Learning Disentangled Representations of Satellite Image Time Series

Eduardo Hugo Sanchez, Mathieu Serrurier, Mathias Ortner

▶ **To cite this version:**

Eduardo Hugo Sanchez, Mathieu Serrurier, Mathias Ortner. Learning Disentangled Representations of Satellite Image Time Series. 2019. hal-02046537

HAL Id: hal-02046537
https://hal.science/hal-02046537

Preprint submitted on 20 Mar 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Learning Disentangled Representations of Satellite Image Time Series

Eduardo H. Sanchez
IRT Saint Exupéry, IRIT
Toulouse, France

Mathieu Serrurier
IRIT
Toulouse, France

Mathias Ortner
IRT Saint Exupéry
Toulouse, France

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 from 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.

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 [15] 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 Goodfelow *et al*. [7, 8].

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*. [13] and Zhu *et al*. [24]. 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 knowledge about the specific information of each image could be useful to perform 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*. [25] to address multimodal generation and the model presented by Gonzalez-Garcia *et al*. [6]

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*. [6] under the GAN and VAE constraints to address representation disentanglement and multimodal generation. Our model is adapted to satellite image time series analysis using a simple architecture. 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. Third, our model generates a disentangled representation that isolates the common information of the entire time series and the exclusive information of each image. Finally, our experiments suggest that these feature representations are useful to perform several tasks such as image classification, image retrieval, image segmentation and change detection.

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 [15]. 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 an interesting method since it creates a low-dimensional representation where relevant attributes of data are captured.

Generative adversarial networks (GAN). Due to its great success in many different domains, GANs [7, 8] 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 [21], LSGAN [19], BEGAN [2], WGAN [1], WGAN-GP [9], LaplacianGAN [4], EBGAN [23], 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 [5] or VAE-GAN [16] 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*. [5].

Image-to-image translation. It is one of the most popular applications using conditional GANs [20]. 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 [13] and cycleGAN [24] 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*. [6], Zhu *et al*. [25], Huang *et al*. [12], Lee *et al*. [17] and Ma *et al*. [18]. 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*. [25] where a low-dimensional latent vector represents the diversity of the output domain. However, while the BicycleGAN model is mainly focus on image generation, we only consider the image-to-image translation task as a way to learn suitable feature representations. For image generation purpose, the output diversity is conditioned at the encoder input level in the BicycleGAN model. Instead the output diversity is conditioned at the decoder input level in our model.

Disentangled feature representation. Recent work is focused on learning disentangled 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*. [6] propose a model based on VAE-GAN image translators and a novel network component called cross-domain autoencoders. This model separates the feature 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. We propose a model that combines the cross-domain autoencoder component under the VAE and GAN constraints in order to create representations containing the common information of the entire time series and the exclusive information of each image. The VAE is used to create a low-dimensional representation that encodes the image variations related to acquisition time and the GAN is used to evaluate the generated image at a given time. We introduce a model adapted for satellite image time series using a simpler architecture since only four functions (the shared representation encoder, the exclusive representation encoder, the decoder and the discriminator) must be learned.

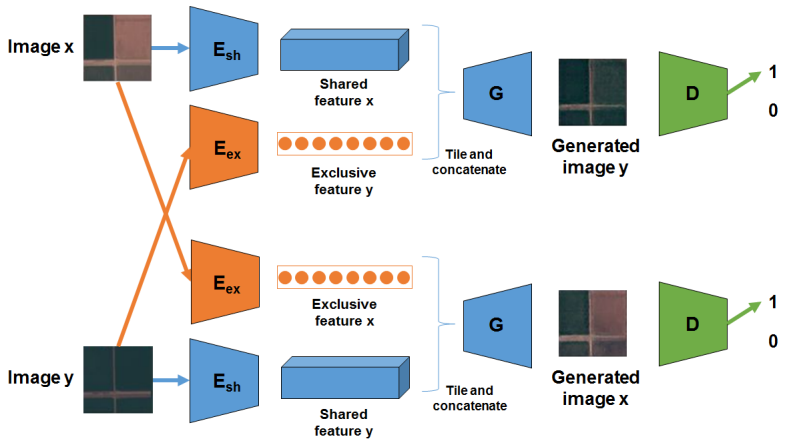


Figure 1. Model overview. The model goal is to learn both image transitions: $\mathbf{x} \rightarrow \mathbf{y}$ and $\mathbf{y} \rightarrow \mathbf{x}$. Both images are passed through the network $\mathbf{E}_{sh}$ in order to extract their shared representations. On the other hand, the network $\mathbf{E}_{ex}$ extracts the exclusive representations corresponding to images $\mathbf{x}$ and $\mathbf{y}$. The exclusive representation encoder output is constrained to follow a standard normal distribution. In order to generate the image $\mathbf{y}$, the decoder network $\mathbf{G}$ takes the shared feature of image $\mathbf{x}$ and the exclusive feature of image $\mathbf{y}$. A similar procedure is performed to generate the image $\mathbf{x}$. Finally, the discriminator $\mathbf{D}$ is used to evaluate the generated images.

3. Method

Let $\mathbf{x}, \mathbf{y}$ be two images randomly sampled from a given time series $\mathbf{t}$ in a region $\mathbf{c}$. 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 $\mathcal{E}$, $\mathcal{R} = [\mathcal{S}, \mathcal{E}]$. The subdomain $\mathcal{S}$ contains the common information between images $\mathbf{x}$ and $\mathbf{y}$ and the subdomain $\mathcal{E}$ contains the particular information of each image. Since images $\mathbf{x}$ and $\mathbf{y}$ belong to the same time series, their shared representations must be identical, *i.e.* $\mathcal{S}_{\mathbf{x}} = \mathcal{S}_{\mathbf{y}}$. On the other hand, as images are acquired at different times, their exclusive representations $\mathcal{E}_{\mathbf{x}}$ and $\mathcal{E}_{\mathbf{y}}$ correspond to the specific information of each image.

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

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

3.1. Objective function

As the work presented by Zhu *et al.* [25] and Gonzalez-Garcia *et al.* [6], our objective function is composed of several terms in order to obtain a disentangled representation.

Concerning the shared representation, images $\mathbf{x}$ and $\mathbf{y}$ must have identical shared feature representations, *i.e.* $\mathbf{E}_{sh}(\mathbf{x}) = \mathbf{E}_{sh}(\mathbf{y})$. A simple solution is to minimize the L_1 distance between their shared feature representations as can be seen in Equation 1.

$$\mathbf{L}_1^{sh} = \mathbb{E}_{\mathbf{x},\mathbf{y}\sim\mathcal{X}} \left[|\mathbf{E}_{sh}(\mathbf{x}) - \mathbf{E}_{sh}(\mathbf{y})| \right] \tag{1}$$

The exclusive representation must only contain the particular information that corresponds to each image. To enforce the disentanglement between shared and exclusive features, we include a reconstruction loss in the objective function where the shared representations of $\mathbf{x}$ and $\mathbf{y}$ are switched. The loss term corresponding to the reconstruction of image $\mathbf{x}$ is represented in Equation 2. Moreover, this loss term can be thought of as the reconstruction loss in the VAE model [15] which maximizes a lower bound of the log-likelihood function. As we enforce representation disentanglement, we simultaneously maximize the log-likelihood function which is equivalent to Kullback-Leibler divergence minimization between the real distribution and the generated distribution.

$$\mathbf{L}_1^{\mathbf{x},\mathbf{y}} = \mathbb{E}_{\mathbf{x},\mathbf{y}\sim\mathcal{X}} \left[|\mathbf{x} - \mathbf{G}(\mathbf{E}_{sh}(\mathbf{y}), \mathbf{E}_{ex}(\mathbf{x}))| \right] \tag{2}$$

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 feature 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. In contrast to the approach employed by Gonzalez-Garcia *et al*. [6] which uses a GAN approach to constraint the exclusive feature distribution, a simpler solution which proves to be effective is to include a Kullback-Leibler divergence term between the distribution of the exclusive feature representation and the prior $\mathcal{N}(0, I)$. Assuming that the exclusive feature encoder $\mathbf{E}_{ex}(\mathbf{x})$ is distributed as a normal distribution $\mathcal{N}(\mu_{\mathbf{E}_{ex}(\mathbf{x})}, \sigma_{\mathbf{E}_{ex}(\mathbf{x})})$, the Kullback-Leibler divergence can be written as follows

$$\mathbf{L}^{\mathbf{x}}_{KL} = -\tfrac{1}{2}\mathbb{E}_{\mathbf{x}\sim\mathcal{X}}\left[1 + \log(\sigma^2_{\mathbf{E}_{ex}(\mathbf{x})}) - \mu^2_{\mathbf{E}_{ex}(\mathbf{x})} - \sigma^2_{\mathbf{E}_{ex}(\mathbf{x})}\right] \quad (3)$$

In order to minimize the distance between the real distribution and the generated distribution of images, a LSGAN loss [19] is included in the objective function. 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*. $\mathbf{D}(\mathbf{G}(\mathbf{E}_{sh}(\mathbf{y}), \mathbf{E}_{ex}(\mathbf{x}))) \rightarrow 1$. The corresponding loss term for image $\mathbf{x}$ and its reconstructed version can be seen in Equation 4 where the discriminator maximizes this term while the generator minimizes it.

$$\begin{aligned}\mathbf{L}^{\mathbf{x}}_{GAN} = \mathbb{E}_{\mathbf{x}\sim\mathcal{X}}\left[(\mathbf{D}(\mathbf{x}))^2\right] + \\ \mathbb{E}_{\mathbf{x},\mathbf{y}\sim\mathcal{X}}\left[(1 - \mathbf{D}(\mathbf{G}(\mathbf{E}_{sh}(\mathbf{y}), \mathbf{E}_{ex}(\mathbf{x}))))^2\right]\end{aligned} \quad (4)$$

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 ($\mathbf{E}_{ex}$, $\mathbf{E}_{sh}$, $\mathbf{G}$) while it is maximized by the discriminator $\mathbf{D}$.

$$\begin{aligned}\min_{\mathbf{E}_{ex},\mathbf{E}_{sh},\mathbf{G}} \max_{\mathbf{D}} \mathcal{L} = \mathbf{L}^{\mathbf{x}}_{GAN} + \mathbf{L}^{\mathbf{y}}_{GAN} + \lambda_{L_1}(\mathbf{L}^{\mathbf{x},\mathbf{y}}_1 + \mathbf{L}^{\mathbf{y},\mathbf{x}}_1) \\ + \lambda_{L_{KL}}(\mathbf{L}^{\mathbf{x}}_{KL} + \mathbf{L}^{\mathbf{y}}_{KL}) + \lambda^{sh}_{L_1}\mathbf{L}^{sh}_1\end{aligned} \quad (5)$$

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

3.2. Implementation

Network architectures: Our model is architectured around four network blocks: the shared feature encoder, the exclusive feature encoder, the decoder and the discriminator. The shared feature encoder is composed of 5 convolutional layers while the exclusive feature encoder is composed of a first convolutional layer and three consecutive ResNet blocks [10]. Since the exclusive feature encoder must provide a normally distributed vector, 2 fully-connected layers of size 64 are appended on top of the ResNet blocks to estimate its mean value μ and standard deviation σ. The decoder consists of 4 transposed convolutional layers. Finally, the discriminator is composed of 5 convolutional layers. During training and test experiments, we use batch normalization in all the networks. All the layers use a kernel of size 4×4, a stride of 2 and leaky ReLU as activation function (except the discriminator and the decoder outputs where the sigmoid and hyperbolic tangent functions are used, respectively).

Optimization setting: To train our model, we use batches of 64 randomly selected pairs of images of size $64 \times 64 \times 4$ from our time series dataset. 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: $\lambda_{L_1} = 10$, $\lambda^{sh}_{L_1} = 0.5$ and $\lambda_{L_{KL}} = 0.01$ during training. The training procedure is summarized in Algorithm 1.

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 this paper, we use the RGBI bands which correspond to bands at 10m spatial resolution. 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 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. As explained by Kempeneers and Soille [14], 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. Then, we selected 25 patches of size 1024×1024 from the center of the tiles to reduce the effect of the satellite orbit view angle. Finally, our dataset is composed of 1050 times series each of which is composed of 12 images of size $1024 \times 1024 \times 4$.

In order to analyze the entire time series using smaller patches the following strategy is applied: a batch of time

Algorithm 1 Training algorithm.

Random initialization of model parameters
$(\theta_{\mathbf{D}}^{(0)}, \theta_{\mathbf{E}_{sh}}^{(0)}, \theta_{\mathbf{E}_{ex}}^{(0)}, \theta_{\mathbf{G}}^{(0)})$
for $k = 1; k = k + 1; k <$ number of iterations **do**
  Sample a batch of m time series $\{\mathbf{t}_s^{(1)}, ..., \mathbf{t}_s^{(m)}\}$
  Sample a batch of m image pairs
  $\{(\mathbf{x}^{(1)}, \mathbf{y}^{(1)}), ..., (\mathbf{x}^{(m)}, \mathbf{y}^{(m)})\}$ from $\{\mathbf{t}_s^{(i)}\}$
  Compute $\mathcal{L}^{(k)}(\mathbf{x}^{(i)}, \mathbf{y}^{(i)}, \theta_{\mathbf{D}}^{(k)}, \theta_{\mathbf{E}_{sh}}^{(k)}, \theta_{\mathbf{E}_{ex}}^{(k)}, \theta_{\mathbf{G}}^{(k)})$

$$
\begin{aligned}
\mathcal{L}^{(k)} = \frac{1}{m}\sum_{i=1}^{m}\Bigg[& \left(\mathbf{D}(\mathbf{x}^{(i)})\right)^2 + \left(\mathbf{D}(\mathbf{y}^{(i)})\right)^2 \\
& + \left(1 - \mathbf{D}(\mathbf{G}(\mathbf{E}_{sh}(\mathbf{y}^{(i)}), \mathbf{E}_{ex}(\mathbf{x}^{(i)})))\right)^2 \\
& + \left(1 - \mathbf{D}(\mathbf{G}(\mathbf{E}_{sh}(\mathbf{x}^{(i)}), \mathbf{E}_{ex}(\mathbf{y}^{(i)})))\right)^2 \\
& + \lambda_{L_1}\Big(|\mathbf{x}^{(i)} - \mathbf{G}(\mathbf{E}_{sh}(\mathbf{y}^{(i)}), \mathbf{E}_{ex}(\mathbf{x}^{(i)}))| \\
& + |\mathbf{y}^{(i)} - \mathbf{G}(\mathbf{E}_{sh}(\mathbf{x}^{(i)}), \mathbf{E}_{ex}(\mathbf{y}^{(i)}))|\Big) \\
& + \lambda_{L_1}^{sh}\Big(|\mathbf{E}_{sh}(\mathbf{x}^{(i)}) - \mathbf{E}_{sh}(\mathbf{y}^{(i)})|\Big) \\
& - \frac{1}{2}\lambda_{L_{KL}}\Big(2 + \log(\sigma^2_{\mathbf{E}_{ex}(\mathbf{x}^{(i)})}) - \mu^2_{\mathbf{E}_{ex}(\mathbf{x}^{(i)})} \\
& - \sigma^2_{\mathbf{E}_{ex}(\mathbf{x}^{(i)})} + \log(\sigma^2_{\mathbf{E}_{ex}(\mathbf{y}^{(i)})}) - \mu^2_{\mathbf{E}_{ex}(\mathbf{y}^{(i)})} \\
& - \sigma^2_{\mathbf{E}_{ex}(\mathbf{y}^{(i)})}\Big)\Bigg]
\end{aligned}
\tag{6}
$$

  Update the model parameters:

$$\theta_{\mathbf{D}}^{(k+1)} \leftarrow \text{Adam}\left(-\nabla_{\theta_{\mathbf{D}}^{(k)}}\mathcal{L}^{(k)}, \theta_{\mathbf{D}}^{(k)}\right) \tag{7}$$

$$\theta_{\mathbf{E}_{sh}}^{(k+1)} \leftarrow \text{Adam}\left(\nabla_{\theta_{\mathbf{E}_{sh}}^{(k)}}\mathcal{L}^{(k)}, \theta_{\mathbf{E}_{sh}}^{(k)}\right) \tag{8}$$

$$\theta_{\mathbf{E}_{ex}}^{(k+1)} \leftarrow \text{Adam}\left(\nabla_{\theta_{\mathbf{E}_{ex}}^{(k)}}\mathcal{L}^{(k)}, \theta_{\mathbf{E}_{ex}}^{(k)}\right) \tag{9}$$

$$\theta_{\mathbf{G}}^{(k+1)} \leftarrow \text{Adam}\left(\nabla_{\theta_{\mathbf{G}}^{(k)}}\mathcal{L}^{(k)}, \theta_{\mathbf{G}}^{(k)}\right) \tag{10}$$

end for

series composed of images of size $64 \times 64 \times 4$ is randomly sampled from the time series of size $1024 \times 1024 \times 4$. Since our model takes 2 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 in [22]. 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.

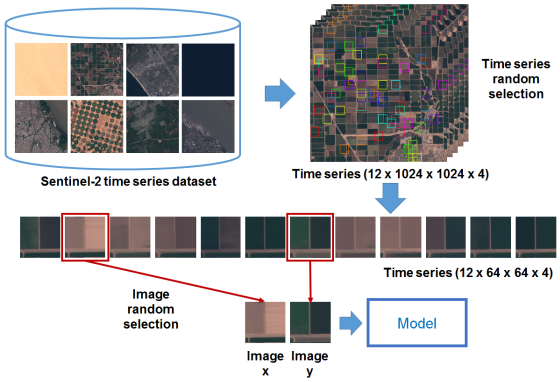


Figure 2. Training data selection. A batch of smaller time series is randomly sampled from the dataset. At each iteration two images are randomly selected from each time series to be used as input for our model.

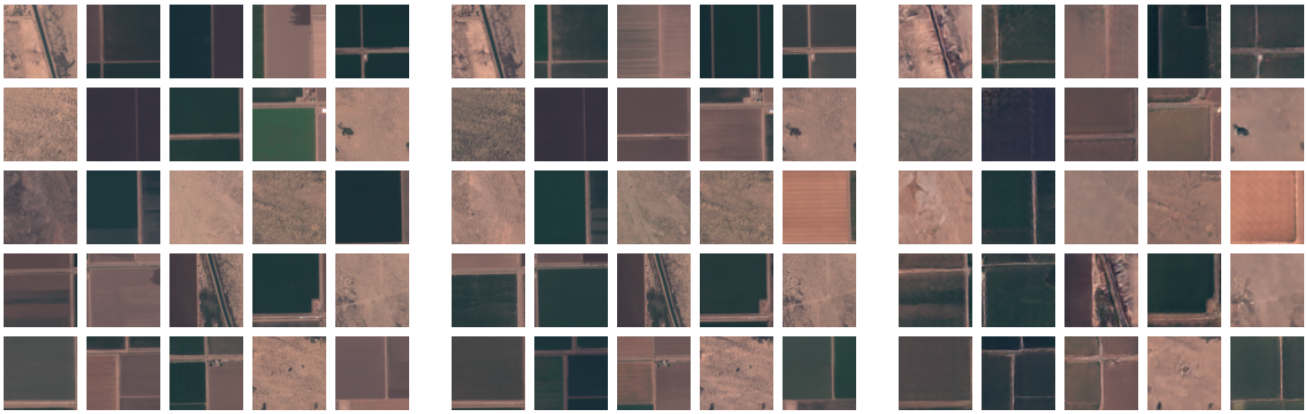
4.2. Image-to-image translation

It seems natural to first test the model performance at image translation. We use a test dataset which is composed of time series acquired from different tiles to guarantee that training and test datasets are independent. For each dataset, 150 batches of 64 time series are randomly selected. It represents around 20k processed images of size $64 \times 64 \times 4$.

An example of image-to-image translation can be observed in Figure 3. For instance, let us consider the image in the third row, fifth column. The shared feature is extracted from an image $\mathbf{x}$ which corresponds to growing crop fields while the exclusive feature is extracted from another image $\mathbf{y}$ where fields have been harvested. Consequently, the generated image contains harvested fields which is defined by the exclusive feature of image $\mathbf{y}$. 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 in the generator part of the model.

We quantify the L_1 distance between generated images $\mathbf{G}(\mathbf{E}_{sh}(\mathbf{x}), \mathbf{E}_{ex}(\mathbf{y}))$ and images $\mathbf{y}$ used to extract the exclusive feature. Results can be observed in Table 1 (first row). Pixel values in generated images and real images are in the range of $[-1, 1]$, thus a mean difference of 0.0152 in the training dataset indicates that the model performs well at image-to-image translation. A slightly difference is obtained in the test dataset where the L_1 distance is 0.0207.

A special image-to-image translation case is image autoencoding where the shared and exclusive features are extracted from the same image. Additionally, we compute the L_1 distance between images $\mathbf{x}$ and autoencoded im-



(a) (b) (c)

Figure 3. Image translation performed on images of Brawley, California. (a) Images used to extract the shared features; (b) Images used to extract the exclusive features; (c) Generated images from the shared representation of (a) and the exclusive representation of (b).

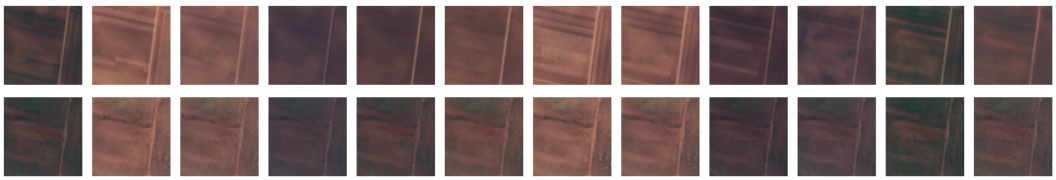


Figure 4. Multimodal generation. The first row corresponds to a time series sampled from the test dataset. The second row corresponds to a time series where each image is generated by using the same shared feature and only modifying the exclusive feature.

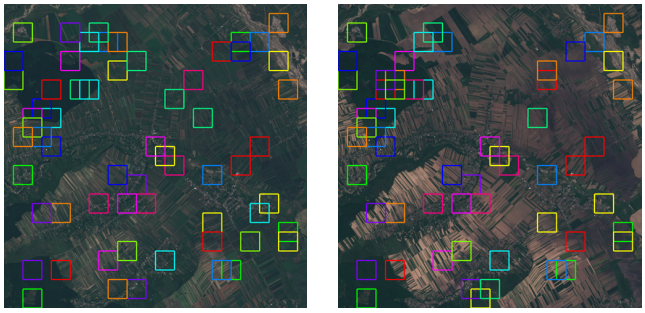
ages $\mathbf{G}(\mathbf{E}_{sh}(\mathbf{x}), \mathbf{E}_{ex}(\mathbf{x}))$ for comparison purpose in Table 1 (second row). Lower values in terms of L_1 distance are obtained with respect to those of image-to-image translation. It is important to note that input images are considerably well reconstructed even if this case is not considered during training. Finally, we perform times series reconstruction in order to show that the exclusive feature encodes the specific information of each image. An image is randomly selected from a time series to extract its shared feature. While keeping the shared feature constant and only modifying the exclusive feature, we reconstruct the original time series. Results in terms of L_1 distance between the original time series and the reconstructed one can be observed in Table 1 (third row). Similar values to those of image-to-image translation are obtained. An example of time series reconstruction can be seen in Figure 4.

Task	Training set	Test set
Image translation	0.0152 ± 0.0573	0.0207 ± 0.0774
Autoencoding	0.0084 ± 0.0309	0.0087 ± 0.0312
Time series	0.0177 ± 0.0622	0.0223 ± 0.0828

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

4.3. Image retrieval

In this experiment, we want to evaluate whether the shared feature 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 5(a). The corresponding shared features are extracted for each image of the batch. The main idea is to use the information provided by the shared feature 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 features 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



(a) (b)

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

analyzed.

To illustrate the retrieval algorithm, let us consider a test image of agricultural fields. We plot the patches to be retrieved in Figure 5(a) and the retrieved patches by the algorithm in Figure 5(b). 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 (first 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.

Method	Recall@1
Shared features	0.7372
Raw pixels	0.5083

Table 2. Image retrieval results in terms of Recall@1 using the shared feature representation and the raw pixels of the image as feature.

As a baseline to compare to the retrieval image based on the shared features, we use the raw pixels of the image to find the image location. Our experiments show that using raw pixels as feature yields a poor performance to locate the patches (see Table 2, second row). We note that even if the retrieved images look similar to the query images, they do not come from the same location. The recommended images using raw pixels are mainly based on the image color. 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 features since seasonal changes are ignored in the shared representation.

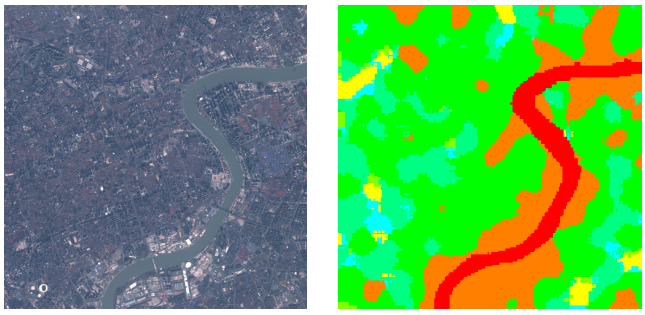
4.4. Image classification

A common method to evaluate the performance of unsupervised features is to apply them to perform image classification. We test the shared features extracted by our model using a novel dataset called EuroSAT [11]. 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 feature encoder $\mathbf{E}_{sh}(\cdot)$ as feature extractor from the pretrained 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. Results can be observed in Table 3. We obtain an accuracy of 92.38% while we achieve an accuracy of 94.54% by not freezing the weights of the feature extractor during training. 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).

Model	Accuracy	Epochs
Pretrained + Fine-tuning	92.38%	10
Pretrained + Full-training	94.54%	10
From scratch	62.13%	50

Table 3. Accuracy results in the test dataset from classification experiments.



(a) (b)

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

(a) (b) (c)

Figure 7. Change detection in Brawley, USA. A window slides on the images to be compared and extracts their exclusive features. (a) Image $\mathbf{x}$; (b) Image $\mathbf{y}$; (c) A red colored box is plotted in the regions where the L_1 distance of the exclusive features extracted by the window is higher than a user-selected threshold.

Higher accuracy (98.57%) in the EuroSAT dataset is claimed by Helber *et al*. [11] using supervised pretrained GoogLeNet or ResNet-50 models. However, we show that using a very simple model trained in an unsupervised manner allows us to obtain excellent results.

4.5. Image segmentation

Since the shared feature representation are related to the location and texture of the image, we perform a qualitative experiment to illustrate that it can be used to perform 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 the patches. The shared features extracted from these 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 features from the image. Each of the extracted shared features is 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. As can be seen, this unsupervised image segmentation method achieves interesting results. It is able to segment the river, the port area and the residential area, among others. We think that segmentation results can be improved by using a smaller windows to achieve better resolution. On the other hand, experiments using the raw pixels of the image as features produce segmentation maps of lower visual quality.

4.6. Change detection

Another qualitative experiment is performed in order to illustrate some properties of the exclusive feature representation. Since the particular information of each image of a time series is contained in the exclusive feature, we leverage this information to propose a naive change detection method. Two images of size $1024 \times 1024 \times 4$ are selected from a given 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 features are extracted and compared using the L_1 distance. Then, a threshold is defined to determine whether a change has occurred or not. If the L_1 distance is higher than the threshold, a red colored box is plotted indicating that change has occurred. An example can be seen in Figure 7. Despite the method simplicity, our experiments suggest that the low-dimensional exclusive feature captures the factors of variation in time series generating visually coherent change detection maps.

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.

References

[1] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein generative adversarial networks. In D. Precup and Y. W. Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 214–223, 2017. 2

[2] D. Berthelot, T. Schumm, and L. Metz. BEGAN: boundary equilibrium generative adversarial networks. *CoRR*, abs/1703.10717, 2017. 2

[3] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2172–2180, 2016. 2

[4] E. L. Denton, S. Chintala, R. Fergus, et al. Deep generative image models using a laplacian pyramid of adversarial networks. In *Advances in neural information processing systems*, pages 1486–1494, 2015. 2

[5] J. Donahue, P. Krähenbühl, and T. Darrell. Adversarial feature learning. In *International Conference on Learning Representations*, 2017. 2

[6] A. Gonzalez-Garcia, J. van de Weijer, and Y. Bengio. Image-to-image translation for cross-domain disentanglement. *arXiv preprint arXiv:1805.09730*, 2018. 1, 2, 3, 4

[7] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014. 1, 2

[8] I. J. Goodfellow. NIPS 2016 tutorial: Generative adversarial networks. *CoRR*, abs/1701.00160, 2016. 1, 2

[9] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville. Improved training of wasserstein GANs. In *Advances in Neural Information Processing Systems*, pages 5769–5779, 2017. 2

[10] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 4

[11] P. Helber, B. Bischke, A. Dengel, and D. Borth. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *CoRR*, abs/1709.00029, 2017. 7, 8

[12] X. Huang, M.-Y. Liu, S. Belongie, and J. Kautz. Multimodal unsupervised image-to-image translation. *arXiv preprint arXiv:1804.04732*, 2018. 2

[13] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 1, 2

[14] P. Kempeneers and P. Soille. Optimizing Sentinel-2 image selection in a big data context. *Big Earth Data*, 1(1-2):145–158, 2017. 4

[15] D. P. Kingma and M. Welling. Auto-encoding variational bayes. In *International Conference on Learning Representations*, 2014. 1, 2, 3

[16] A. B. L. Larsen, S. K. Snderby, H. Larochelle, and O. Winther. Autoencoding beyond pixels using a learned similarity metric. In M. F. Balcan and K. Q. Weinberger, editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1558–1566, 2016. 2

[17] A. X. Lee, R. Zhang, F. Ebert, P. Abbeel, C. Finn, and S. Levine. Stochastic adversarial video prediction. *arXiv preprint arXiv:1804.01523*, 2018. 2

[18] L. Ma, X. Jia, S. Georgoulis, T. Tuytelaars, and L. Van Gool. Exemplar guided unsupervised image-to-image translation. *arXiv preprint arXiv:1805.11145*, 2018. 2

[19] X. Mao, Q. Li, H. Xie, R. Y. Lau, Z. Wang, and S. P. Smolley. Least squares generative adversarial networks. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2813–2821. IEEE, 2017. 2, 4

[20] M. Mirza and S. Osindero. Conditional generative adversarial nets. *CoRR*, abs/1411.1784, 2014. 2

[21] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In *International Conference on Learning Representations*, 2016. 2

[22] L. Theis, A. van den Oord, and M. Bethge. A note on the evaluation of generative models. In *International Conference on Learning Representations*, 2016. 5

[23] J. J. Zhao, M. Mathieu, and Y. LeCun. Energy-based generative adversarial network. In *International Conference on Learning Representations*, 2017. 2

[24] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, 2017. 1, 2

[25] J.-Y. Zhu, R. Zhang, D. Pathak, T. Darrell, A. A. Efros, O. Wang, and E. Shechtman. Toward multimodal image-to-image translation. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 465–476. Curran Associates, Inc., 2017. 1, 2, 3