Learning Disentangled Representations of Satellite Image Time Series

Eduardo H. Sanchez[1,2] (✉), Mathieu Serrurier[1,2], and Mathias Ortner[1]

[1] IRT Saint Exupéry, Toulouse, France
{eduardo.sanchez, mathias.ortner}@irt-saintexupery.com
[2] IRIT, Université Toulouse III - Paul Sabatier, Toulouse, France
mathieu.serrurier@irit.fr

Abstract. In this paper, we investigate how to learn a suitable representation of satellite image time series in an unsupervised manner by leveraging large amounts of unlabeled data. Additionally, we aim to disentangle the representation of time series into two representations: a shared representation that captures the common information between the images of a time series and an exclusive representation that contains the specific information of each image of the time series. To address these issues, we propose a model that combines a novel component called cross-domain autoencoders with the variational autoencoder (VAE) and generative adversarial network (GAN) methods. In order to learn disentangled representations of time series, our model learns the multimodal image-to-image translation task. We train our model using satellite image time series provided by the Sentinel-2 mission. Several experiments are carried out to evaluate the obtained representations. We show that these disentangled representations can be very useful to perform multiple tasks such as image classification, image retrieval, image segmentation and change detection.

Keywords: Unsupervised learning · Image-to-image translation · VAE · GAN · Disentangled representation · Satellite image time series.

1 Introduction

Deep learning has demonstrated impressive performance on a variety of tasks such as image classification, object detection, semantic segmentation, among others. Typically, these models create internal abstract representations from raw data in a supervised manner. Nevertheless, supervised learning is a limited approach since it requires large amounts of labeled data. It is not always possible to obtain labeled data since it requires time, effort and resources. As a consequence, semi-supervised or unsupervised algorithms have been developed to reduce the required number of labels. Unsupervised learning is intended to learn useful representations of data easily transferable for further usage. As using smart data representations is important, another desirable property of unsupervised methods is to perform dimensionality reduction while keeping the most important characteristics of data. Classical methods are principal component analysis

(PCA) or matrix factorization. For the same purpose, autoencoders learn to compress data into a low-dimensional representation and then, to uncompress that representation into the original data. An autoencoder variant is the variational autoencoder (VAE) introduced by Kingma and Welling [13] where the low-dimensional representation is constrained to follow a prior distribution. The VAE provides a way to extract a low-dimensional representation while learning the probability distribution of data. Other unsupervised methods of learning the probability data distribution have been recently proposed using generative models. A generative model of particular interest is generative adversarial networks (GANs) introduced by Goodfellow *et al.* [6, 7].

In this work, we present a model that combines the VAE and GAN methods in order to create a useful representation of satellite image time series in an unsupervised manner. To create these representations we propose to learn the image-to-image translation task introduced by Isola *et al.* [11] and Zhu *et al.* [21]. Given two images from a time series, we aim to translate one image into the other one. Since both images are acquired at different times, the model should learn the common information between these images as well as their differences to perform translation. We also aim to create a disentangled representation into a shared representation that captures the common information between the images of a time series and an exclusive representation that contains the specific information of each image. For instance, the common information accross time series could be useful to perform image classification while the knowledge about the specific information of each image could be useful for change detection.

Since we aim to generate any image of the time series from any of its images, we address the problem of multimodal generation, *i.e.* multiple output images can be generated from a single input image. For instance, an image containing harvested fields could be translated into an image containing growing crop fields, harvested fields or a combination of both.

Our approach is inspired by the BicycleGAN model introduced by Zhu *et al.* [22] to address multimodal generation and the model presented by Gonzalez-Garcia *et al.* [5] to address representation disentanglement.

In this work, the following contributions are made. First, we propose a model that combines the cross-domain autoencoder principle proposed by Gonzalez-Garcia *et al.* [5] under the GAN and VAE constraints to address representation disentanglement and multimodal generation. Differences with respect to models [5, 22] can be seen in Section 2. Our model is adapted to satellite image time series analysis using a simpler architecture (see Section 3). Second, we show that our model is capable to process a huge volume of high-dimensional data such as Sentinel-2 image time series in order to create feature representations (see Section 4). Third, our model generates a disentangled representation that isolates the common information of the entire time series and the exclusive information of each image. Our experiments suggest that these representations are useful to perform several tasks such as image classification, image retrieval, image segmentation and change detection by outperforming other state-of-the-art methods in some cases (see Sections 4.2, 4.3, 4.4, 4.5 and 4.6).

2 Related work

Variational autoencoder (VAE). In order to estimate the data distribution of a dataset, a common approach is to maximize the log-likelihood function given the samples of the dataset. A lower bound of the log-likelihood is introduced by Kingma and Welling [13]. To learn the data distribution, the authors propose to maximize the lower bound instead of the log-likelihood function which in some cases is intractable. The model is implemented using an autoencoder architecture and trained via a stochastic gradient descent method. It is capable to create a low-dimensional representation where relevant attributes of data are captured.
Generative adversarial networks (GANs). Due to its great success in many different domains, GANs [6, 7] have become one of the most important research topics. The GAN model can be thought of as a game between two players: the generator and the discriminator. In this setting, the generator aims to produce samples that look like drawn from the same distribution as the training samples. On the other hand, the discriminator receives samples to determine whether they are real (dataset samples) or fake (generated samples). The generator is trained to fool the discriminator by learning a mapping function from a latent space which follows a prior distribution to the data space. However, traditional GANs (DCGAN [18], LSGAN [16], WGAN [1], WGAN-GP [8], EBGAN [20], among others) does not provide a means to learn the inverse mapping from the data space to the latent space. To solve this problem, several models were proposed such us BiGAN [4] or VAE-GAN [14] which include an encoder from the data space to the latent space in the model. The data representation obtained in the latent space via the encoder can be used for other tasks as shown by Donahue *et al.* [4].
Image-to-image translation. It is one of the most popular applications using conditional GANs [17]. The image-to-image translation task consists of learning a mapping function between an input image domain and an output image domain. Impressive results have been achieved by the pix2pix [11] and cycleGAN [21] models. Nevertheless, most of these models are monomodal. That is, there is a unique output image for a given input image.
Multimodal image-to-image translation. One of the limitations of previous models is the lack of diversity of generated images. Certain models address this problem by combining the GAN and VAE methods. On the one hand, GANs are used to generate realistic images while VAE is used to provide diversity in the output domain. Recent work that deals with multimodal output is presented by Gonzalez-Garcia *et al.* [5], Zhu *et al.* [22], Huang *et al.* [10] and Ma *et al.* [15]. In particular, to be able to generate an entire time series from a single image, we adopt the principle of the BicycleGAN model proposed by Zhu *et al.* [22] where a low-dimensional latent vector represents the diversity of the output domain. Since the BicycleGAN model is mainly focused on image generation, the model architecture is not suitable for feature extraction. Instead, we propose a model capable to split the shared information across the time series and the exclusive information of each image that generates the diversity of the output domain.
Disentangled feature representation. Recent work is focused on learning dis-

entangled representations by isolating the factors of variation of high-dimensional data in an unsupervised manner. A disentangled representation can be very useful for several tasks that require knowledge of these factors of variation. Chen *et al.* [3] propose an objective function based on the maximization of the mutual information. Gonzalez-Garcia *et al.* [5] propose a model based on VAE-GAN image translators and a novel network component called cross-domain autoencoders. This model separates the representation of two image domains into three parts: the shared part which contains common information from both domains and the exclusive parts which only contain factors of variation that are specific to each domain.

In this paper, we propose a model that combines the cross-domain autoencoder component under the VAE and GAN constraints in order to analyze satellite image time series by creating a shared representation that captures the spatial information and an exclusive representation that captures the temporal information. While our method is inspired by the model proposed by Gonzalez-Garcia *et al.* [5], we would like to highlight some differences: a) The model [5] considers two image domains whose representation space can be split into two exclusive parts and a shared part. For instance, the authors use a colored MNIST dataset which can be split into: background color (exclusive part), digit color (exclusive part) and digit (shared part). In our case, we consider only a shared part which corresponds to spatial information at a given location on the Earth's surface and an exclusive part which is related to the acquisition time of the images; b) The model [5] performance is analyzed using simple datasets (colored MNIST, 3D cars and 3D chairs) while running the code provided by the authors to learn their model on Sentinel-2 data fails to converge generating unsatisfactory results as shown in the additional material (see Section 1); c) We use a simpler model architecture composed of 4 networks that implements the exclusive and shared representation encoder, the decoder and the discriminator functions while the model [5] uses 10 networks (2 encoders, 2 decoders, 4 discriminators and 2 GRL decoders) to achieve representation disentanglement which can be difficult to train simultaneously.

3 Method

Let X, Y be two images randomly sampled from a given time series T in a region C. Let $\mathcal{X}$ be the image domain where these images belong to and let $\mathcal{R}$ be the representation domain of these images. The representation domain $\mathcal{R}$ is divided into two subdomains $\mathcal{S}$ and $\mathcal{E}$, then $\mathcal{R} = [\mathcal{S}, \mathcal{E}]$. The subdomain $\mathcal{S}$ contains the common information between images X and Y and the subdomain $\mathcal{E}$ contains the particular information of each image. Since images X and Y belong to the same time series, their shared representations must be identical, *i.e.* $S_X = S_Y$. On the other hand, as images are acquired at different times, their exclusive representations E_X and E_Y correspond to the specific information of each image.

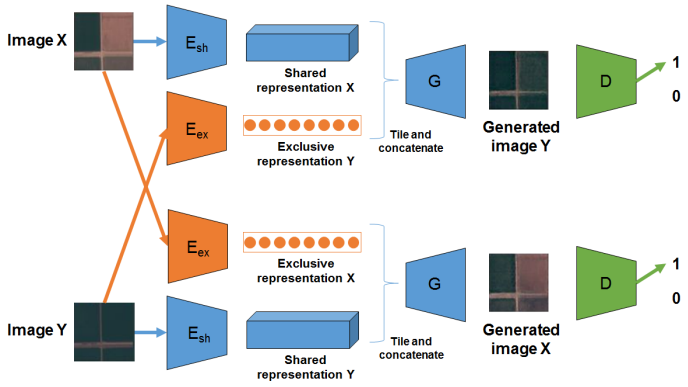


Fig. 1. Model overview. The model goal is to learn both image transitions: $X \rightarrow Y$ and $Y \rightarrow X$. Both images are passed through the network E_{sh} in order to extract their shared representations. Similarly, the network E_{ex} extracts the exclusive representations corresponding to images X and Y. In order to generate the image Y, the decoder network G takes the shared representation of image X and the exclusive representation of image Y. A similar procedure is performed to generate the image X. Finally, the discriminator D is used to evaluate the generated images.

We propose a model that learns the transition from X to Y as well as the inverse transition from Y to X. In order to accomplish this, an autoencoder-like architecture is used. In Figure 1, an overview of the model can be observed. Let $E_{sh} : \mathcal{X} \rightarrow \mathcal{S}$ be the shared representation encoder and $E_{ex} : \mathcal{X} \rightarrow \mathcal{E}$ be the exclusive representation encoder. To generate the image Y, the shared representation of X, *i.e.* $E_{sh}(X)$, and the exclusive representation of Y, *i.e.* $E_{ex}(Y)$ are computed. Then both representations are passed through the decoder function $G : \mathcal{R} \rightarrow \mathcal{X}$ which generates a reconstructed image $G(E_{sh}(X), E_{ex}(Y))$. A similar process is followed to reconstruct the image X. Then, these images are passed through a discriminator function $D : \mathcal{X} \rightarrow [0, 1]$ in order to evaluate the generated images.

The model functions E_{ex}, E_{sh}, G and D are represented by neural networks with parameters $\theta_{E_{ex}}$, $\theta_{E_{sh}}$ and θ_G and θ_D, respectively. The training procedure to learn these parameters is explained below.

3.1 Objective function

Similarly to Zhu *et al.* [22] and Gonzalez-Garcia *et al.* [5], our objective function is composed of several terms to obtain a disentangled representation.

Concerning the shared representation, images X and Y must have identical shared representations, *i.e.* $E_{sh}(X) = E_{sh}(Y)$. A simple solution is to minimize the L_1 distance between their shared representations as shown in Equation 1.

$$L_1^{sh} = \mathbb{E}_{X,Y \sim \mathcal{X}} \left[|E_{sh}(X) - E_{sh}(Y)| \right] \tag{1}$$

The exclusive representation must only contain the particular information that corresponds to each image. To enforce the disentanglement between shared and exclusive representations, we include a reconstruction loss in the objective function where the shared representations of X and Y are switched. The loss term corresponding to the reconstruction of image X is represented in Equation 2. Moreover, this loss term can be thought of as the reconstruction loss in the VAE model [13] which maximizes a lower bound of the log-likelihood function. As we enforce representation disentanglement, we minimize the Kullback-Leibler divergence between the data distribution and the generated distribution.

$$L_1^{X,Y} = \mathbb{E}_{X,Y\sim\mathcal{X}}\left[|X - G(E_{sh}(Y), E_{ex}(X))|\right] \tag{2}$$

On the other hand, the lower bound proposed in the VAE model constraints the feature representation to follow a prior distribution. In our model, we only force the exclusive representation to be distributed as a standard normal distribution $\mathcal{N}(0, I)$ in order to generate multiple outputs by sampling from this space during inference while keeping the shared representation constant. In contrast to Gonzalez-Garcia *et al.* [5] where a GAN approach is used to constraint the exclusive representation, a simpler and effective solution is to include a Kullback-Leibler divergence term between the distribution of the exclusive representation and the prior $\mathcal{N}(0, I)$. Assuming that the exclusive representation encoder $E_{ex}(X)$ is distributed as a normal distribution $\mathcal{N}(\mu_{E_{ex}(X)}, \sigma_{E_{ex}(X)})$, the Kullback-Leibler divergence can be written as follows,

$$L_{KL}^X = -\frac{1}{2}\mathbb{E}_{X\sim\mathcal{X}}\left[1 + \log(\sigma^2_{E_{ex}(X)}) - \mu^2_{E_{ex}(X)} - \sigma^2_{E_{ex}(X)}\right] \tag{3}$$

We include a LSGAN loss [16] in the objective function to encourage the model to generate realistic and diverse images thus improving the learned representations. The discriminator is trained to maximize the probability of assigning the correct label to real images and generated images while the generator is trained to fool the discriminator by classifying generated images as real, *i.e.* $D(G(E_{sh}(Y), E_{ex}(X))) \rightarrow 1$. The corresponding loss term for image X and its reconstructed version can be seen in Equation 4 where the discriminator maximizes this term while the generator minimizes it.

$$L_{GAN}^X = \mathbb{E}_{X\sim\mathcal{X}}\left[(D(X))^2\right] + \mathbb{E}_{X,Y\sim\mathcal{X}}\left[(1 - D(G(E_{sh}(Y), E_{ex}(X))))^2\right] \tag{4}$$

To summarize, the training procedure can be seen as a minimax game (Equation 5) where the objective function $\mathcal{L}$ is minimized by the generator functions of the model (E_{ex}, E_{sh}, G) while it is maximized by the discriminator D.

$$\begin{aligned}\min_{E_{ex},E_{sh},G}\max_{D}\ \mathcal{L} = L_{GAN}^X + L_{GAN}^Y + \lambda_{L_1}\left(L_1^{X,Y} + L_1^{Y,X}\right) \\ + \lambda_{L_{KL}}(L_{KL}^X + L_{KL}^Y) + \lambda_{L_1}^{sh} L_1^{sh}\end{aligned} \tag{5}$$

Where λ_{L_1}, $\lambda_{L_1}^{sh}$ and $\lambda_{L_{KL}}$ are constant coefficients to weight the loss terms.

3.2 Implementation details

Our model is architectured around four neural networks: the shared representation encoder, the exclusive representation encoder, the decoder and the discriminator. The architecture details are provided in the additional material section. To train our model, we use batches of 64 randomly selected image pairs of size $64 \times 64 \times 4$ from our satellite image time series dataset (see Section 4.1). Every network is trained from scratch by using randomly initialized weights as starting point. The learning rate is implemented as a staircase function which starts with an initial value of 0.0002 and decays every 50000 iterations. We use Adam optimizer to update the network weights using a $\beta = 0.5$ during 150000 iterations. Concerning the loss coefficients, we use the following values: $\lambda_{L_1} = 10$, $\lambda_{L_1}^{sh} = 0.5$ and $\lambda_{L_{KL}} = 0.01$ during training. The training algorithm was executed on a NVIDIA Tesla K80 GPU during 3 days to process 100GB of satellite image time series. The training procedure is summarized in Algorithm 1.

Algorithm 1 Training algorithm.

1: Random initialization of model parameters $(\theta_D^{(0)}, \theta_{E_{sh}}^{(0)}, \theta_{E_{ex}}^{(0)}, \theta_G^{(0)})$
2: **for** $k = 1$; $k = k + 1$; $k <$ number of iterations **do**
3: Sample a batch of m time series $\{T_s^{(1)}, ..., T_s^{(m)}\}$
4: Sample a batch of m image pairs $\{(X^{(1)}, Y^{(1)}), ..., (X^{(m)}, Y^{(m)})\}$ from $\{T_s^{(i)}\}$
5: Compute $\mathcal{L}^{(k)}(X^{(i)}, Y^{(i)}, \theta_D^{(k)}, \theta_{E_{sh}}^{(k)}, \theta_{E_{ex}}^{(k)}, \theta_G^{(k)})$

$$\begin{aligned}\mathcal{L}^{(k)} = \frac{1}{m}\sum_{i=1}^{m}\Bigg[&\Big(D(X^{(i)})\Big)^2 + \Big(1 - D(G(E_{sh}(Y^{(i)}), E_{ex}(X^{(i)})))\Big)^2 + \Big(D(Y^{(i)})\Big)^2 \\ &+ \Big(1 - D(G(E_{sh}(X^{(i)}), E_{ex}(Y^{(i)})))\Big)^2 + \lambda_{L_1}^{sh}\Big(|E_{sh}(X^{(i)}) - E_{sh}(Y^{(i)})|\Big) \\ &+ \lambda_{L_1}\Big(|X^{(i)} - G(E_{sh}(Y^{(i)}), E_{ex}(X^{(i)}))| + |Y^{(i)} - G(E_{sh}(X^{(i)}), E_{ex}(Y^{(i)}))|\Big) \\ &- \frac{1}{2}\lambda_{L_{KL}}\Big(2 + \log(\sigma^2_{E_{ex}(X^{(i)})}) - \mu^2_{E_{ex}(X^{(i)})} - \sigma^2_{E_{ex}(X^{(i)})} + \log(\sigma^2_{E_{ex}(Y^{(i)})}) \\ &- \mu^2_{E_{ex}(Y^{(i)})} - \sigma^2_{E_{ex}(Y^{(i)})}\Big)\Bigg]\end{aligned}$$

6: Update the parameters $\theta_{E_{sh}}^{(k+1)}$, $\theta_{E_{ex}}^{(k+1)}$ and $\theta_G^{(k+1)}$ by gradient descent of $\mathcal{L}^{(k)}$.
7: Update the parameters $\theta_D^{(k+1)}$ by gradient ascent of $\mathcal{L}^{(k)}$.
8: **end for**

4 Experiments

4.1 Sentinel-2

The Sentinel-2 mission is composed of a constellation of 2 satellites that orbit around the Earth providing an entire Earth coverage every 5 days. Both satellites acquire images at 13 spectral bands using different spatial resolutions. In

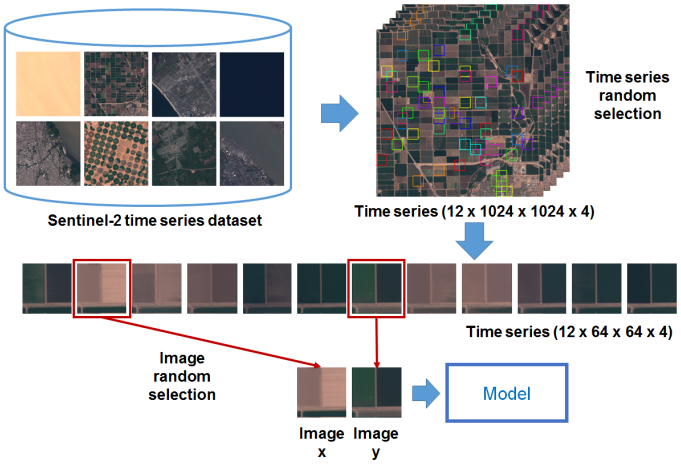


Fig. 2. Training data selection. A batch of smaller time series is randomly sampled from the dataset. At each iteration two images are randomly selected from each time series to be used as input for our model.

this paper, we use the RGBI bands[3] which correspond to bands at 10m spatial resolution. Bands are acquired after L-1C processing. In order to organize the data acquired by the mission, Earth surface is divided into square tiles of approximately 100 km on each side. One tile acquired at a particular time is referred to as a granule.

To create our training dataset, we selected 42 tiles containing several regions of interest such as the Amazon rainforest, the Dead Sea, the city of Los Angeles, the Great Sandy Desert, circular fields in Saudi Arabia, among others. The list of tiles is provided in the additional material. As explained by Kempeneers and Soille [12], many of the acquired granules might carry useless information. In our case, the availability of granules for a given tile depends on two factors: the cloud coverage and the image completeness. Therefore, we defined a threshold in order to avoid these kind of problems that affect Earth observation by setting a cloud coverage tolerance of 2% and completeness tolerance of 85%. For each tile, we extracted 12 granules from March 2016 to April 2018 keeping a regular time-step between granules. Then, we selected 25 non-overlapping patches of size 1024×1024 from the center of the tiles to reduce the effect of the satellite orbit view angle. Finally, our training dataset is composed of $42 \times 25 = 1050$ times series each of which is composed of 12 images of size $1024 \times 1024 \times 4$. The training dataset size is around 100GB. Similarly, we create a test dataset by selecting 6 different tiles whose size is around 14GB.

[3] Red (band 4), Green (band 3), Blue (band 2) and Near infrared (band 8) bands

In order to analyze the entire time series using smaller patches the following strategy is applied: a batch of time series composed of images of size $64 \times 64 \times 4$ is randomly sampled from the time series of images of size $1024 \times 1024 \times 4$. Since our model takes two images as input, at each iteration two images are randomly selected from the time series to be used as input for our model. Thus, the whole time series is learned as the training procedure progresses. Data sampling procedure is depicted in Figure 2.

To evaluate the model performance and the learned representations, we perform several supervised and unsupervised experiments on Sentinel-2 data as suggested by Theis [19]. We evaluate our model on: a) image-to-image translation to validate the representation disentanglement; b) image retrieval, image classification and image segmentation to validate the shared representation and c) change detection to analyze the exclusive representation. We also provide several examples of the experiment results in the additional material section.

As explained in Section 2, the model proposed by Gonzalez-Garcia *et al.* [5] fails to converge using Sentinel-2 data. As a consequence, it was not possible to evaluate the learned representations and compare the performance on the proposed tasks with respect to our method. Nevertheless, we compare our model with the BicycleGAN [22] and VAE [13] models and show that our model achieves better results at image classification, image retrieval and change detection.

4.2 Image-to-image translation

It seems natural to first test the model performance at image-to-image translation. We sample 9600 time series of size $12 \times 64 \times 64 \times 4$ to evaluate our model. It represents around 20k processed images of size $64 \times 64 \times 4$.

An example of image-to-image translation using our model can be observed in Figure 3. For instance, let us consider the image in the third row, fifth column. The shared representation is extracted from an image X which corresponds to growing crop fields while the exclusive representation is extracted from another image Y where fields have been harvested. Consequently, the generated image contains harvested fields which is defined by the exclusive representation of image Y. In general, generated images look realistic in both training and test datasets except for small details which are most likely due to the absence of skip connections between the encoders and generator.

We quantify the L_1 distance between generated images $G(E_{sh}(X), E_{ex}(Y))$ and images Y used to extract the exclusive representations. Results can be observed in Table 1 (first column). Pixel values in generated images and real images are in the range of $[-1, 1]$, thus a mean difference of 0.0155 indicates that the model performs well at image-to-image translation. The BicycleGAN model [22] achieves a slightly better result of 0.0136 which is probably due to the use of skip connections. However, our model is mainly focused on representation learning to perform downstream tasks and not on image generation.

A special image-to-image translation case is image autoencoding where the shared and exclusive representations are extracted from the same image. The L_1 distance between images X and autoencoded images $G(E_{sh}(X), E_{ex}(X))$ is

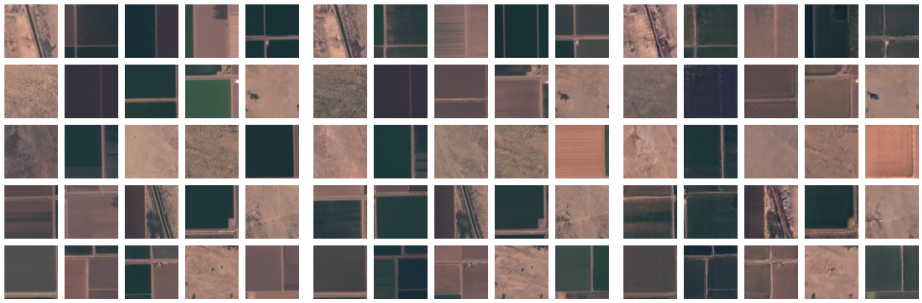


Fig. 3. Image translation performed on images of Brawley, California. (a) Images used to extract the shared representations; (b) Images used to extract the exclusive representations; (c) Generated images from the shared representation of (a) and the exclusive representation of (b). More examples are available in the additional material section.

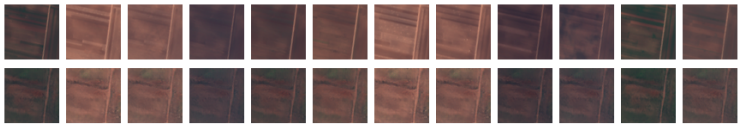


Fig. 4. Multimodal generation. The first row corresponds to a time series sampled from the test dataset. The second row corresponds to a time series where each image is generated by using the same shared representation and only modifying the exclusive representation.

computed for comparison purpose in Table 1 (second column). Lower values in terms of L_1 distance are obtained with respect to those of image-to-image translation. We provide the result obtained from the VAE [13] model as a baseline. Our model achieves a similar performance generating well-reconstructed images even if this case is not considered during training.

Finally, we perform times series reconstruction in order to show that the exclusive representation encodes the specific information of each image. An image is randomly selected from a time series to extract its shared representation. While keeping the shared representation constant and only modifying the exclusive representation, we reconstruct all the images of the original time series. Results in terms of L_1 distance between the original time series and the reconstructed one are shown in Table 1 (third column). As in the previous cases, the BicycleGAN [22] achieves a slightly better result of 0.0140 with respect to our model performance of 0.0184. An example of time series reconstruction using our model can be seen in Figure 4. Since the shared representation which represents the spatial location is constant, the experiment suggests that the exclusive representation controls the image information related to the acquisition time.

Method	Image translation	Image autoencoding	Time series reconstruction
VAE [13]	-	0.0086 ± 0.0300	-
BicycleGAN [22]	0.0136 ± 0.0538	0.0045 ± 0.0138	0.0140 ± 0.0503
Ours	0.0155 ± 0.0595	0.0085 ± 0.0318	0.0184 ± 0.0664

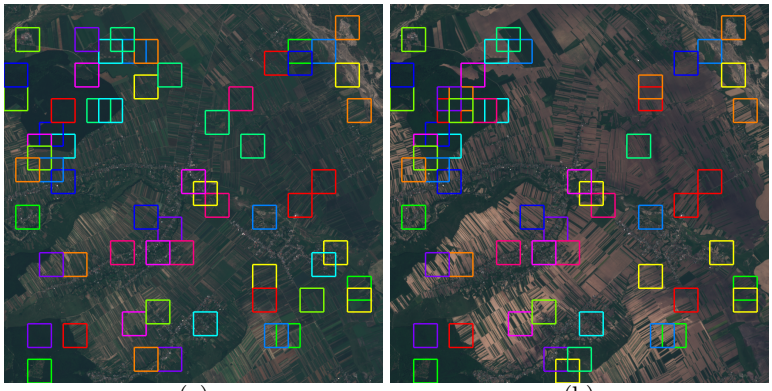
Table 1. Mean and standard deviation values in terms of the L_1 distance for image-to-image translation (first column), image autoencoding (second column) and time series reconstruction (third column).

4.3 Image retrieval

In this experiment, we want to evaluate whether the shared representation provides information about the geographical location of time series via image retrieval. Given an image patch from a granule acquired at time t_o, we would like to locate it in a granule acquired at time t_f. The procedure is the following: a time series of size $12 \times 1024 \times 1024 \times 4$ is randomly sampled from the dataset. Then, a batch of 64 image patches of size $64 \times 64 \times 4$ is randomly selected as shown in Figure 5a. The corresponding shared representations are extracted for each image of the batch. The main idea is to use the information provided by the shared representation to locate the image patches in every image of the time series. For each image of the time series, a sliding window of size $64 \times 64 \times 4$ is applied in order to explore the entire image. As the window slides, the shared representations are extracted and compared to those of the images to be retrieved. The nearest image in terms of L_1 distance is selected as the retrieved image at each image of the time series. In our experiment, 150 time series of size $12 \times 1024 \times 1024 \times 4$ are analyzed. It represents around 115k images of size $64 \times 64 \times 4$ to be retrieved and 110M images of size $64 \times 64 \times 4$ to be analyzed.

To illustrate the retrieval algorithm, let us consider an image of agricultural fields. We plot the image patches to be retrieved in Figure 5a and the retrieved image patches by the algorithm in Figure 5b. As can be seen, even if some changes have occurred, the algorithm is able to spatially locate most of the patches. In spite of the seasonal changes in the agricultural fields, the algorithm performs correctly since the image retrieval leverages the shared representation which contains common information of the time series. Results in terms of Recall@1 are displayed in Table 2 (last row). We obtain a high value in terms of Recall@1 even if it is not so close to 1. This result can be explained since the dataset contains several time series from the desert, forest and ocean tiles which could be notoriously difficult to retrieve even for humans. For instance, image retrieval performs better in urban scenarios since the city provides details that can be easily identified in contrast to agricultural fields where distinguishing textures can be confusing (see the additional material section).

As a baseline to compare to our method based on the shared representation, we use the raw pixels as feature to find the image location. Our experiments show that using raw pixels yields a poor performance to locate the image patches (see Table 2, third row). We note that even if the retrieved images look similar to



(a) (b)

Fig. 5. Image retrieval using shared representation comparison. (a) Selected image from a time series where a batch of 64 image patches (colored boxes) are extracted from; (b) Another image from the same time series is used to locate the selected image patches. The algorithm plots colored boxes corresponding to the nearest image patches in terms of shared representation distance.

the query images, they do not come from the same location. The recommended images using raw pixels are mainly based on the image color. For instance, whenever a harvest fields is used as query image the retrieved images correspond to harvested fields as well. This is not the case when using shared representations since seasonal changes are ignored in the shared representation. Additionally, we perform the same experiment using the representations extracted from the BicycleGAN [22] and VAE [13] models. As expected, since these models do not disentangle the spatial and temporal information of time series, the performance achieved is considerably poor (see Table 2, first and second rows).

Method	Recall@1
VAE [13]	0.4536
BicycleGAN [22]	0.2666
Raw pixels	0.5083
Ours	0.7372

Table 2. Image retrieval results in terms of Recall@1.

Model	Accuracy	Epochs
Fully-supervised	62.13%	50
VAE [13]	87.64%	10
BicycleGAN [22]	87.59%	10
Ours	92.38%	10

Table 3. Accuracy results in the test dataset.

4.4 Image classification

A common method to evaluate the performance of unsupervised feature representations is to apply them to perform image classification. We test the shared

representation extracted by our model using a novel dataset called EuroSAT [9]. It contains 27000 labeled images in 10 classes (residential area, sea, river, highway, etc.). We divide the dataset into a training and test dataset using a 80:20 split keeping a proportional number of examples per class.

We recover the shared representation encoder E_{sh} as feature extractor from our model. We append two fully-connected layers of 64 and 10 units, respectively on top of the feature extractor. We only train these fully-connected layers while keeping frozen the weights of the feature extractor in a supervised manner using the training split of EuroSAT. To provide a comparison, we train a fully-supervised model using the same architecture but randomly initialized weights. Additionally, we use the BicycleGAN [22] and VAE [13] models as feature extractors to train a classifier. Results can be observed in Table 3.

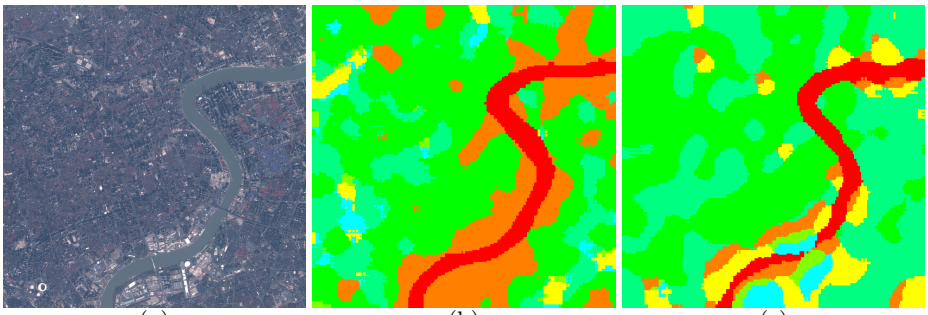
Our classifier achieves an accuracy of 92.38% outperforming the classifiers based on the BicycleGAN [22] and VAE [13] models. Nevertheless, it is important to note that using pretrained weights reduces the training time and allows to achieve better performance with respect to randomly initialized weights (62.13% of accuracy after 50 epochs).

4.5 Image segmentation

Since the shared representation is related to the location and texture of the image, we perform a qualitative experiment to illustrate its use for image segmentation. An image of size $1024 \times 1024 \times 4$ is randomly selected from a time series. Then, a sliding window of size $64 \times 64 \times 4$ and stride of size 32×32 is used to extract image patches. The shared representations extracted from these image patches are used to perform clustering via k-means. A new sliding window with a stride of 8×8 is used to extract the shared representations from the image. The extracted shared representations are assigned to a cluster. Since several clusters are assigned for each pixel, the cluster is decided by the majority of voted clusters. In Figure 6, a segmentation map example in Shanghai is displayed. Despite its simplicity, this unsupervised method achieves interesting results. It is able to segment the river, the port area and the residential area, among others. On the other hand, experiments using the raw pixels of the image as feature produce segmentation maps of lower visual quality.

4.6 Change detection

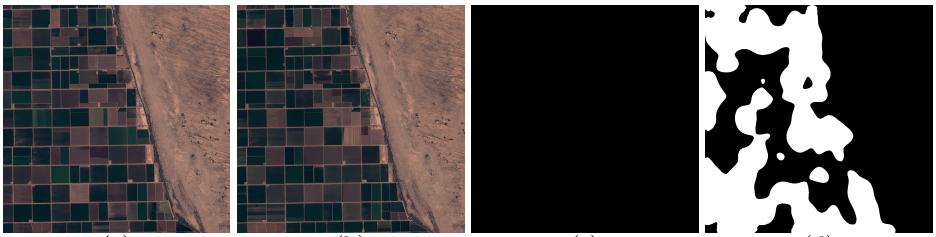
We perform an experiment to illustrate the use of the exclusive representation for seasonal change detection. Two images of size $1024 \times 1024 \times 4$ are selected from a time series. A sliding window of size $64 \times 64 \times 4$ is used to explore both images using a stride of size 32×32. As the window slides, the exclusive representations are extracted and compared using the L_1 distance. A threshold is defined to determine whether a change has occurred or not. Figure 7 shows an example of change detection maps using the shared and exclusive representations. As can be seen, the exclusive representation is able to identify seasonal changes while the shared representation is not as expected. Our experiments suggest that the



(a) (b) (c)

Fig. 6. Image segmentation in Shanghai, China. A sliding window is used to extract the shared representations of the image which in turn are used to perform clustering with 7 classes. (a) Image to be segmented; (b) Segmentation map using shared representations; (c) Segmentation map using raw pixels.

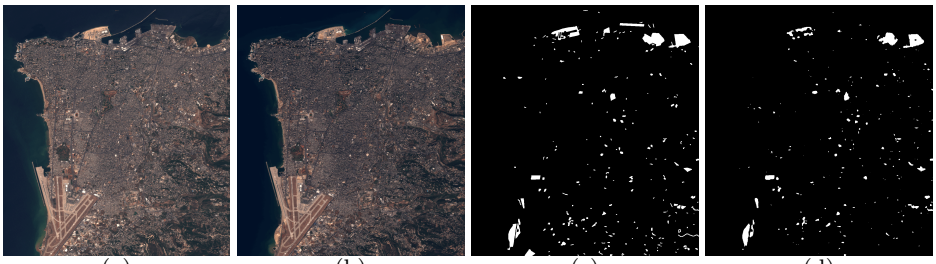
low-dimensional exclusive representation captures the factors of variation in time series generating visually coherent change detection maps.



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

Fig. 7. Seasonal change detection in Brawley, USA. (a) Image X; (b) Image Y; (c) Change detection map using shared representations (d) Change detection map using exclusive representations.

Additionally, we use our learned representations to perform urban change detection on the OSCD dataset [2] which provides 14 training images and 10 test images. Keeping frozen the weights of the encoders, we learn a decoder to create a change detection map of size 64×64. A sliding window is used to generate a complete change detection map. Figure 8 shows an urban change detection example. As the ground truth is not available for test images, the authors [2] provide a website to evaluate them. We obtain an average accuracy of 63.07% outperforming the VAE [13] and BicycleGAN [22] models which achieve an average accuracy of 59.31% and 60.01% respectively. We also train a fully-supervised model using the same architecture of our model but randomly initialized weights



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

Fig. 8. Urban change detection in Beirut, Lebanon. (a) Image X; (b) Image Y; (c) Ground truth; (d) Change detection map using our model.

which achieves an average accuracy of 60.67%. It suggests that the use of disentangled representations improves the results at image change detection.

5 Conclusion

In this work, we investigate how to obtain a suitable data representation of satellite image time series. We first present a model based on VAE and GAN methods combined with the cross-domain autoencoder principle. This model is able to learn a disentangled representation that consists of a common representation for the images of the same time series and an exclusive representation for each image. We train our model using Sentinel-2 time series which indicates that the model is able to deal with huge amounts of high-dimensional data. Finally, we show experimentally that the disentangled representation can be used to achieved interesting results at multiple tasks such as image classification, image retrieval, image segmentation and change detection. We think the learned representations can be improved by taking into account the time order of images in the model. We leave the development of such algorithm for future work.

Acknowledgments

We would like to thank the projects SYNAPSE and DEEL of the IRT Saint Exupéry for funding and providing computational resources to conduct the experiments.

References

1. Arjovsky, M., Chintala, S., Bottou, L.: Wasserstein generative adversarial networks. In: Proceedings of the 34th International Conference on Machine Learning (2017)
2. Caye Daudt, R., Le Saux, B., Boulch, A., Gousseau, Y.: Urban change detection for multispectral earth observation using convolutional neural networks. In: IEEE International Geoscience and Remote Sensing Symposium (IGARSS) (2018)

3. Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I., Abbeel, P.: Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In: Advances in Neural Information Processing Systems (2016)
4. Donahue, J., Krähenbühl, P., Darrell, T.: Adversarial feature learning. In: International Conference on Learning Representations (2017)
5. Gonzalez-Garcia, A., van de Weijer, J., Bengio, Y.: Image-to-image translation for cross-domain disentanglement. In: Advances in Neural Information Processing Systems (2018)
6. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Advances in neural information processing systems (2014)
7. Goodfellow, I.J.: NIPS 2016 tutorial: Generative adversarial networks (2016), http://arxiv.org/abs/1701.00160
8. Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., Courville, A.C.: Improved training of wasserstein GANs. In: Advances in Neural Information Processing Systems (2017)
9. Helber, P., Bischke, B., Dengel, A., Borth, D.: Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification (2017), http://arxiv.org/abs/1709.00029
10. Huang, X., Liu, M.Y., Belongie, S., Kautz, J.: Multimodal unsupervised image-to-image translation. In: European Conference on Computer Vision (2018)
11. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. In: Conference on Computer Vision and Pattern Recognition (2017)
12. Kempeneers, P., Soille, P.: Optimizing Sentinel-2 image selection in a big data context. Big Earth Data (2017)
13. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. In: International Conference on Learning Representations (2014)
14. Larsen, A.B.L., Sønderby, S.K., Larochelle, H., Winther, O.: Autoencoding beyond pixels using a learned similarity metric. In: Proceedings of The 33rd International Conference on Machine Learning (2016)
15. Ma, L., Jia, X., Georgoulis, S., Tuytelaars, T., Van Gool, L.: Exemplar guided unsupervised image-to-image translation. International Conference on Learning Representations (2019)
16. Mao, X., Li, Q., Xie, H., Lau, R.Y., Wang, Z., Smolley, S.P.: Least squares generative adversarial networks. In: International Conference on Computer Vision (2017)
17. Mirza, M., Osindero, S.: Conditional generative adversarial nets (2014), http://arxiv.org/abs/1411.1784
18. Radford, A., Metz, L., Chintala, S.: Unsupervised representation learning with deep convolutional generative adversarial networks. In: International Conference on Learning Representations (2016)
19. Theis, L., van den Oord, A., Bethge, M.: A note on the evaluation of generative models. In: International Conference on Learning Representations (2016)
20. Zhao, J.J., Mathieu, M., LeCun, Y.: Energy-based generative adversarial network. In: International Conference on Learning Representations (2017)
21. Zhu, J.Y., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. In: International Conference on Computer Vision (2017)
22. Zhu, J.Y., Zhang, R., Pathak, D., Darrell, T., Efros, A.A., Wang, O., Shechtman, E.: Toward multimodal image-to-image translation. In: Advances in Neural Information Processing Systems (2017)