Sentinel-2 RGB and NIR bands simulation after fire events using a multi-temporal conditional generative adversarial network

Iris Dumeur, Yang Chen, Frédéric Sur, Zheng-Shu Zhou

Sentinel-2 RGB and NIR bands simulation after fire events using a multi-temporal conditional generative adversarial network

Iris Dumeur[*], Yang Chen[†], Frédéric Sur[‡], Zheng-Shu Zhou[§]

Abstract

In remote sensing applications, optical images are widely used to monitor land changes. However, clouds, haze, or smoke hide the area below and, therefore, limit the use of optical data to favorable weather conditions. Since the SAR signal can penetrate through clouds, haze, or smoke, it has been recently proposed to combine optical images and SAR data to overcome this limitation. In this report, we investigate a deep-learning model based on a multi-temporal conditional generative adversarial neural network that generates optical images from SAR data, based on optical cloud-free images and SAR data previously acquired. Quantitative and qualitative results over the region of Goulburn, Australia, are also provided to evaluate the effectiveness of this multi-temporal approach in monitoring vegetation changes after fire events. Software code is publicly available.

Keywords: land monitoring; cloud, haze, or smoke removal; conditional generative adversarial neural network; multi-temporal SAR and optical data

[*]École des Mines de Nancy, Université de Lorraine, France
[†]CSIRO Data61, Australia
[‡]LORIA (Université de Lorraine, CNRS, INRIA), Nancy, France.
[§]CSIRO Data61, Perth, Australia

1 Introduction

1.1 Context

Satellite remote sensing imagery provides the consistent and regular observation of the Earth surface with a wide range of the electromagnetic spectrum in time and space. Different objects have different tendencies to selectively absorb, reflect or transmit light or electromagnetic waves at certain frequencies. Optical satellite remote sensing technologies have been used for climate change assessment, landcover, and land-use change detection, deforestation and urbanisation mapping, monitoring natural hazards, and assessing their impact on the environment and the community. Optical imagery using passive sensors can only be used to detect and measure the reflected energy when the natural energy sources are available (i.e. the sun). The common limitation that persists in optical imagery for Earth surface observation activities is the presence of thick clouds and smoke [16]. They appear opaque in the optical frequency bands and contaminate the reflectance signal and ultimately obstruct the detection of the objects underneath.

Various methods for temporal gap-filling [19], spatial filtering [17], and multi optical sensor data blending [4, 5] were introduced to address the missing data issue in historical satellite images caused by clouds and smoke contamination, but they are unable to capture true events (e.g., floods and bushfires) that rapidly evolve, especially when factoring in the potential clouds. Consequently, they fail to provide information to predict the severity and the impact areas. The disaster and emergency management agencies share a common challenge in providing instant assistance to the community and the stakeholders, due to a lack of information for assessing damaged properties, land, and ecosystems in real-time.

Diverging imaging capabilities can be reconciled by blending optical images from high-temporal-frequency (HTF) and high-spatial-resolution (HSR) sensors (e.g., Sentinel-2, abbreviated as S2 here) with radar or Lidar data from active sensors (e.g., Sentinel-1, abbreviated as S1, or GEDI) to produce images that possess the HTF and HSR characteristics across large areas regardless of the weather conditions [3, 16]. Radars work in the microwave frequency range with wavelengths longer than the optical bands which can penetrate through clouds to sense the objects underneath. They are also proven to be sensitive to vegetation changes[1, 14]. Blending active and passive remote sensing images to overcome the challenge in monitoring Earth surface in near real-time is an emerging field in the remote sensing domain given the accessibility of supercomputing systems and powerful data-driven supervised and unsupervised learning algorithms.

To this end, a deep learning model is trained to learn the correlations between S1 and S2 images at a time t_1 when the optical S2 image is not occluded, in order to subsequently infer the S2 image from the radar S1 image at a different time t_2. The model used here is a Generative Adversarial Network (GAN) [7] which, from a recent study [10], looks promising in such an application.

More specifically, the region of interest for this study is located in the east of Goulburn,

New South Wale (Australia). It is imaged in a map of size 5389×7851 pixels with a resolution of 10 meters (one pixel covers 10 square meters). The majority of this area is covered by state forests and agricultural crops, as illustrated in Figure 1. The Wingello State Forest in the southeast of the region of interest was affected by a bushfire that occurred in December 2019, the fire front line being clearly visible in the right bottom corner of the shortwave infrared (SWIR) band.

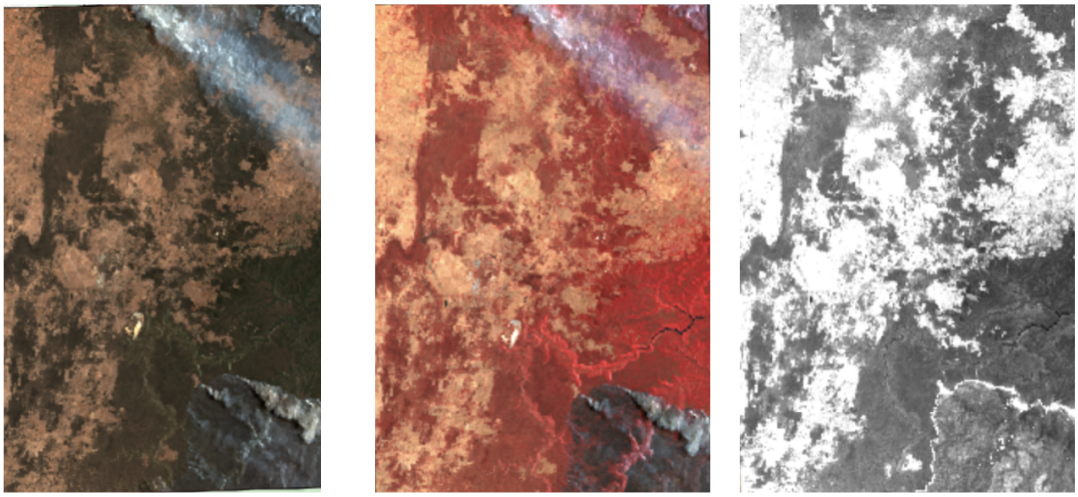

Figure 1: Region of interest imaged by Sentinel-2 satellite on the December 31st 2019. From left to right, true color (RGB), false color (NIR,R,G) and SWIR band. The smoke is visible in the R,G,B,NIR bands, and the fire front line is visible in the right bottom corner of the SWIR band.

Figure 2 shows the area of interest before and after this fire, namely at dates t_1 corresponding to November 6th 2019 and t_2 to January 29th 2020. Many vegetation changes can be seen in the forest area (framed in blue in Figure 2) and in the cropping land as well (framed in red in Figure 2), this latter area being affected by inter-seasonal changes due to the plant phenology and the management strategies. Our aim is to demonstrate that such a multi-temporal conditional generative adversarial network is able to generate relevant optical images at t_2 from optical images at t_1 and SAR data at t_1 and t_2. This is especially challenging after a fire effect as the reflectance dramatically changes.

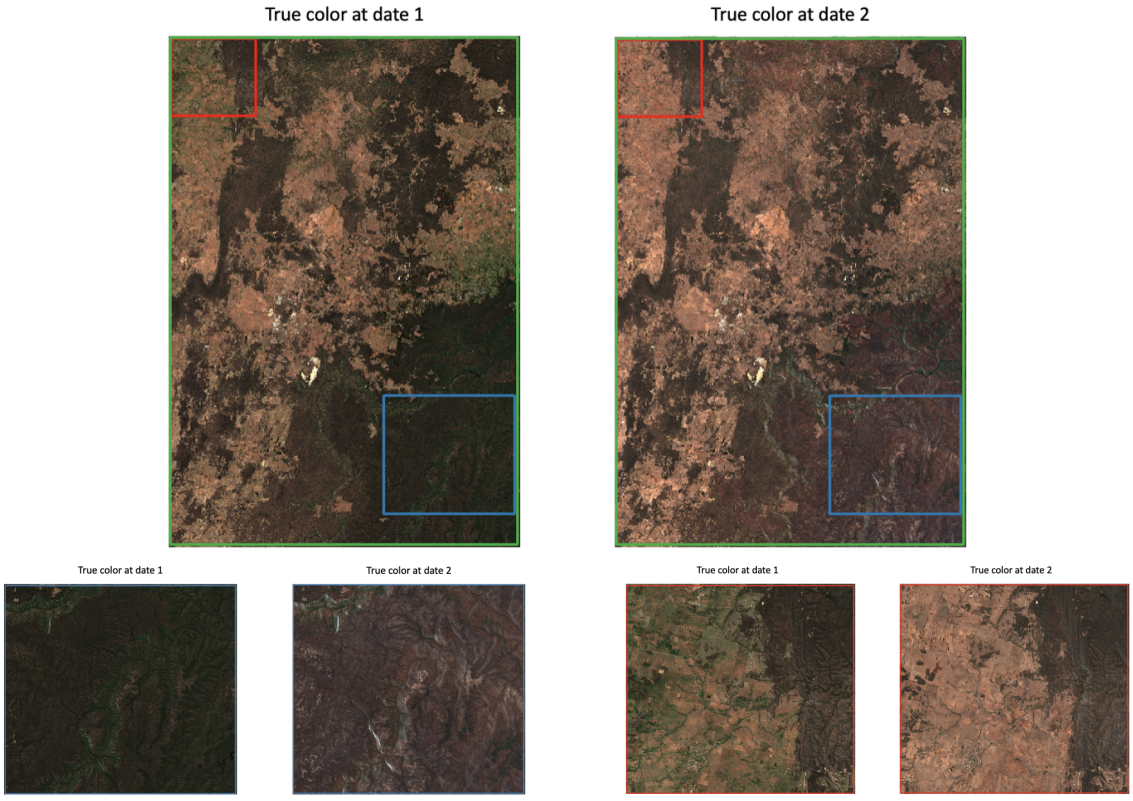

Figure 2: Region of interest in the true color images before and after the bushfire in the upper row, and close-up views of the framed areas in the lower row (forest region framed in blue, and agricultural crop area in red). On the left: data from $t1$ (November 6th, 2019). On the right:data from $t2$ (January 29th, 2020).

1.2 Related work: deep learning for cloud and smoke removal from optical images

While merging data from different sources or acquisition modalities is a keystone of several remote sensing applications, recent years have seen a shift from traditional image processing methods (see for instance [11, 23]) to machine learning approaches. A popular application is to digitally remove clouds, haze, or smoke impairing optical images by making use of SAR data. In particular, generative adversarial networks (GAN) [7] and methods derive from it as conditional GAN (cGAN) [12] or cycleGAN [25] have been recently introduced to solve this problem. GANs and related methods are popular approaches in image processing to transfer the style of an image to another one. In the present study, the goal is not only to transfer the style of optical images to SAR data, but also to reproduce actual details in

synthetic optical images from SAR data, in spite of the presence of clouds, haze, or smoke.

For instance, a cycleGAN is trained in [20] to reconstruct optical S2 information hidden by clouds from unpaired images from S1 images in cloudy and uncloudy situations. In [6], a GAN architecture is used to reconstruct an optical S2-like image from S1 data. Since cloudy images together with the corresponding cloud-free versions are required for the learning step, clouds are manually added to S2 images with a photo-editing software. In order to realistically remove clouds, it is indeed possible to take into account information from SAR data, which is not impaired by clouds, at a very near date, instead of inferring information from optical images taken without clouds, potentially at a very different date. As suggested by several authors, a correlation can be shown between optical and SAR data: it has been demonstrated [8, 15] that SAR-like data can be rendered from optical images (and vice versa). In order to go beyond the single-temporal method described in [6], a promising approach is thus to consider information from SAR and optical data at a time t_1 where no clouds impair the optical images, together with SAR data acquired when the zone of interest is covered by clouds at a time t_2: the correlation between SAR and optical data at t_1 permits to render a cloud-free optical image at t_2 from SAR data at the same time. The rendered image is expected to reproduce details present in SAR data. Such an approach is investigated in [10]: a multi-temporal conditional generative adversarial network (MTcGAN) is designed to generate S2 data from S1 and S2 data at time t_1 and S1 data at time t_2. A very recent paper [24] elaborates on this approach and proposes a so-called multi-channel conditional generative adversarial network (MCcGAN). Compared to MTcGAN, the main difference is the architecture of the generator network. It is shown that such a multitemporal approach does not only successfully transfer the style of optical data to SAR images, but is also able to recover actual details and to track vegetation changes in spite of clouds. The authors of [24] show that MCcGAN is better than MTcGAN in cropland and not as effective in mountain or town areas.

While these recent papers address cloud or smoke removal through conditional generative adversarial networks, they do not explicitly address cropland changes. To the best of our knowledge, a study such as the one we propose, dedicated to the situation where croplands have changed between both acquisition times, for example because of a fire event, is still to be done.

1.3 Organization of the report and contribution

Section 2 describes the methods. First, we detail data processing, which is an important matter both in machine learning algorithms and in remote sensing applications. While we reproduce the main lines of the conditional generative adversarial network introduced in [10], we also incorporate some tricks from the recent literature in image-to-image translation. The experimental setting and the metrics on which assessment relies are also detailed. Section 3 shows illustrative and quantitative results. We conclude with Section 4. Additional results are available in Appendix A.

The generated images turn out to be comparable to the results of other studies [6, 10] in similar, yet different, application contexts. The cGAN model is shown to provide valuable optical images in regions affected by changes between times t_1 and t_2. To the best of our knowledge, software implementation of the model of [10] is not publicly available. The present study confirms that the results of this paper can be reproduced. Software code is publicly available at the following URL: `https://github.com/irisdum/cGAN_sent2_sim`

2 Methods

2.1 Data preprocessing

Preprocessing of the Sentinel-1 images corresponding to the region shown in Figure 2 was done using the GPT tool of the SNAP software[1] distributed by the European Space Agency. It consists in the following operations: Apply Orbit File, Thermal Noise Removal, Remove GRD Border Noise, Calibration, Terrain Flattening, Speckle Filter, Multilook and Terrain Correction. With the "Terrain Correction" operation, the S1 images are projected onto the WGS84 UTM 55S, same as the datum and projection of the S2 data. After Sentinel-1 preprocessing with SNAP, the advanced interpolation methods used during the processing may conduct some spurious negative values in the image. Negative SAR data are replaced by the average value of the neighboring pixels. Satellite data used in this study are the Sentinel-1 IW VV and VH polarization data and the R, G, B and NIR bands of Sentinel-2 imagery. We do not consider SWIR band in Sentinel-2 data since they have a lower resolution (20 meters) than R, G, B and NIR bands (10 meters).

Eventually, the preprocessed Sentinel images are split into 623 patches (called tiles here) of size 256×256 pixels using GDAL/OGR[2], a licensed translator library for raster and vector geospatial data formats. These tiles are randomly divided into three datasets as shown in Table 1, namely train (80% of the data), validation (5%), and test (15%) datasets, which will be used respectively to train the cGAN, to tune the hyper-parameter and to assess the model performance.

	train	validation	test
number of tiles	496	32	95

Table 1: Number of tiles in the train, validation, and test datasets.

Input features were standardized, as it is a common practice in machine learning. Standardization parameters is computed on the training dataset and used afterwards on validation and test datasets. We have noticed that red, green, and blue bands have similar

[1] `https://step.esa.int/main/toolboxes/snap/`
[2] `https://gdal.org`

statistical distributions. In order to keep the relationship between these bands, the mean and standard deviation used to rescale those data were computed over all these bands. Because NIR band, as well as VV and VH polarization channels, have different statistical distributions, the mean and the standard deviation were computed on each band separately. The resulting distributions showing heavy tails, the values of optical bands and radar data were divided by constants, equal to 7 and 5 respectively, in order to rescale data and ensure the numerical stability of the training process.

2.2 Conditional generative adversarial network

A generator is used to render a 256×256 optical image at t_2 from S1 (SAR) and S2 (optical) data at time t_1, together with S1 data at time t_2. We adopt the general approach proposed by the authors of [10] in which a conditional generative adversarial network (cGAN) is used to generate optical images from multi-temporal SAR and optical data. A GAN [7] consists of the combination of two convolutional neural networks (CNN), namely a generator and a discriminator, which are simultaneously trained. While in a traditional GAN the input of the generator is simply noise, additional information (such as one or several images) is provided in a cGAN, which makes it a successful approach in many image-to-image translation problems [12, 18, 25]. Here, the generator network G simulates an S2 image at time t_2 from S1 and S2 data at t_1 (denoted by $x_{t_1}^{S1}$ and $x_{t_1}^{S2}$, respectively) and S1 data at t_2 (denoted by $x_{t_2}^{S1}$). The output image of the generator is denoted by $G\left(x_{t_1}^{S1}, x_{t_2}^{S1}, x_{t_1}^{S2}\right)$. The discriminator network takes a pair (x, y) of optical images of the same geographic area at time t_1 and t_2, respectively, and estimates the probability $D(x, y)$ that both x and y images are real S2 data and the probability $1 - D(x, y)$ that y comes from the generator. Training consists in successively and alternatively adapting the weights of the discriminator D to improve its classification performance, and then the weights of the generator G so that its output is misclassified by the discriminator. This is achieved by considering the following loss function:

$$L_{cGAN}(G, D) = E \log\left(D\left(x_{t_1}^{S2}, x_{t_2}^{S2}\right)\right) + E \log\left(1 - D\left(x_{t_1}^{S2}, G\left(x_{t_1}^{S1}, x_{t_2}^{S1}, x_{t_1}^{S2}\right)\right)\right) \quad (1)$$

where E denotes the expectation (average value), and $x_{t_j}^{Si}$ $(i, j \in \{1, 2\})$ are the images from Sentinel Si at time t_j of a given location.

Training consists in alternating the following steps.

- Sample a batch of image 4-tuples $(x_{t_1}^{S1}, x_{t_2}^{S1}, x_{t_1}^{S2}, x_{t_2}^{S2})$ from the training dataset, that is, Sentinel-1 (S1) and Sentinel-2 (S2) image pairs of the very same location at time t_1 and t_2.
- For each location, generate a S2 image at t_2 from the S2 image at t_1 and S1 images at t_1 and t_2, that is, compute $G\left(x_{t_1}^{S1}, x_{t_2}^{S1}, x_{t_1}^{S2}\right)$.

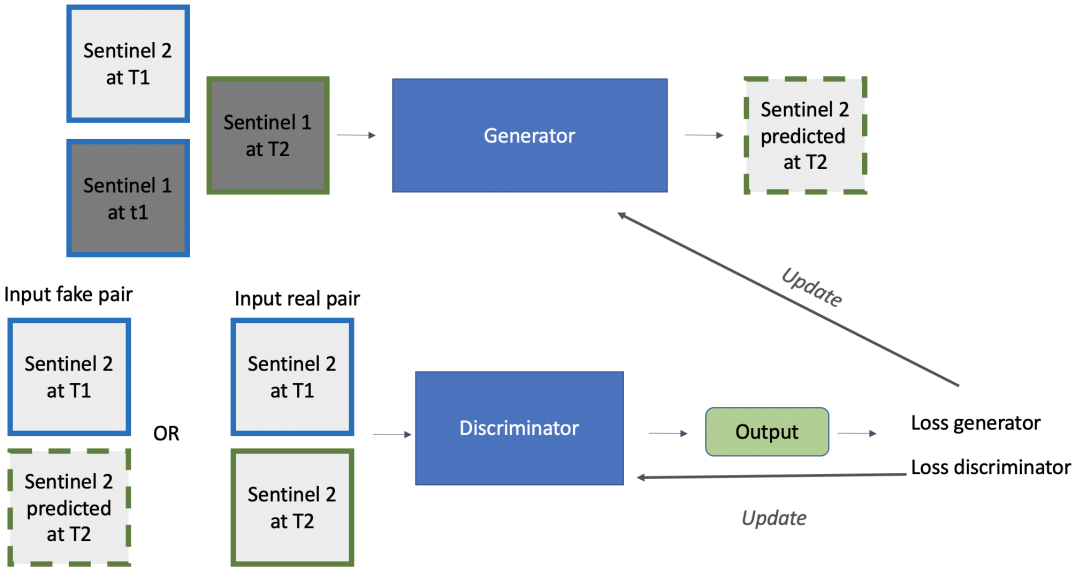

Figure 3: Description of the interactions between the discriminator and the generator during the training stage.

- Train the discriminator by modifying its weights in order to maximize $L_{cGAN}(G, D)$, so that its classification performance improves.
- Train the generator by modifying the weights of G in order to minimize the loss function, so that the ability of the generator to fool the discriminator improves. This is simply achieved by minimizing

$$E \log \left(1 - D\left(x_{t_1}^{S2}, G\left(x_{t_1}^{S1}, x_{t_2}^{S1}, x_{t_1}^{S2}\right)\right)\right)$$

  with respect to the weights of G, since the loss function only depends on this latter term in this case.

After training, the generator is therefore able to produce an optical image at t_2 that is likely indistinguishable from a real one, and hopefully makes use of the correlation between S1 and S2 data at time t_1 to output an optical image from S1 data at time t_2

It is a common practice to add to L_{cGAN} a so-called L1 loss to avoid reconstruction artifacts and to obtain sharper images from the generator. The considered loss function is thus actually given by:

$$L_{cGAN}(G, D) + \lambda E \left\| x_{t_2}^{S2} - G\left(x_{t_1}^{S1}, x_{t_2}^{S1}, x_{t_1}^{S2}\right) \right\|_1 \tag{2}$$

where $\|\cdot\|_1$ is the L1 norm, and $\lambda > 0$ is an hyperparameter of the model.

Figure 3 describes the cGAN architecture. Although this architecture is close to the one proposed in [10], we shortly describe the generator and the discriminator in the following sections for the sake of completeness. Figure 4 gives a comprehensive overview of the model.

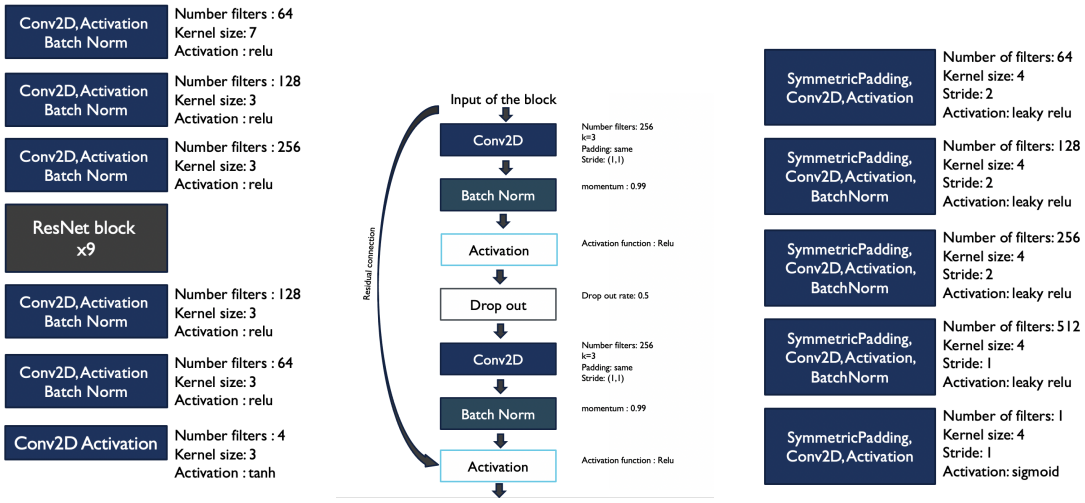

Figure 4: Conditional generative adversarial model. From left to right: generator architecture, ResNet block, and discriminator architecture.

2.2.1 Generator

The generator is a convolutional neural network which takes as input the stack of S1 and S2 data at time t_1 and S1 data at time t_2, and generates an optical image. Since S2 data have four bands (R,G,B,NIR) and S1 data have two channels (VV,VH), the size of the input is $256 \times 256 \times 8$ pixels and the size of the output is $256 \times 256 \times 4$ pixels. The generator is mainly composed of the succession of nine ResNet blocks. A ResNet block is made up of the succession of convolution, batch normalisation, activation, and dropout layers, with a residual connection. Residual connections are known to increase the performance of the neural network and to shorten the training time, see [9]. Before entering the ResNet blocks, the input goes through three Convolution - Activation - BatchNorm modules. After ResNets, three convolution layers are used to progressively reduce the number of channels to four. Zero-padding is used in each convolution layer to keep the 256×256 dimension of the input tile.

2.2.2 Discriminator

The discriminator is a convolutional neural network admitting a stack of two 4-channel images as input and giving the probability that it corresponds to a true pair of S2 images as output. We use a PatchGAN discriminator [12] with a 30×30 array as output. Compared to a traditional image classifier, PatchGAN has fewer parameters and a faster computation time; it has been also proved to make the generator give sharper image in cGAN applications.

2.2.3 Changes to [10] and hyper-parameter setting

The dropout rate in the generator is fixed to 0.5 as in [10]. Following the guidelines of [18], the last activation function is the hyperbolic tangent function which ensures that the generated data is bounded, like real optical images. In order to avoid gradient saturation during training caused by minimizing $\log\left(1 - D\left(x_{t_1}^{S2}, G\left(x_{t_1}^{S1}, x_{t_2}^{S1}, x_{t_1}^{S2}\right)\right)\right)$, we maximize $\log\left(D\left(x_{t_1}^{S2}, G\left(x_{t_1}^{S1}, x_{t_2}^{S1}, x_{t_1}^{S2}\right)\right)\right)$ as advised in [7].

Concerning the discriminator, the standard ReLU activation used in [10] is changed to a Leaky ReLU activation function as recommended in [18], defined by $f(x) = \max(\epsilon x, x)$ with $\epsilon = 0.2$ in our implementation. In addition, we also make use of symmetric padding instead of zero-padding to prevent spurious artifacts on the image borders.

Loss optimization is performed with Adam optimizer with a constant learning rate equal to 10^{-4}. Batch normalization (BN) with momentum equal to 0.99 was used to normalize inputs to zero mean and unit variance. It solves both poor initialization issues and helps the gradient propagation through the layers. BN is not applied to the output layer of the generator and the input layer of the discriminator to avoid numerical instabilities [18], as in [10]. The weight λ of the L_1 loss in L_{cGAN} is set to 100. Learning is performed with a batch size of 8. The cGAN model was trained for 690 epochs (instead of 200 epochs in [10]) during 19 hours on four Nvidia RTX2080Ti GPUs on a Grid5000 cluster[3].

3 Results and discussion

3.1 Assessment method

The generated optical images are assessed by comparing them with the corresponding S2 tile at time t_2 which plays the role of ground-truth data. To this end, we use metrics commonly used in the remote sensing literature [6, 10] and also with a metric dedicated to the task of interest, namely monitoring of vegetation changes.

[3] `https://www.grid5000.fr/w/Nancy:Hardware`

3.1.1 Common metrics

Three common similarity indices are computed: peak signal to noise ratio (PSNR), structural similarity measure (SSIM) [22], and mean spectral angle (MSA). The mean spectral angle is a popular method to compare multiband spectral images [21]. It consists in computing the average over the whole field of the spectral angle θ at each pixel, defined by:

$$\theta = \arccos\left(\frac{\sum_{i=1}^{4} b_i^{\mathrm{GEN}} b_i^{\mathrm{GT}}}{\|b^{\mathrm{GEN}}\|_2 \, \|b^{\mathrm{GT}}\|_2}\right) \tag{3}$$

where the four bands of the generated and ground truth optical images are given at each pixel by $(b_i^{\mathrm{GEN}})_{1\leq i\leq 4}$ and $(b_i^{\mathrm{RT}})_{1\leq i\leq 4}$ respectively, and $||\cdot||_2$ is the Euclidean norm. While similar images give large values for PSNR and SSIM, they give low values for MSA.

3.1.2 Specifying the metrics on vegetation changes

In order to assess the proposed model in near-real-time land monitoring, we estimate the preceding metrics in areas affected by changes (either fire, seasonal variations, or other changes as human interference) between times t_1 and t_2. To this end, we use the unsupervised change detection algorithm described in [2]. It consists in three steps: first, computing the difference between corresponding image patches in two images; second, reducing the dimensionality of these differences by PCA (in the present application, the PCA matrix is computed over all tiles of the training dataset to make it more robust) to obtain a feature vector; and third, classify a pixel as "changed" or "unchanged" depending on the nearest centroid defined by K-means ($K = 2$) when clustering the training feature vectors.

This procedure is applied independently on each (R,G,B,NIR) band of the optical images. Each band has therefore its own change map.

3.2 Metrics over the datasets

The values of PSNR, SSIM and MSA between ground-truth data and generated data are shown in Table 2.

metric name	Datasets		
	train	validation	test
PSNR	41.8	40.4	41.8
SSIM	0.983	0.975	0.982
MSA in rad	0.054	0.067	0.0548

Table 2: Similarity indices

First, comparing these values between the train and test/validation datasets shows that training does not give overfitting, since the similarity indices are within the same

range. Second, although it is difficult to compare results across papers because datasets are different and data pre-processing is not always clearly described, it should be noted that SSIM values are consistent with the results mentioned in [10] with a similar cGAN approach. While we obtain an SSIM value of 0.98 on our test dataset, the best SSIM value in [10] is 0.95. Concerning the MSA value, we have a value in the range of values obtained in [6]. We obtain an MSA value on the test dataset at 0.0548 radian or 3.13 degrees; the MSA is between 3.12 and 5 degrees in [6].

3.3 Output visualization

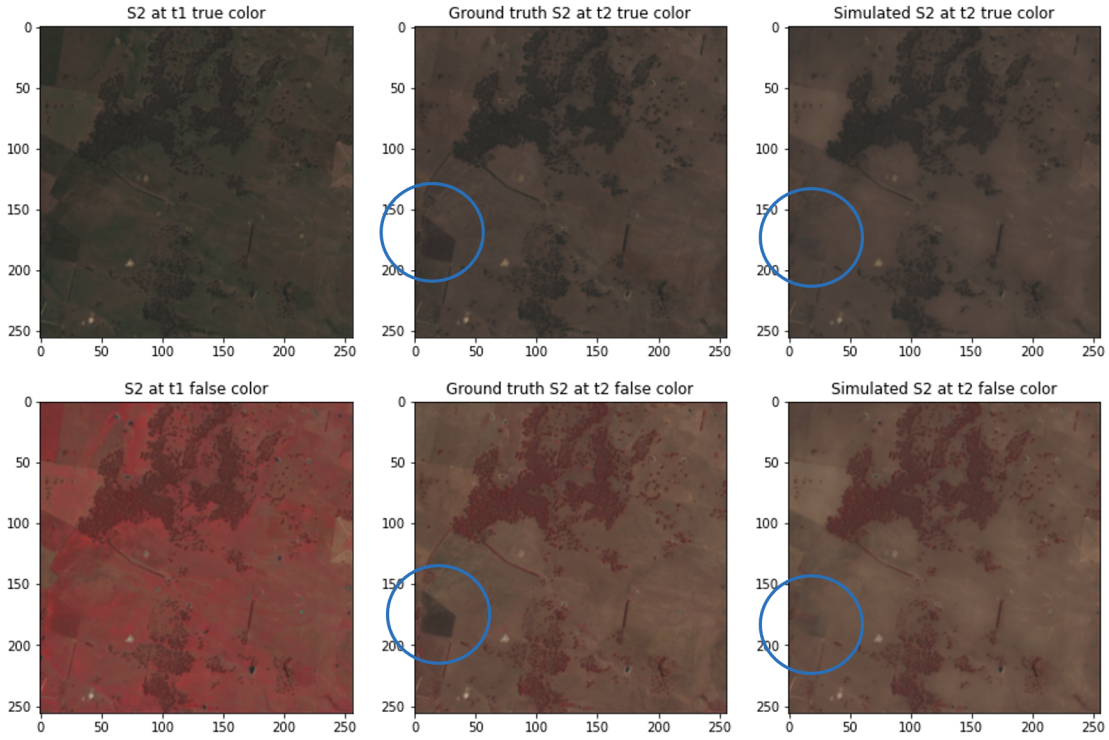

Figure 5: From the left to the right: S2 image at time t_1, ground-truth at t_2, and GAN-simulated tile at t_2, both in true (first row) and false (second row) colors. We can see that most vegetation changes are correctly reproduced, in the sense that the simulated image shows details that are present in ground-truth data but not in t_1 data. However, some parts are not correctly reproduced, as in the area circled in blue.

Figure 5 shows, for one typical tile from the test dataset, the input S2 data, the output of the generator, and the ground-truth data to be compared with the output. We can see

that the generated tile has the same general aspect as the ground truth tile. While some changes are correctly generated, it turns out that all changes are not mapped correctly. For instance, the area circled in blue shows a field with higher NIR band values in the generated tile than in the ground truth one. However, the other part of the image correctly reproduce vegetation changes. This representative example illustrates that the cGAN approach does not only transfer the style (i.e., the overall aspect) of the SAR data to optical bands, but also reproduces details that are present in the SAR band at time t_2.

3.4 Vegetation changes for land monitoring

In this section, we focus on changes detected by the unsupervised clustering algorithm of Section 3.1.2.

3.4.1 Visual assessment

Figure 6 shows, for each of the optical bands of a representative tile, the images at time t_1, at time t_2 (ground truth), the output of the generator, and the map of the pixels marked as "changed" superposed to the ground truth. We can see that most changes are detected in the upper right corner of the tile, which indeed seems consistent. We can also notice that intensity changes over the whole image domain do not mark all pixels as "changed": the procedure is robust to these intensity variations.

3.4.2 Metric-based assessment

We now discuss PSNR results by bands on changed and unchanged pixels. MSA and SSIM are not suited for this task as MSA is a band-wise average and SSIM is not adapted to non-rectangular areas such as changed and unchanged areas. We compute, for each band, both the PSNR of the difference between the ground truth (considered band at time t_2) and the generated image, denoted by PSNR(I2,G), and the PSNR of the difference between the considered band and at time t_1 and the generated image, denoted by PSNR(I1,I2). The higher the PSNR, the more similar the two images are. All PSNR are averaged over train and test datasets, in order to verify the absence of overfitting.

As a reference, Table 3 gives results for the whole image domains (all pixels are taken into account in the differences). Table 4 shows PSNR computed over changed and unchanged pixels.

PSNR computed over train and test datasets are similar, which confirms that the cGAN does not overfit the training data. In all bands, PSNR(I2,G) is larger than PSNR(I1,I2). This means that the generated image at t_2 is closer to the ground truth than to the image at date t_1, change between t_1 and t_2 is therefore well accounted in the generated image.

We can notice that these tables confirm the effectiveness of the change detection algorithm in the R,G,B bands. Except for the NIR band, PSNR(I1,I2) is indeed higher in the unchanged pixels group than in the changed one.

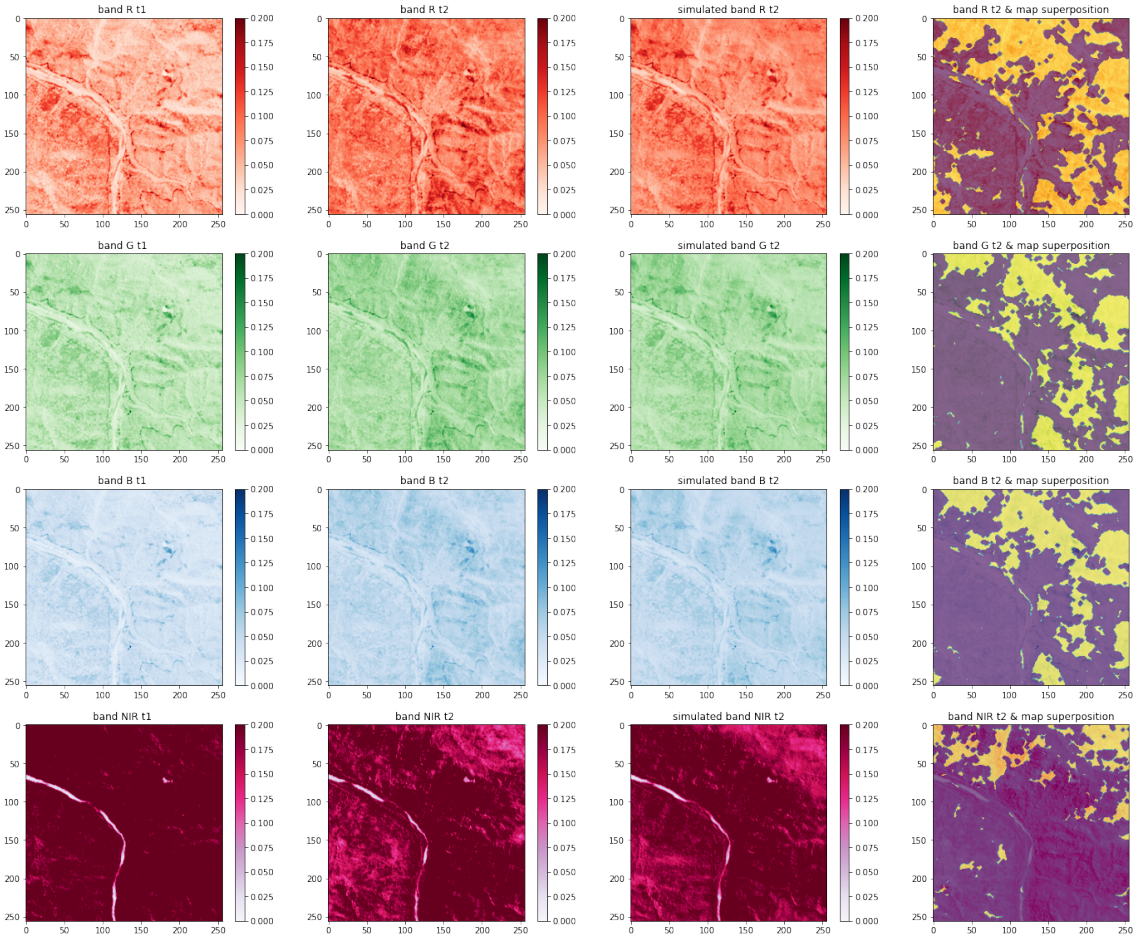

Figure 6: From top to bottom lines: red, green, blue, NIR bands. From left to right: band at time t_1, at time t_2 (ground truth), generated band at t_2, change map superposed with ground truth.

	data	Train	Test
red band	PSNR(I2,G)	39.8	39.9
	PSNR(I1,I2)	22.6	22.6
green band	PSNR(I2,G)	43.2	43.3
	PSNR(I1,I2)	35.7	35.6
blue band	PSNR(I2,G)	43.6	44.6
	PSNR(I1,I2)	38.9	39.0
NIR band	PSNR(I2,G)	34.6	35.0
	PSNR(I1,I2)	19.0	19.1

Table 3: PSNR over the whole image domain (changed and unchanged pixels).

changed pixels	data	Train	Test
red band	PSNR(I2,G)	36.2	36.0
	PSNR(I1,I2)	21.4	21.7
green band	PSNR(I2,G)	38.3	38.2
	PSNR(I1,I2)	32.9	32.9
blue band	PSNR(I2,G)	40.8	40.8
	PSNR(I1,I2)	37.0	37.0
NIR band	PSNR(I2,G)	32.0	31.3
	PSNR(I1,I2)	20.3	20.7

unchanged pixels	data	Train	Test
red band	PSNR(I2,G)	41.0	41.4
	PSNR(I1,I2)	22.9	23.0
green band	PSNR(I2,G)	44.7	44.8
	PSNR(I1,I2)	36.3	36.2
blue band	PSNR(I2,G)	44.5	46.2
	PSNR(I1,I2)	39.5	39.6
NIR band	PSNR(I2,G)	35.6	36.8
	PSNR(I1,I2)	18.8	18.8

Table 4: PSNR averaged over changed pixels (on the left) and unchanged pixels (on the right).

PSNR(I2,G) has a larger value than PSNR(I1,I2) in unchanged pixels because of the effect of intensity variations not detected as changes, as discussed in the illustrative example of Figure 6.

Moreover PSNR(I2,G) for the NIR band is always significantly lower than in the other bands. However, PSNR(I1,I2) is also much lower in NIR than in other bands. This means that NIR is affected by a strong variation between t_1 and t_2. The reason is simply that the region of interest shows significant vegetation changes between these two dates. Besides, it can be noted that NIR band is not standardized with the same process than RGB bands, as explained in Section 2.1. Both temporal variation and standardization method may explain this lower value of PSNR(I2,G).

4 Conclusion

This study was dedicated to a multi-temporal conditional generative adversarial network able to render optical images at a given time, based on SAR data at this time and both SAR data and optical images at an earlier time. Metrics-based and visual assessments show that

this approach permits to render optical images that correctly map the vegetation changes, which is of great interest for near-real-time monitoring of agricultural crop areas and forests, especially after fire events.

Future works include the use of SAR and optical information at more than two dates in order to enhance the generalization ability of the cGAN, and possibly to generate valuable images at times not included in the learning database. Using 3D convolution seems also a promising improvement, as pointed out in [13].

Acknowledgments Experiments presented in this report were carried out using the Grid'5000 testbed, supported by a scientific interest group hosted by Inria and including CNRS, RENATER and several Universities as well as other organizations (see `https://www.grid5000.fr`).

References

[1] M.A. Belenguer-Plomer, M.A. Tanase, A. Fernandez-Carrillo, and E. Chuvieco. Burned area detection and mapping using Sentinel-1 backscatter coefficient and thermal anomalies. *Remote Sensing of Environment*, 233:111345, 2019.

[2] T. Celik. Unsupervised change detection in satellite images using principal component analysis and k-means clustering. *IEEE Geoscience and Remote Sensing Letters*, 6(4):772–776, 2009.

[3] Y. Chen, T.R. McVicar, R.J. Donohue, N. Garg, F. Waldner, N. Ota, L. Li, and R. Lawes. To blend or not to blend? A framework for nationwide Landsat–MODIS data selection for crop yield prediction. *Remote Sensing*, 12(10):1653, 2020.

[4] I.V. Emelyanova, T.R. McVicar, T.G. Van Niel, L.T. Li, and A.I.J.M. van Dijk. Assessing the accuracy of blending Landsat–MODIS surface reflectances in two landscapes with contrasting spatial and temporal dynamics: A framework for algorithm selection. *Remote Sensing of Environment*, 133:193–209, 2013.

[5] F. Gao, J. Masek, M. Schwaller, and F. Hall. On the blending of the Landsat and MODIS surface reflectance: predicting daily landsat surface reflectance. *IEEE Transactions on Geoscience and Remote Sensing*, 44(8):2207–2218, 2006.

[6] J. Gao, Q. Yuan, J. Li, H. Zhang, and X. Su. Cloud removal with fusion of high resolution optical and SAR images using generative adversarial networks. *Remote Sensing*, 12:191, 2020.

[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*, volume 27, 2014.

[8] C. Grohnfeldt, M. Schmitt, and X. Zhu. A conditional generative adversarial network to fuse sar and multispectral optical data for cloud removal from Sentinel-2 images. In *Proc. of IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, pages 1726–1729, 2018.

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

[10] W. He and N. Yokoya. Multi-temporal Sentinel-1 and -2 data fusion for optical image simulation. *ISPRS International Journal of Geo-Information*, 7(10):389, 2018.

[11] T. Hilker, M.A. Wulder, N.C. Coops, N. Seitz, J.C. White, F. Gao, J.G. Masek, and G. Stenhouse. Generation of dense time series synthetic Landsat data through data blending with MODIS using a spatial and temporal adaptive reflectance fusion model. *Remote Sensing of Environment*, 113(9):1988–1999, 2009.

[12] P. Isola, J.-Y. Zhu, T. Zhou, and A. Efros. Image-to-image translation with conditional adversarial networks. In *Proc. of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[13] S. Ji, Z. Chi, A. Xu, and Y. Duan. 3D convolutional neural networks for crop classification with multi-temporal remote sensing images. *Remote Sensing*, 10:75, 2018.

[14] S. Khabbazan, P. Vermunt, S. Steele-Dunne, L. Ratering Arntz, C. Marinetti, D. van der Valk, L. Iannini, R. Molijn, K. Westerdijk, and C. van der Sande. Crop monitoring using Sentinel-1 data: A case study from the Netherlands. *Remote Sensing*, 11(16):1887, 2019.

[15] L. Liu and B. Lei. Can SAR images and optical images transfer with each other? In *Proc. of IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, pages 7019–7022, 2018.

[16] A. Meraner, P. Ebel, X.X. Zhu, and M. Schmitt. Cloud removal in Sentinel-2 imagery using a deep residual neural network and SAR-optical data fusion. *ISPRS Journal of Photogrammetry and Remote Sensing*, 166:333–346, 2020.

[17] L. Poggio, A. Gimona, and I. Brown. Spatio-temporal MODIS EVI gap filling under cloud cover: An example in Scotland. *ISPRS Journal of Photogrammetry and Remote Sensing*, 72:56–72, 2012.

[18] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv:1511.06434 [cs]*, 2016.

[19] R. A. Schowengerdt. *Remote sensing: models and methods for image processing*. 2006.

[20] P. Singh and N. Komodakis. Cloud-Gan: Cloud removal for sentinel-2 imagery using a cyclic consistent generative adversarial networks. In *Proc. of IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, pages 1772–1775, 2018.

[21] Y. Sohn and R.M. McCoy. Mapping desert shrub rangeland using spectral unmixing and modeling spectral mixtures with TM data. *Photogrammetric Engineering and Remote Sensing*, 63(6):707–716, 1997.

[22] Z. Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.

[23] Q. Weng, P. Fu, and F. Gao. Generating daily land surface temperature at landsat resolution by fusing Landsat and MODIS data. *Remote Sensing of Environment*, 145:55–67, 2014.

[24] Q. Xiong, L. Di, Q. Feng, D. Liu, W. Liu, X. Zan, L. Zhang, D. Zhu, Z. Liu, X. Yao, and X. Zhang. Deriving Non-Cloud Contaminated Sentinel-2 Images with RGB and Near-Infrared Bands from Sentinel-1 Images Based on a Conditional Generative Adversarial Network. *Remote Sensing*, 13(8):1512, 2021.

[25] J. Zhu, T. Park, P. Isola, and A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proc. of the International Conference on Computer Vision (ICCV)*, pages 2242–2251, 2017.

A Appendix

This appendix gives additional illustrations of the rendering of S2 data in several areas.

In Figure 7, false color representation shows that there is a significant decrease of the intensity of the NIR band between date 1 and 2, especially in the bottom of the image. The resulting false color image is indeed darker at t_2 than at t_1. We can see that this decrease is properly reflected in the simulated false color image. However, the simulation of the NIR band shows small differences with ground truth, especially in the center part of the simulated tile. Indeed, this region shows low amplitude variations of the NIR bands that are not retrieved by simulation. For example, the GAN overestimate the NIR band values in the hillsides.

The multitemporal GAN approach is able to recreate major changes in different kinds of landscapes, either croplands (Figures 9, 11, 5) or areas not affected by human activites (Figures 7, 13). This is confirmed by a band-by-band comparison between the ground

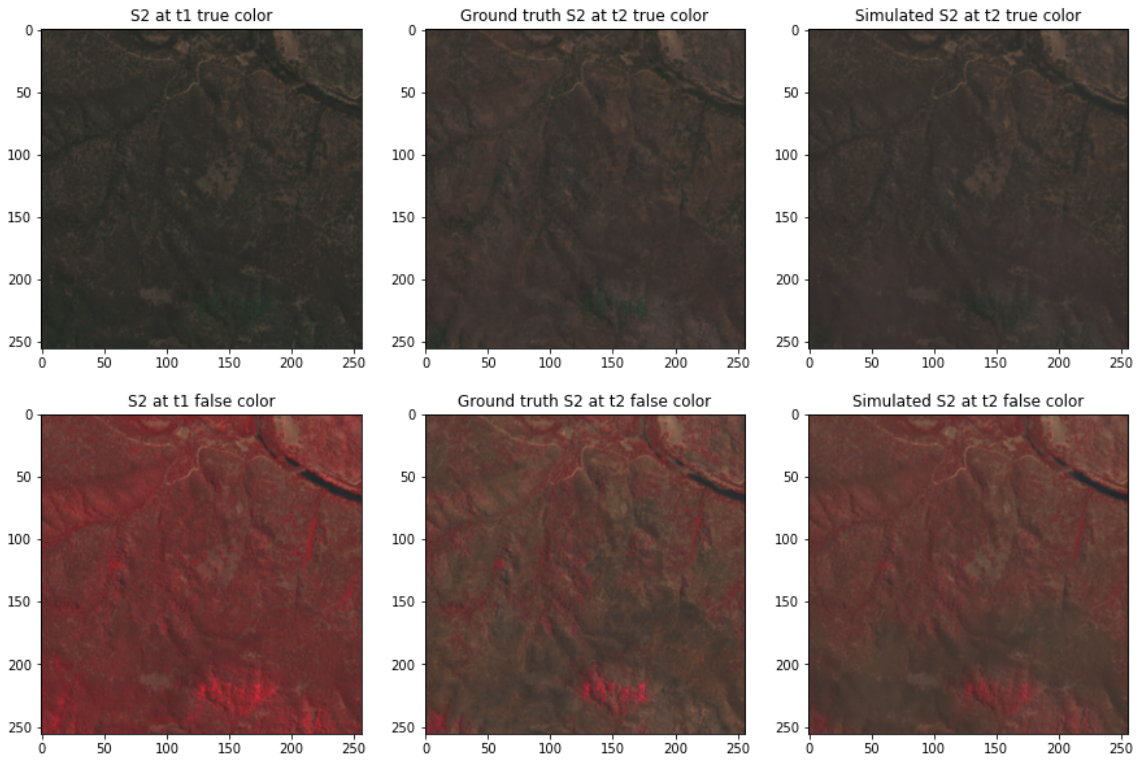

Figure 7: Comparison of the ground truth tile (S2 at t2) and the simulated tile in true and false color.

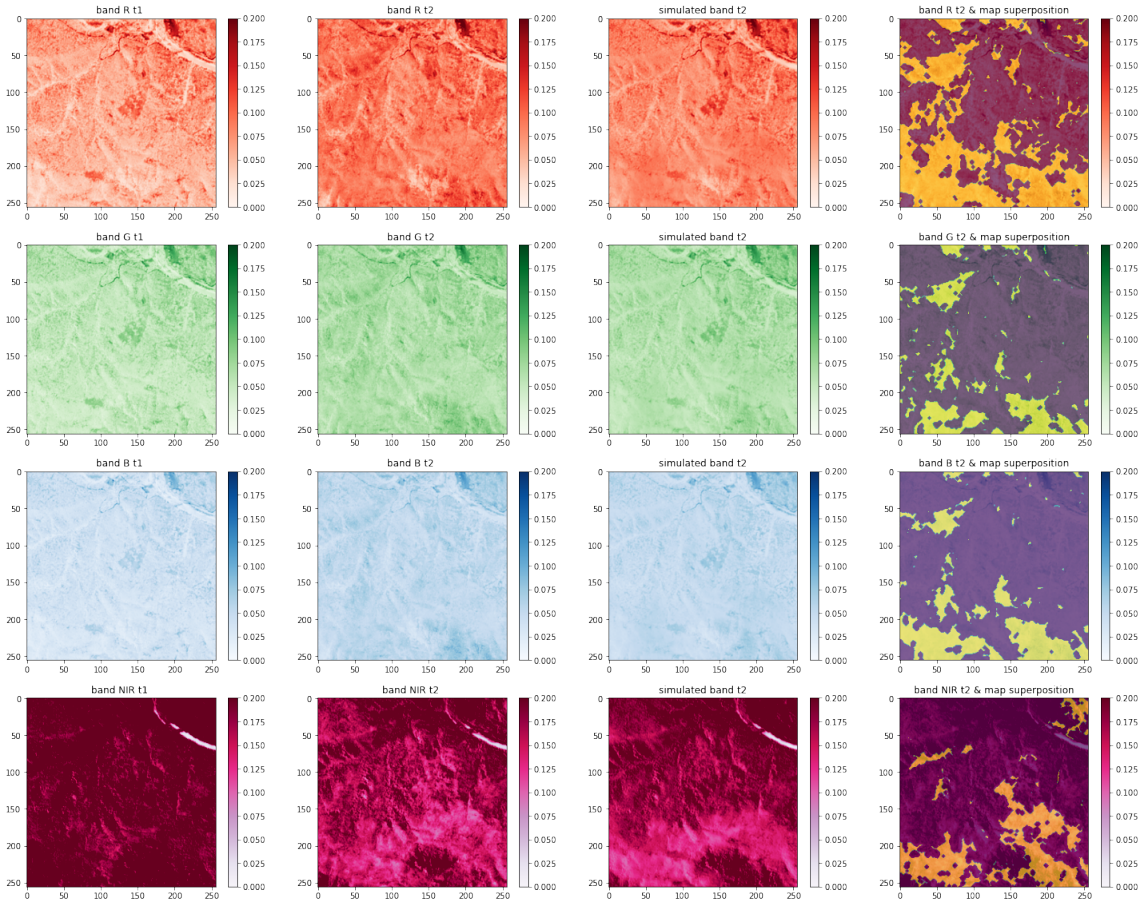

Figure 8: From top to bottom lines: red, green, blue, NIR bands. From left to right: band at time t_1, at time t_2 (ground truth), generated band at t_2, change map superposed with ground truth.

truth and the simulated tile (Figures 8, 10, 12, 14, 15). However, some differences can be noted between the simulated red band and the ground truth one. For example in Figure 8, the simulated red band values are not as large as expected from the ground truth images, which is reflected by a slightly clearer simulated red band than the ground truth red band. On the contrary, Figure 12 shows some area with a larger value of the red band than expected. Visually, we notice wide dark areas in the simulated tile on both sides of a thin structure which likely corresponds to a wooded path. These remarks are in accordance with the results in Table 3 where the PSNR values between ground truth and simulation PNSR(I2,G) for red band and NIR band are lower than the blue and green ones.

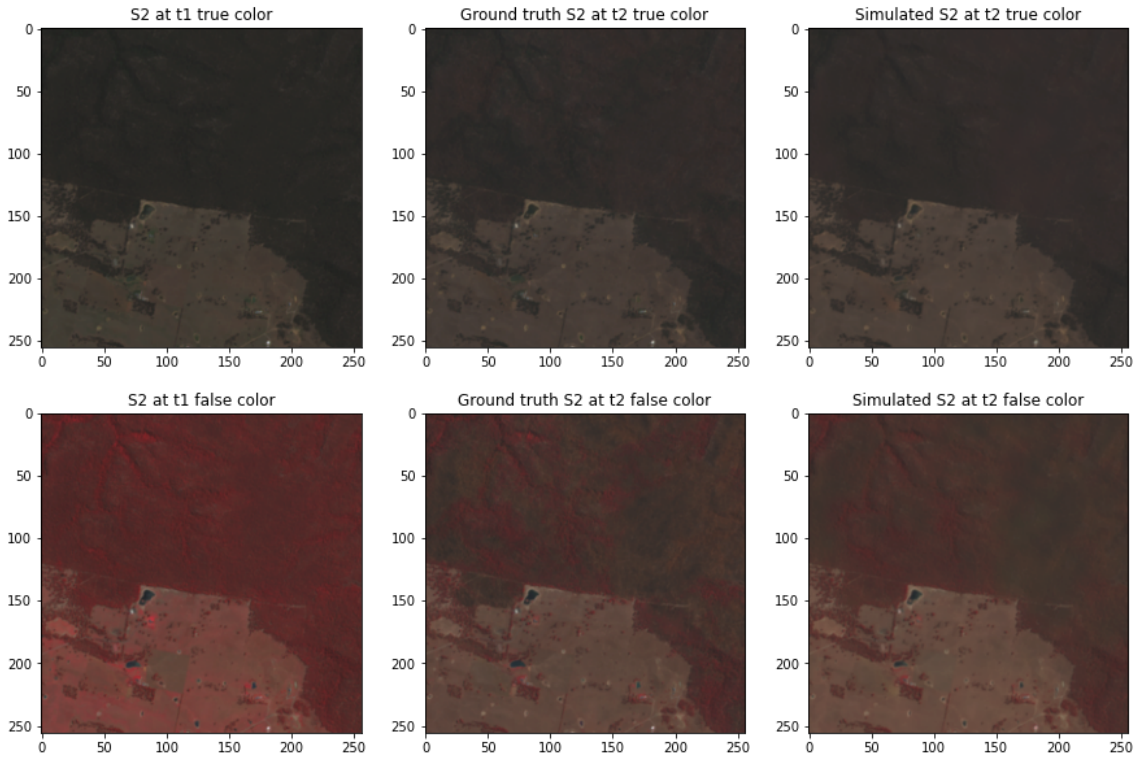

Figure 9: Comparison of the ground truth tile (S2 at t2) and the simulated tile in true and false color.

In Figures 9 and 11, we see that the simulated tile are slightly blurred compared to the ground truth. However as in Figure 7, significant variations in the NIR band are well recreated by the GAN.

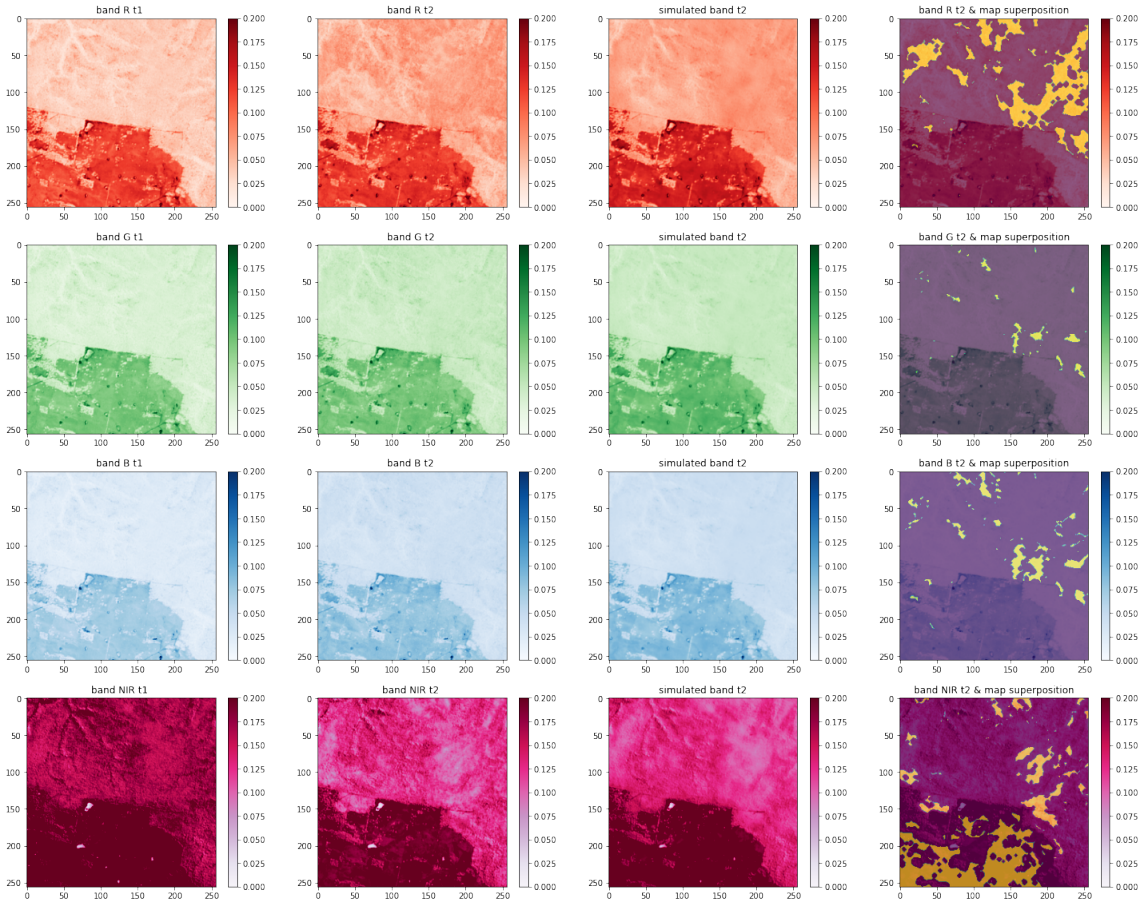

Figure 10: From top to bottom lines: red, green, blue, NIR bands. From left to right: band at time t_1, at time t_2 (ground truth), generated band at t_2, change map superposed with ground truth.

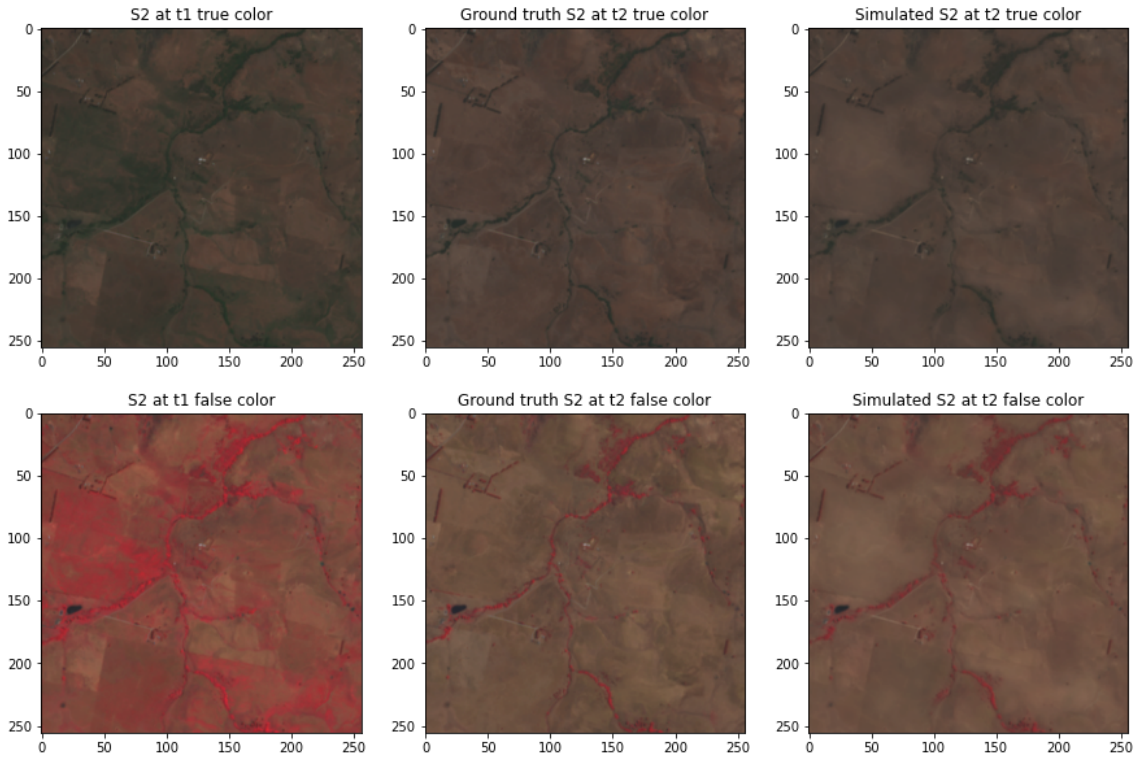

Figure 11: Comparison of the ground truth tile (S2 at t2) and the simulated tile in true and false color.

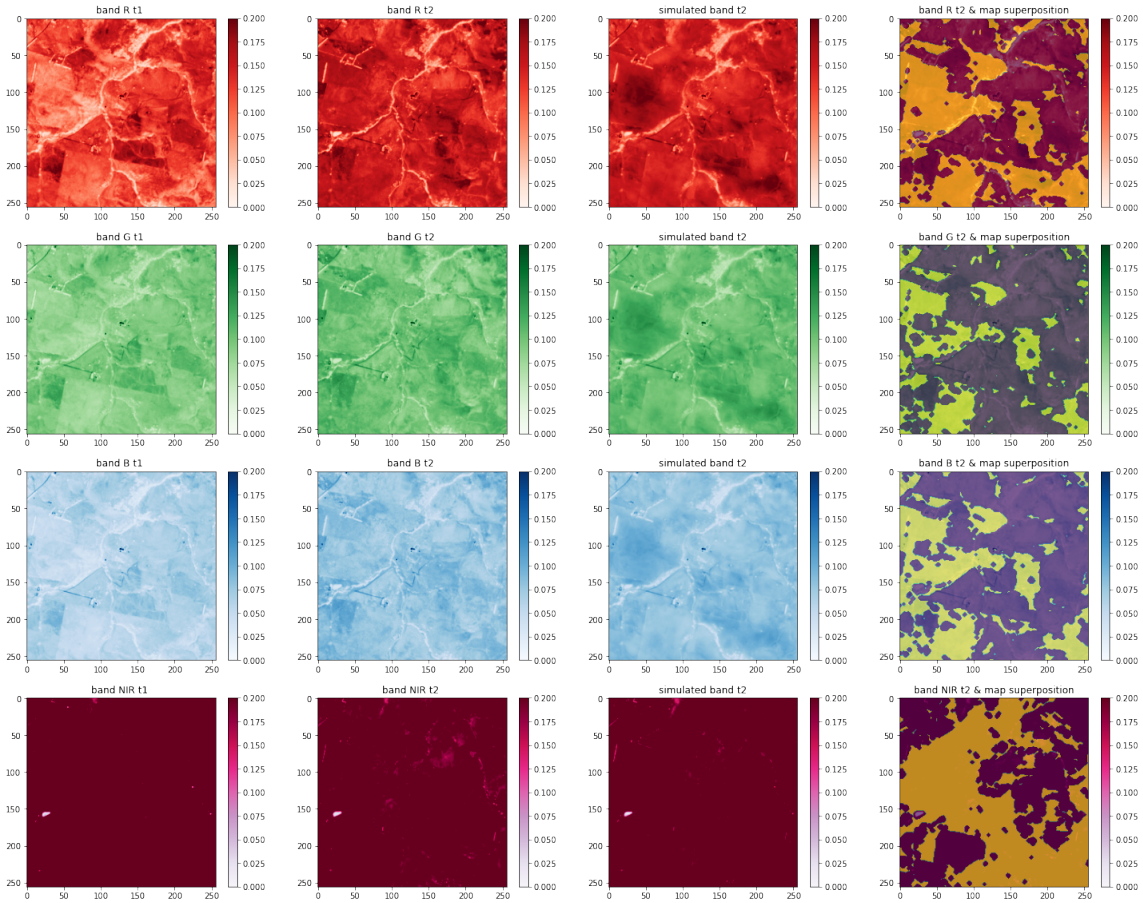

Figure 12: From top to bottom lines: red, green, blue, NIR bands. From left to right: band at time t_1, at time t_2 (ground truth), generated band at t_2, change map superposed with ground truth.

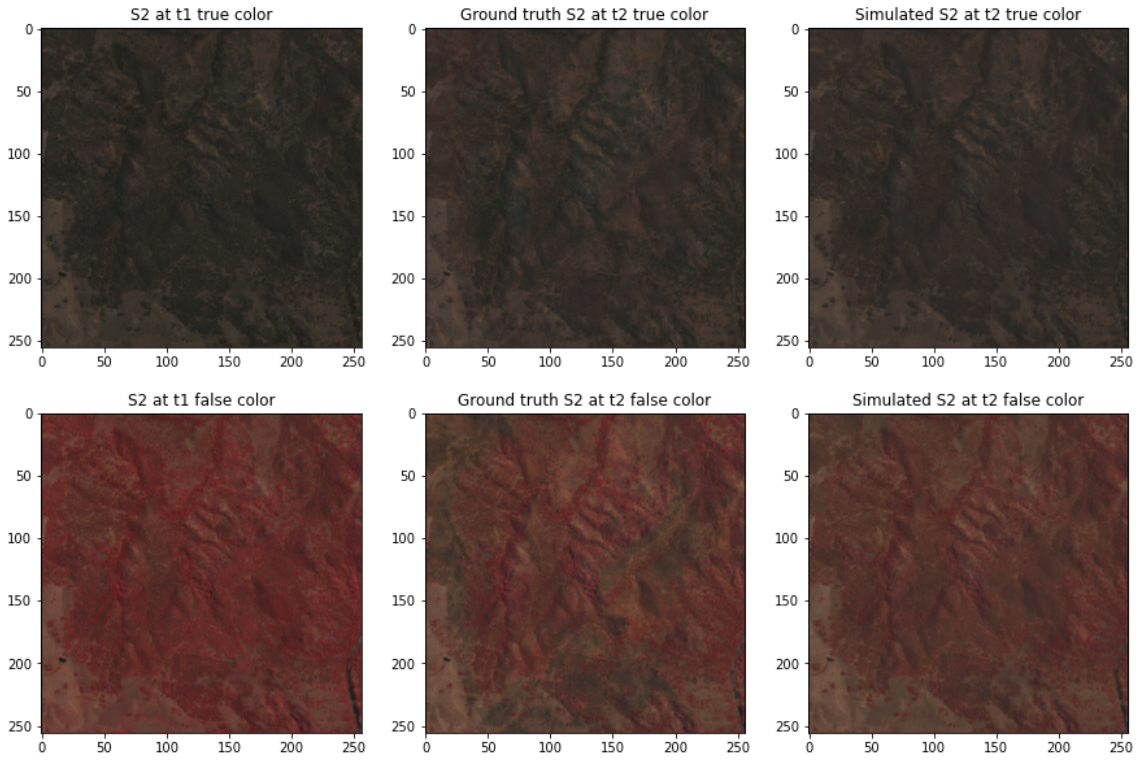

Figure 13: Comparison of the ground truth tile (S2 at t2) and the simulated tile in true and false color.

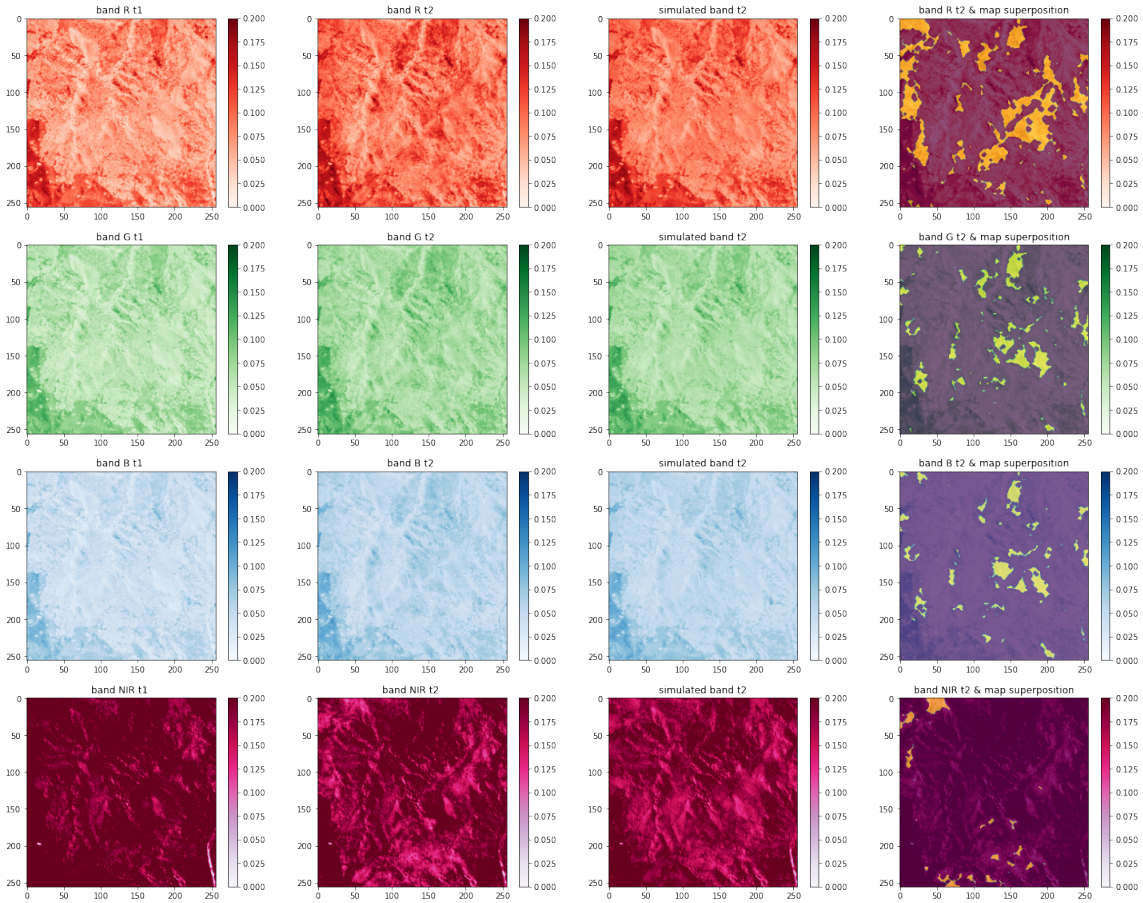

Figure 14: From top to bottom lines: red, green, blue, NIR bands. From left to right: band at time t_1, at time t_2 (ground truth), generated band at t_2, change map superposed with ground truth.

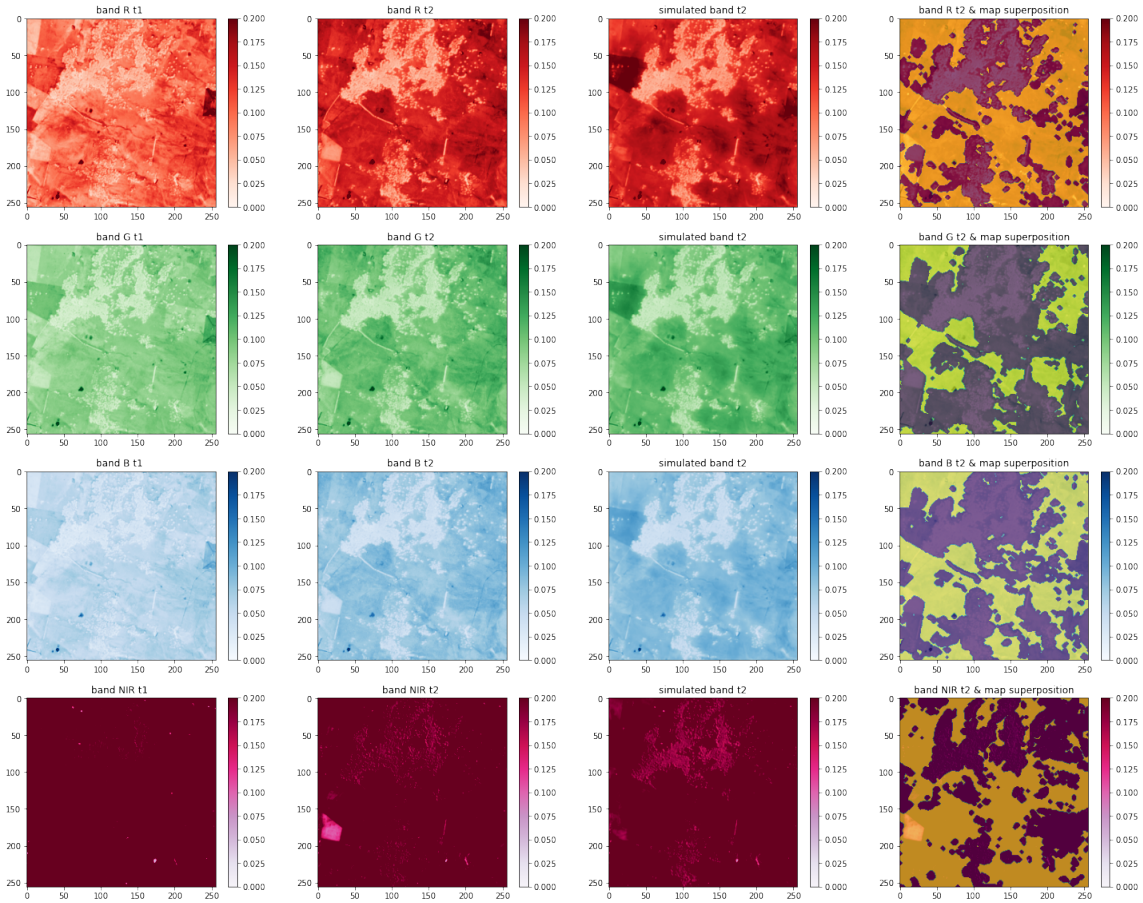

Figure 15: From top to bottom lines: red, green, blue, NIR bands. From left to right: band at time t_1, at time t_2 (ground truth), generated band at t_2, change map superposed with ground truth.