2] X. Li and A. G.-O. Yeh, "Analyzing spatial restructuring of land use patterns in a fast growing region using remote sensing and GIS," *Landscape and Urban planning*, vol. 69, no. 4, pp. 335–354, 2004.

[3] M. Herold, J. Scepan, and K. C. Clarke, "The use of remote sensing and landscape metrics to describe structures and changes in urban land uses," *Environment and Planning A*, vol. 34, no. 8, pp. 1443–1458, 2002.

[4] S. H. Khan, X. He, F. Porikli, and M. Bennamoun, "Forest change detection in incomplete satellite images with deep neural networks," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 9, pp. 5407–5423, 2017.

[5] A. A. Alesheikh, A. Ghorbanali, and N. Nouri, "Coastline change detection using remote sensing," *Int. J. Environmental Sci. Tech.*, vol. 4, no. 1, pp. 61–66, 2007.

[6] E. Berthier, Y. Arnaud, R. Kumar, S. Ahmad, P. Wagnon, and P. Chevallier, "Remote sensing estimates of glacier mass balances in the Himachal Pradesh (Western Himalaya, India)," *Remote Sens. Environ.*, vol. 108, no. 3, pp. 327–338, 2007.

[7] P. Griffiths, P. Hostert, O. Gruebner, and S. van der Linden, "Mapping megacity growth with multi-sensor data," *Remote Sens. Environ.*, vol. 114, no. 2, pp. 426–439, 2010.

[8] D. Brunner, G. Lemoine, and L. Bruzzone, "Earthquake damage assessment of buildings using VHR optical and SAR imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 48, no. 5, pp. 2403–2420, 2010.

[9] L. T. Luppino, F. M. Bianchi, G. Moser, and S. N. Anfinsen, "Unsupervised image regression for heterogeneous change detection," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 12, pp. 9960–9975, 2019.

[10] M. Volpi, G. Camps-Valls, and D. Tuia, "Spectral alignment of multitemporal cross-sensor images with automated kernel canonical correlation analysis," *ISPRS J. Photogram. Remote Sens.*, vol. 107, pp. 50–63, 2015.

[11] L. T. Luppino, F. M. Bianchi, G. Moser, and S. N. Anfinsen, "Remote sensing image regression for heterogeneous change detection," in *Proc. IEEE Int. Workshop Mach. Learn. Signal Process. (MLSP)*, 2018, pp. 1–6.

[12] B. Storvik, G. Storvik, and R. Fjortoft, "On the combination of multisensor data using meta-Gaussian distributions," *IEEE Trans. Geosci. Remote Sens.*, vol. 47, no. 7, pp. 2372–2379, 2009.

[13] Z.-G. Liu, J. Dezert, G. Mercier, and Q. Pan, "Dynamic evidential reasoning for change detection in remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 50, no. 5, pp. 1955–1967, 2011.

[14] Z.-G. Liu, G. Mercier, J. Dezert, and Q. Pan, "Change detection in heterogeneous remote sensing images based on multidimensional evidential reasoning," *IEEE Geosci. Remote Sens. Lett.*, vol. 11, no. 1, pp. 168–172, 2014.

[15] D. Tuia, J. Munoz-Mari, L. Gomez-Chova, and J. Malo, "Graph matching for adaptation in remote sensing," *IEEE Trans. Geosci. Remote Sens.*, vol. 51, no. 1, pp. 329–341, Jan 2013.

[16] J. Prendes, M. Chabert, F. Pascal, A. Giros, and J.-Y. Tourneret, "A new multivariate statistical model for change detection in images acquired by homogeneous and heterogeneous sensors," *IEEE Trans. Image Process.*, vol. 24, no. 3, pp. 799–812, 2015.

[17] Y. Zhou, H. Liu, D. Li, H. Cao, J. Yang, and Z. Li, "Cross-sensor image change detection based on deep canonically correlated autoencoders," in *Proc. Int. Conf. Artif. Intell. Commun. Netw.*, 2019, pp. 251–257.

[18] J. Yang, Y. Zhou, Y. Cao, and L. Feng, "Heterogeneous image change detection using deep canonical correlation analysis," in *Proc. Int. Conf. Pattern Recogn. (ICPR)*, 2018, pp. 2917–2922.

[19] M. Gong, P. Zhang, L. Su, and J. Liu, "Coupled dictionary learning for change detection from multisource data," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 12, pp. 7077–7091, 2016.

[20] G. Liu, J. Delon, Y. Gousseau, and F. Tupin, "Unsupervised change detection between multi-sensor high resolution satellite images," in *Proc. Eur. Signal Process. Conf. (EUSIPCO)*. IEEE, 2016, pp. 2435–2439.

[21] G. Liu, Y. Gousseau, and F. Tupin, "A contrario comparison of local descriptors for change detection in very high spatial resolution satellite images of urban areas," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 6, pp. 3904–3918, 2019.

[22] D. Marcos, R. Hamid, and D. Tuia, "Geospatial correspondences for multimodal registration," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recogn. (CVPR)*, 2016, pp. 5091–5100.

[23] L. T. Luppino, S. N. Anfinsen, G. Moser, R. Jenssen, F. M. Bianchi, S. Serpico, and G. Mercier, "A clustering approach to heterogeneous change detection," in *Proc. Scand. Conf. Image Anal. (SCIA)*, 2017, pp. 181–192.

[24] R. Touati and M. Mignotte, "An energy-based model encoding nonlocal pairwise pixel interactions for multisensor change detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 2, pp. 1046–1058, 2018.

[25] R. Touati, M. Mignotte, and M. Dahmane, "Change detection in heterogeneous remote sensing images based on an imaging modality-invariant MDS representation," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, 2018, pp. 3998–4002.

[26] Z. Liu, G. Li, G. Mercier, Y. He, and Q. Pan, "Change detection in heterogenous remote sensing images via homogeneous pixel transformation," *IEEE Trans. Image Process.*, vol. 27, no. 4, pp. 1822–1834, 2018.

[27] P. Zhang, M. Gong, L. Su, J. Liu, and Z. Li, "Change detection based on deep feature representation and mapping transformation for multispatial-resolution remote sensing images," *ISPRS J. Photogram. Remote Sens.*, vol. 116, pp. 24–41, 06 2016.

[28] J. Liu, M. Gong, K. Qin, and P. Zhang, "A deep convolutional coupling network for change detection based on heterogeneous optical and radar images," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 3, pp. 545–559, 2016.

[29] W. Zhao, Z. Wang, M. Gong, and J. Liu, "Discriminative feature learning for unsupervised change detection in heterogeneous images based on a coupled neural network," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 12, pp. 7066–7080, 2017.

[30] L. Su, M. Gong, P. Zhang, M. Zhang, J. Liu, and H. Yang, "Deep learning and mapping based ternary change detection for information unbalanced images," *Pattern Recognition*, vol. 66, pp. 213–228, 2017.

[31] T. Zhan, M. Gong, X. Jiang, and S. Li, "Log-based transformation feature learning for change detection in heterogeneous images," *IEEE Geosci. Remote Sens. Lett.*, vol. 15, no. 9, pp. 1352–1356, 2018.

[32] T. Zhan, M. Gong, J. Liu, and P. Zhang, "Iterative feature mapping network for detecting multiple changes in multi-source remote sensing images," *ISPRS J. Photogram. Remote Sens.*, vol. 146, pp. 38–51, 2018.

[33] X. Niu, M. Gong, T. Zhan, and Y. Yang, "A conditional adversarial network for change detection in heterogeneous images," *IEEE Geoscience and Remote Sensing Letters*, vol. 16, no. 1, pp. 45–49, 2018.

[34] M. Gong, X. Niu, T. Zhan, and M. Zhang, "A coupling translation network for change detection in heterogeneous images," *Int. J. Remote Sens.*, vol. 40, no. 9, pp. 3647–3672, 2019.

[35] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proc. IEEE Comput. Soc. Int. Conf. Comput. Vis. Pattern Recogn. (CVPR)*, July 2017.

[36] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proc. IEEE Int. Conf. Comp. Vis. (ICCV)*, 2017, pp. 2223–2232.

[37] L. T. Luppino, M. Kampffmeyer, F. M. Bianchi, G. Moser, S. B. Serpico, R. Jenssen, and S. N. Anfinsen, "Deep image translation with an affinity-based change prior for unsupervised multimodal change detection," `arXiv:2001.04271 [cs.LG]`, 2020.

[38] M. Kampffmeyer, S. Løkse, F. M. Bianchi, R. Jenssen, and L. Livi, "The deep kernelized autoencoder," *Applied Soft Computing*, vol. 71, pp. 816–825, 2018.

[39] F. M. Bianchi, L. Livi, K. Ø. Mikalsen, M. Kampffmeyer, and R. Jenssen, "Learning representations of multivariate time series with missing data," *Pattern Recognition*, vol. 96, no. 106973, 2019.

[40] M. Caron, P. Bojanowski, A. Joulin, and M. Douze, "Deep clustering for unsupervised learning of visual features," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 132–149.

[41] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Deep image prior," in *Proc. IEEE Comp. Soc. Conf. Comput. Vis. Pattern Recogn. (CVPR)*, 2018, pp. 9446–9454.

[42] F. Bovolo and L. Bruzzone, "A theoretical framework for unsupervised change detection based on change vector analysis in the polar domain," *IEEE Trans. Geosci. Remote Sens.*, vol. 45, no. 1, pp. 218–236, 2007.

[43] ——, "The time variable in data fusion: A change detection perspective," *IEEE Geosci. Remote Sens. Mag.*, vol. 3, no. 3, pp. 8–26, 2015.

[44] Y. Mack and M. Rosenblatt, "Multivariate k-nearest neighbor density estimates," *J. Multivar. Anal.*, vol. 9, no. 1, pp. 1–15, 1979.

[45] M. P. Wand and M. C. Jones, *Kernel Smoothing*, ser. Monographs on Statistics and Applied Probability. Chapman & Hall/CRC, 1995, vol. 60.

[46] P. Krähenbühl and V. Koltun, "Efficient inference in fully connected CRFs with Gaussian edge potentials," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2011, pp. 109–117.

[47] J. N. Kapur, P. K. Sahoo, and A. K. Wong, "A new method for gray-level picture thresholding using the entropy of the histogram," *Computer Vision, Graphics, and Image Processing*, vol. 29, no. 3, pp. 273–285, 1985.

[48] A. G. Shanbhag, "Utilization of information measure as a means of image thresholding," *CVGIP: Graphical Models and Image Processing*, vol. 56, no. 5, pp. 414–419, 1994.

[49] J.-C. Yen, F.-J. Chang, and S. Chang, "A new criterion for automatic multilevel thresholding," *IEEE Trans. Image Process.*, vol. 4, no. 3, pp. 370–378, 1995.

[50] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Trans. Syst. Man Cybern.*, vol. 9, no. 1, pp. 62–66, 1979.

[51] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *Proc. Int. Conf. Mach. Learn. (ICML)*, vol. 30, no. 1, 2013, p. 3.

[52] Y. A. LeCun, L. Bottou, G. B. Orr, and K.-R. Müller, "Efficient backprop," in *Neural networks: Tricks of the trade*. Springer, 2012, pp. 9–48.

[53] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.

[54] S. J. Reddi, S. Kale, and S. Kumar, "On the convergence of ADAM and beyond," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2018.

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

[56] R. Touati, M. Mignotte, and M. Dahmane, "A new change detector in heterogeneous remote sensing imagery," in *Proc. IEEE Int. Conf. Image Process. Theory Tools Applic. (IPTA)*, 2017, pp. 1–6.

[57] ——, "Multimodal change detection in remote sensing images using an unsupervised pixel pairwise-based Markov random field model," *IEEE Trans. Image Proc.*, vol. 29, pp. 757–767, 2019.

[58] J. Prendes, "New statistical modeling of multi-sensor images with application to change detection," Ph.D. dissertation, Université Paris-Saclay, 2015.