

Generation of synthetic generative adversarial network-based multispectral satellite images with improved sharpness

Lydia Abady *, Mauro Barni , Andrea Garzelli , and Benedetta Tondi 
University of Siena, Department of Information Engineering and Mathematics, Siena, Italy

ABSTRACT. The generation of synthetic multispectral satellite images has not yet reached the quality level achievable in other domains, such as the generation and manipulation of face images. Part of the difficulty stems from the need to generate consistent data across the entire electromagnetic spectrum covered by such images at radiometric resolutions higher than those typically used in multimedia applications. The different spatial resolution of image bands corresponding to different wavelengths poses additional problems, whose main effect is a lack of spatial details in the synthetic images with respect to the original ones. We propose two generative adversarial networks-based architectures explicitly thought to generate synthetic satellite imagery by applying style transfer to 13-band Sentinel-2 level1-C images. To avoid losing the finer spatial details and improve the sharpness of the generated images, we introduce a pansharpening-like approach, whereby the spatial structures of the input image are transferred to the style-transferred images without introducing visible artifacts. The results we got by applying the proposed architectures to transform barren images into vegetation images and vice versa and to transform summer (res. winter) images into winter (res. summer) images, which confirm the validity of the proposed solution.

© The Authors. Published by SPIE under a Creative Commons Attribution 4.0 International License. Distribution or reproduction of this work in whole or in part requires full attribution of the original publication, including its DOI. [DOI: [10.1117/1.JRS.18.014510](https://doi.org/10.1117/1.JRS.18.014510)]

Keywords: generative adversarial network; Sentinel-2; Gram–Schmidt adaptive; sharpening; cycleGAN; pix2pix

Paper 230460G received Aug. 29, 2023; revised Dec. 11, 2023; accepted Jan. 17, 2024; published Feb. 7, 2024.

1 Introduction

The use of artificial intelligence (AI) techniques based on generative adversarial networks (GANs) to generate highly realistic synthetic images is progressing at an extremely rapid pace to match the increasing demand for large labeled datasets for computer vision applications. GAN images are also used in the entertainment industry, on social media, and in a wide variety of web applications. On the other hand, AI-generated images are increasingly used for malevolent purposes, such as defamation and disinformation campaigns.

With the increasing diffusion of satellite images and their exploitation in several application areas, such as meteorological forecasts, monitoring, detection of natural disasters, and intelligence and military investigations, just to mention a few, it is only natural that AI architectures have started being used to generate synthetic satellite images as well. Possible uses of such images include training AI tools for earth observation applications, making predictions about the effects of climate change and natural disasters, and raising awareness about the possible effects of global warming and anthropization of natural environments.^{1,2} Even in this case, malevolent uses are possible to create fake images to deny the effect of climate change or

*Address all correspondence to Lydia Abady, lydia.abady@unisi.it

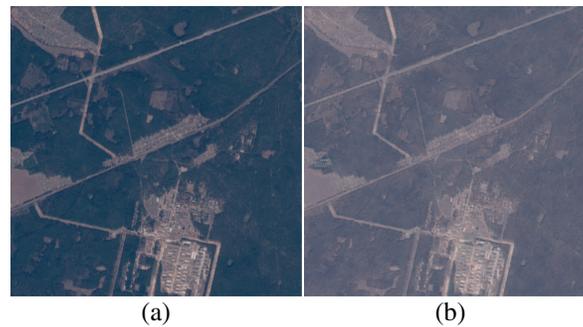


Fig. 1 Example of the sharpness difference between (a) real vegetation and (b) generated barren images, which are more visible in the building area.

artificially augment the impact of human actions on specific regions. Despite this interest, the generation of synthetic multispectral satellite images has not yet reached the quality level achievable in other domains, such as the generation and manipulation of face images. The direct application of GAN architectures used for computer vision applications to generate synthetic remote sensing images is not possible, due to the inherent differences between remote sensing images and natural images, which GANs were specifically designed for.

To start with, optical remote sensing images are typically multispectral images consisting of more than 3 bands and have a pixel depth ranging from 10 to 16 bits, making their generation more difficult than in the case of natural images. In addition, the different spatial resolution of image bands corresponding to different wavelengths poses additional problems, whose main effect is a lack of spatial details in the synthetic images with respect to the original ones. Such an effect is exemplified in Fig. 1, where the difference between the sharpness and richness of details of real and synthetic images is evident. The deficiency in details is particularly obvious in the vicinity of the buildings, where the real image exhibits significantly more intricate details.

In this paper, we propose methods to generate synthetic multispectral remote sensing images, overcoming the difficulties mentioned above.

We focus on Sentinel-2 level-1C image modification using all 13 bands to provide comprehensive spectral information. It is important to understand the differences between the various Sentinel-2 product levels. Level-1B data typically comprises radiometrically corrected top-of-atmosphere reflectance values; level-1C products comprise orthorectified images with a uniform grid; and level-2A products offer atmospherically corrected bottom-of-atmosphere reflectance values. The focus of our research is on image-to-image translation of Sentinel-2 level-1C images, with the goal of creating a synthetic counterpart of real images with content that differs from the original but is still relevant. In particular, we explore two distinct image translation tasks: land cover transfer and season transfer.

Unlike previous works, which predominantly focus on generating the three RGB bands, with limited exploration into synthesizing 4-band [R, G, B, and near-infrared (NIR)] images,³ our approach delves into the generation of realistic 13-band synthetic images. This introduces additional complexities due to the distinct spatial resolutions across the four bands with a 10-m ground sampling distance (GSD), six bands with a 20-m GSD, and three bands with a 60-m GSD. Consequently, the application of image-to-image translation architectures commonly used in computer vision applications is not possible. To address the resolution disparity among bands, we propose an approach utilizing multiple discriminators, each tasked with the classification of bands of the same spatial resolution. The adversarial loss is then adapted to accommodate the outputs of all discriminators. Moreover, to counteract the lack of sharpness in synthetically generated images, we introduce a postprocessing algorithm inspired by image pansharpening.⁴ Pansharpening⁵ is a widely employed technique for enhancing the spatial resolution of multispectral satellite images by incorporating high-resolution panchromatic data. Although traditional pansharpening relies on a separate high-resolution panchromatic image, in our context, a panchromatic version of the GAN-generated images is not available. Instead, we extract fine spatial details from the original to-be-transferred image and inject them into the generated images, akin to pansharpening. The fundamental idea is that when applying style transfer to

alter the season or cover type of an image, the underlying spatial structures, such as roads, rivers, mountains, and buildings remain unchanged. Therefore, these structures can be extracted from the original source image and used to enhance the sharpness of GAN-generated images. Our algorithm specifically employs sharpening by component substitution, utilizing the Gram–Schmidt adaptive (GSA) algorithm,⁶ where the panchromatic image is estimated from the original source image. In this way, we enhance the spatial details of the synthesized images, contributing to their overall quality and realism.

We carried out an extensive set of experiments applying the proposed image-to-image translation networks and the sharpening algorithm to both season transfer and land-cover transfer. In particular, we assessed the quality of the final sharpened images using the perception-based image quality evaluator (PIQUE), which showed similar or slightly better values for the sharpened images than for real input images. Overall, the main contributions of this paper can be summarized as follows.

- We adapt two common GAN architectures, namely cycleGAN and pix2pix, to make them work on multispectral Sentinel-2 level-1C images with 13 bands and different resolutions.
- We introduce a postprocessing technique based on Gram–Schmidt pansharpening to improve the quality of the images generated by the GANs.
- We apply the proposed architectures and pansharpening postprocessing to address two image translation tasks, namely season transfer and land cover transfer.

The remainder of this paper is organized as follows. In Sec. 2, we review the state-of-the-art of GAN-based image generation, with specific reference to remote sensing imagery. Then in Sec. 3, we give a brief introduction to image-to-image translation GANs with particular emphasis on the network architectures we have used in our work, namely, pix2pix and cycleGAN. In Sec. 4, we describe the datasets, the network architectures, and the training procedure we have used to develop the season transfer and the cover transfer models. After that, in Sec. 5, we describe the sharpening procedure we have developed to improve the quality of the generated images. In Sec. 6, we show the results of the experiments we have carried out to validate the effectiveness of the proposed image transfer networks and the sharpening algorithm. In Sec. 7, we conclude the paper with some final remarks and indications for future research.

2 State of the Art

The quality of the images generated by GANs is continuously increasing. Initially, GANs were used to synthesize images that are similar in distribution to the training dataset.⁷ Over the years, several variants have been developed. Some of them have been thought to improve the shortcomings of earlier architectures and deliver more realistic results,^{8–10} and others have been designed to perform a variety of tasks in computer vision applications. Image-to-image translation,^{11,12} classification,^{13,14} and segmentation¹⁵ are just a few of the many applications of GANs.

It is no surprise that GANs are also exploited in various remote sensing applications. One interesting application is the use of image-to-image translation for the generation of synthetic images produced by a certain sensor given an input image acquired by a different sensor. The method described in Ref. 16, for instance, can generate optical (RGB) images starting from SAR input images. The result is obtained by training a cycleGAN model on 512×512 patches of RGB images as the source domain and 512×512 SAR images as the target domain. Similarly in Refs. 3 and 17, the NIR channel is generated using the RGB bands as input. This result is achieved by training a pix2pix model on paired examples of RGB and NIR images. Another type of image-to-image translation was applied in Ref. 18, where historical maps were translated into satellite-like imagery. Finally, in Ref. 19, SAR and RGB images are synthesized starting from land cover maps coupled with auxiliary satellite data like digital elevation models and precipitation maps.

In addition to generating different types of images, another important application of GANs is super-resolution, aiming at overcoming the lack of resolutions of the capturing sensors. In Ref. 20, a modified denseNet (ultradense) is used in a GAN architecture to generate super-resolution satellite images from low-resolution images. Other tasks for which GANs are used in remote sensing applications are pansharpening²¹ and hyperspectral image classification.²²

With regard to the generation of synthetic multispectral images, in Ref. 23, a progressive GAN⁸ was trained on the SEN12MS dataset to generate from scratch $256 \times 256 \times 13$ images that resemble Sentinel-2 level-1C products. In the same paper, image-to-image translation is considered, by training a NICEGAN²⁴ architecture that is able to transfer vegetation land cover into barren land cover and vice versa for the four high-resolution bands of Sentinel 2 level-1C products (RGB and NIR bands). This work was extended in Ref. 25, to season transfer, generating winter (res. summer) images from summer (winter) ones. Even in this case, the translation is applied to the four highest resolution bands only.

Despite the increasing interest, the use of style transfer to change the semantic content of satellite images is still limited. Furthermore, the few existing works focus on generating RGB images, just adding a NIR band in some rare cases. In contrast, in this work, we focus on the application of style transfer to all 13 bands of Sentinel-2 level-1C images. Compensating for the lack of sharpness of synthetic remote sensing images is also something that has received very limited attention so far. The only previous efforts in this context are related to super-resolution applications. However, it is important to highlight a noticeable difference between super-resolution and the problem addressed in this paper. In super-resolution scenarios, there is a lack of information available about the high-resolution content to be reinserted into the processed images. This contrasts with style-transfer scenarios, where it is possible to exploit the information contained within the source image.

3 Background on GAN-Based Image-to-Image Translation

The general GAN framework for image generation consists of two convolutional neural networks: a generator, which is trained to produce images that are similar in distribution to the images used for training, and a discriminator in charge of classifying images as real (genuine) or fake (synthetically generated). The two networks are trained together in a minimax fashion, with the generator aiming at making the discriminator fail and the generator trying to distinguish genuine from fake images despite the efforts of the generator. The weights of the generator and the discriminator are updated alternately in an iterative way: the discriminator is trained for one or more epochs while the generator is kept constant; afterward, the generator is trained for one or more epochs while the discriminator weights are frozen. Training iterations continue until the two networks converge or satisfactory visual results are reached. As discussed in Sec. 2, many variants of GANs have been proposed depending on the application at hand. In this work, we focus on architectures for image-to-image translation. The goal of image-to-image translation is to take an image belonging to a certain domain, e.g., a daylight street image, and remap it onto a different domain, e.g., the night version of the daylight street image.

In this paper, we rely on two specific architectures for image-to-image translation, namely pix2pix and cycleGAN. The former can be used whenever a dataset of corresponding image pairs belonging to the input and output domains is available for training. The latter can also be used when only unpaired examples of the two domains are available.

3.1 Pix2pix

As we mentioned before, a pix2pix architecture is trained by showing the network examples of input–output pairs, with the output sample corresponding to a ground-truth translated version of the input scene. Figure 2 displays the workflow followed to train a pix2pix model. A generator takes an image x from domain A as input and produces a version of x that corresponds to domain B. On the other hand, the discriminator judges whether for a certain image belonging to domain A (always real), the corresponding image in domain B has been generated synthetically (score 0) or not (score 1). The loss used to train the generator includes two terms. The first one, usually referred to as “adversarial loss” (L_{adv}), is a cross-entropy term given by

$$L_{adv} = E_{x,y}[\log D(x, y)] + E_x[\log(1 - D(x, G(x)))], \quad (1)$$

where x is the input image, y is the reference image, D is the discriminator network, and G is the generative network. The second term (L_1) corresponds to the distance between the ground truth image y and the generated image $G(x)$:

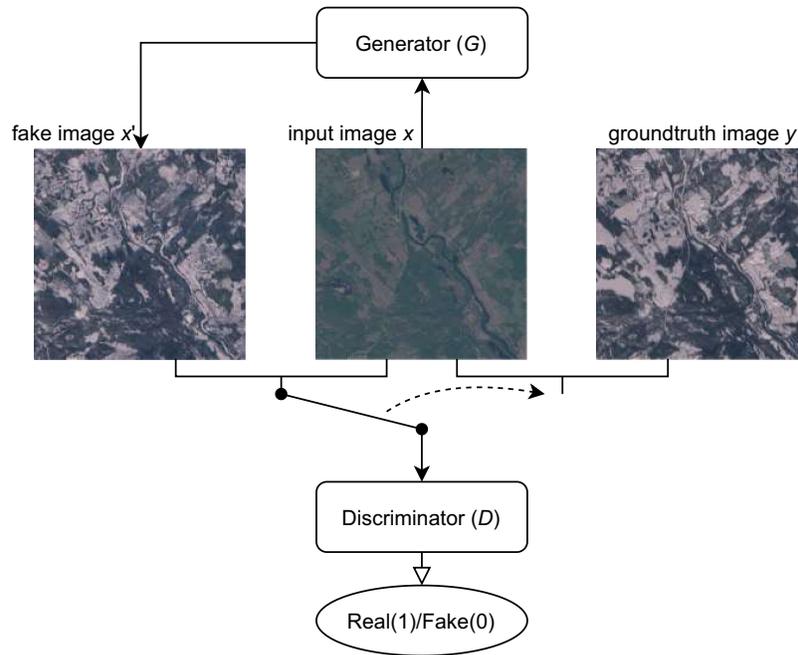


Fig. 2 pix2pix training workflow.

$$L_1 = E_{x,y}[\|y - G(x)\|], \quad (2)$$

The objective of the generator is to minimize the combined loss L_G :

$$\min_G L_G = \min_G (L_{adv} + \lambda L_1), \quad (3)$$

where λ is a parameter balancing the relative importance of the L_{adv} and L_1 losses. During its training turn, the discriminator aims at maximizing L_{adv} , that is labeling real images as 1 (real image label) and the generated ones as 0 (generated image label) by the discriminator.

In this paper, we used the pix2pix architecture for the season transfer task. In such a case, in fact, finding an image in domain B corresponding to a given image in domain A is pretty easy, due to the wide availability of images of the same region taken at different times of the year. This is not the case for the land cover transfer task, for which we had to resort to the cycleGAN network¹² (described in Sec. 3.2).

3.2 CycleGAN

CycleGAN is an image-to-image architecture that does not require the availability of matched image pairs for training, which reduces the difficulty of gathering a proper training dataset. A cycleGAN architecture consists of two generators and two discriminators. Basically, each generator translates images in a unique direction. The generator G_{a2b} translates images from domain A to domain B, whereas the generator G_{b2a} translates the images in the opposite direction. In this way, each generator can act as a constraint for the other. In order to do that, a cycle consistency check is implemented within the architecture. When the output of the first generator is used as input for the second one, the output of the second generator should be as close as possible to the original input image (similarly for the second generator), as shown in Eq. (4). Moreover, an optional identity loss can be added to constrain the cycleGAN architecture. Specifically, the identity loss forces the generator to act as an identity operator when the input image already belongs to the output domain. Figure 3 shows the cycleGAN architecture together with the losses used to train it. The adversarial GAN loss (L_{adv}) measures the capability of the discriminators to distinguish genuine images belonging to a certain domain, from the corresponding synthetic images belonging to the same domain. In our work, we adopted a least square formulation¹¹ of the adversarial loss according to which we have:

$$L_{adv} = E_a[(D_b(G_{a2b}(a)) - 1)^2] + E_b[(D_a(G_{b2a}(b)) - 1)^2], \quad (4)$$

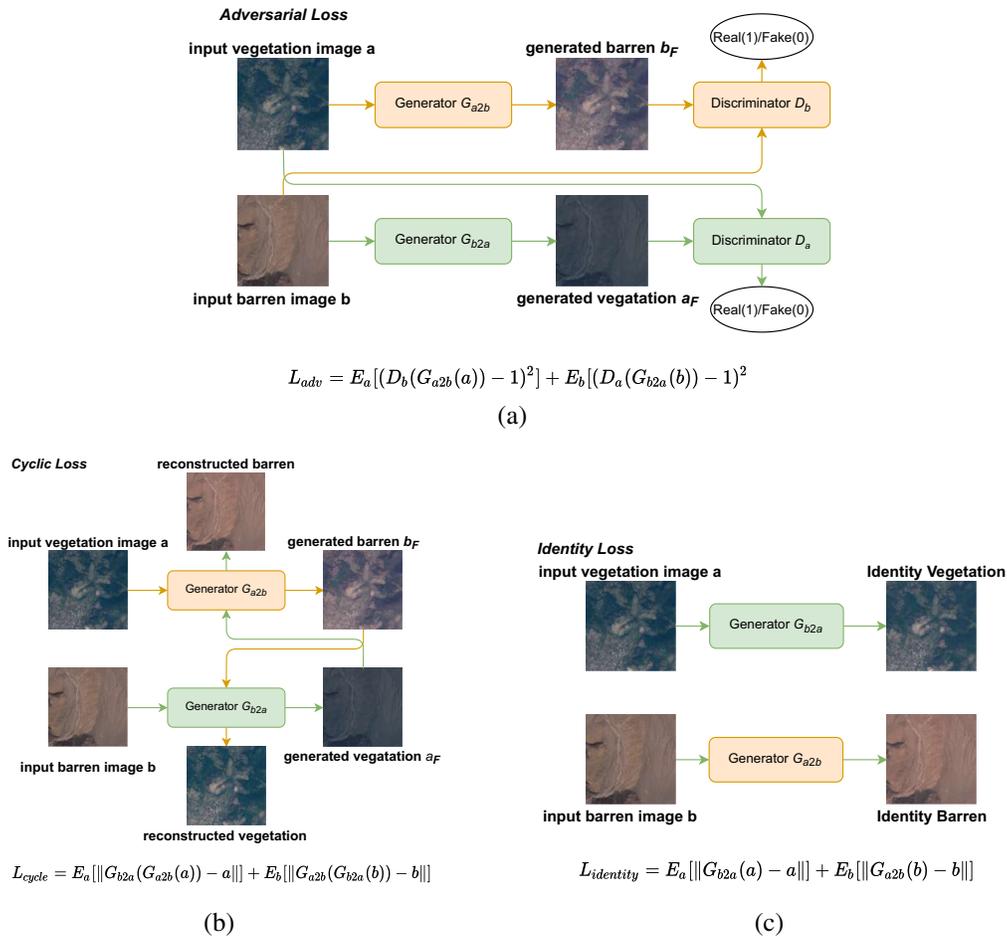


Fig. 3 CycleGAN partial losses with workflow description: (a) adversarial loss, (b) cyclic loss, and (c) identity loss.

where a is an image belonging to domain A , b is an image from domain B , G_{a2b} is the generative network that translates the images from domain A to domain B , G_{b2a} is the generative network that translates the image from domain B to domain A , and D_b is the discriminator network that classifies the images as genuine images belonging to domain B and images generated by G_{a2b} . Similarly, D_a is the discriminator network that distinguishes genuine images of domain A from the images generated by G_{b2a} .

The cyclic consistency loss (L_{cycle}) and the identity loss are defined as follows:

$$L_{cycle} = E_a [\|G_{b2a}(G_{a2b}(a)) - a\|] + E_b [\|G_{a2b}(G_{b2a}(b)) - b\|], \quad (5)$$

$$L_{identity} = E_a [\|G_{b2a}(a) - a\|] + E_b [\|G_{a2b}(b) - b\|]. \quad (6)$$

The objective of the generators is to minimize a global loss defined by

$$\min_{G_{a2b}, G_{b2a}} \alpha_1 L_{adv} + \alpha_2 L_{cycle} + \alpha_3 L_{identity}, \quad (7)$$

where α_1 is the weight of the adversarial loss, α_2 is the weight of the cyclic consistency loss, and α_3 is the weight of the identity loss. On the other hand, the discriminators are trained to minimize the following loss:

$$L_D = E_b [(D_b(b) - 1)^2] + E_a [(D_b(G_{b2a}(a)))^2] + E_a [(D_a(a) - 1)^2] + E_b [(D_a(G_{a2b}(b)))^2], \quad (8)$$

where the first and third terms of the loss aim to ensure that the discriminators are classifying real images as real (label 1). Each term corresponds to one of the discriminators. The second and fourth terms ensure that the discriminators correctly classify the synthetic images (label 0).

4 Generation of Synthetic Satellite Multispectral Images

In this paper, we focus on two specific image translation tasks, namely: (i) season transfer, aiming at translating images from summer to winter and vice versa and (ii) land cover transfer, where we focus on the translation of images with vegetation land cover to barren images and vice versa. The two tasks pose different challenges, hence calling for different design choices. We trained a Pix2pix model¹² for the season transfer task, and a cycle GAN model¹¹ for the land cover transfer. The reason for such a choice is that season transfer tends to be a bit more complicated than land cover transfer in terms of diversity within the same domain. In fact, the same input image in a specific domain can correspond to multiple outputs in the second domain. For example, a snowy winter image can translate into an image with less snow in the summer or into green meadows, with the summer domain containing both meadows and light snow images. This diversity in the mapping makes the unpaired image-to-image translation difficult while pairing the images and using pix2pix facilitates the task since it is easy to gather images of the same area in different seasons. On the other hand, we adopted a cycleGAN architecture for the land cover task because it is not possible to create a large enough dataset with paired images with the same location covered by vegetation in one case and barren soil in the other.

In both cases, we are interested in generating synthetic images that resemble the characteristics of real multispectral satellite images in terms of spectral resolution (number of bands), GSD, and radiometric resolution. In fact, satellite images have peculiar characteristics that make them very different from natural photographs.

In this work, we focus on the generation of multispectral images mimicking Sentinel-2 level-1C images. For this reason, we built the training datasets for both tasks by relying on the Sentinel-2 images available from the ESA Copernicus hub.²⁶ Sentinel-2 level1-C is a 13-band product, with four bands (RGB and NIR) sampled at a 10 m sampling distance with 10,980 × 10,980 resolution, six bands sampled at a 20 m sampling distance, and three bands sampled at a 60 m sampling distance. A summary of the characteristics of Sentinel-2 image bands is given in Table 1. All bands have a radiometric resolution of 12 bits per pixel. Image data are distributed with 16-bit word length for fixed-point representation of the spectral radiance.

To cope with the different spatial resolutions of image bands corresponding to different wavelengths, we bicubically interpolated the bands with a GSD lower than 10m to the same size as the 10 m bands (10,980 × 10,980). For this reason, some bands lack details and are a bit blurry in comparison to the 10 m channels, especially those with GSD = 60 m. After upsampling, the images are cropped to a 512 × 512 size using gdal-retile from the gdal software library.²⁷ Eventually, we removed the tiles with no data pixels (0 brightness).

Table 1 GSD of the MSI instruments of Sentinel-2.

Spatial resolution (m)	Band number	Central wavelength (nm)
10	2 (green)	492.4
	3 (blue)	559.8
	4 (red)	664.6
	8 (NIR)	832.8
20	5	704.1
	6	740.5
	7	782.8
	8a	864.7
	11	1613.7
	12	2202.4
60	1	442.7
	9	945.1
	10	1373.5

4.1 Season Transfer

The procedure we followed for the construction of the training dataset, the choice of the network architecture, and the training procedure for the season transfer task are described below.

4.1.1 Dataset

For the season transfer task, we are interested in translating summer images into winter images and vice versa. To do so, we focused on two different geographical regions, one located in China and the other in Scandinavia. It is worth mentioning that the landscape and season transfer conditions differ greatly between the two regions. For the Scandinavian dataset, the landscape is dominated by meadows, and the transfer from summer to winter corresponds to passing from green meadows to snowy land cover, whereas for the images selected for the China dataset, the winter is characterized by barren land cover and the summer by green land cover. For both datasets, we selected image pairs corresponding to the acquisition of the same region in two different months, one in the winter and one in the summer. Also to avoid preprocessing or generating images with clouds, we filtered the images retaining only those with 0% cloud cover. For the Scandinavian dataset, summer images were taken in June 2020, and winter images were acquired in February. We ended up with 9000 images of size 512×512 for each domain. We made sure that all the downloaded products are within the Scandinavian region of Sweden, Denmark, and Norway. We show an example of the RGB channels of two images of the Scandinavian dataset in part Fig. 4(a). For the China dataset, summer images refer to August 2020, and winter images were taken from November 2020 through January 2021. In the end, we collected 8522 images of size 512×512 for each season domain. Also for this dataset, we made sure that all the downloaded products were within the China borders. In Fig. 4(b), we show an example of the RGB channels of the China dataset.

4.1.2 Architecture

For the season transfer task, we chose the pix2pix architecture described in Sec. 3.1. For the generator, we chose a U-Net network²⁸ with skip connections of eight blocks. Each block has two convolutional layers, two batch normalization layers, a leaky ReLU activation function layer with dropout equal to 0.2, and a ReLU activation function layer again with dropout equal to 0.2. As to the discriminator, we used seven convolutional layers, each followed by batch normalization and leaky ReLU activation.

4.1.3 Training

For the China dataset season transfer task, we used 6000 images for training, 2000 for testing, and 522 for validation. For the Scandinavia dataset season transfer task, we used 6000 images for training, 2000 for testing, and 1000 for validation. Each model was trained for 600 epochs for an input of $512 \times 512 \times 13$ pair. For each network, the Adam optimizer²⁹ was used with β_1 set to 0.5, β_2 to 0.999, and a learning rate equal to 0.0001. The number of filters used is 64 and the slope for the leaky ReLU was set to 0.2. The batch size was constrained to 1 due to GPU limitation. The weight for the L_1 loss function λ is set to 100.

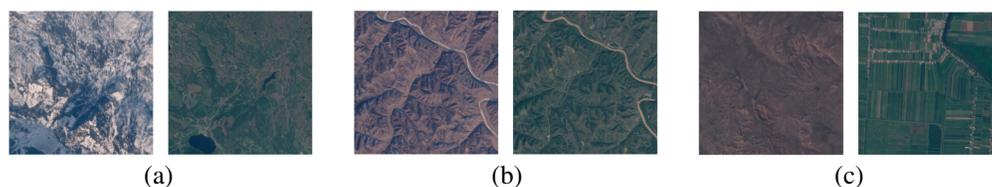


Fig. 4 Examples of Sentinel-2 images of the training datasets. Only a color representation of the RGB bands is shown. (a) Scandinavian and (b) China datasets (RGB channels)—season transfer task: winter (left) and summer (right). (c) Land cover dataset (RGB channels)—land cover transfer task: barren (left) and vegetation (right).

4.2 Land-Cover Transfer

In this section, we focus on the land cover transfer task.

4.2.1 Dataset

For this task, we collected data from two different domains, images with barren cover lands and images with vegetation. For the vegetation domain, we picked an area of interest that is mostly made up of vegetation based on the statistics provided by the organization of economic cooperation and development (OECD).³⁰ In particular, we considered areas of Congo, Salvador, Montenegro, Gabon, and Guyana. The data collected from those regions span from June 2019 to December 2019. Even in this case, we retained only images with 0% cloud cover. Since there is no guarantee that the 512×512 cropped images are representatives of their respective domains, we trained a linear discriminant analysis (LDA) classifier using four images from the training dataset that belong to the vegetation domain and four images belong to the barren domain. Then we manually labeled pixels as belonging to vegetation, barren, water, or artificial surfaces. We used the LDA classifier to make sure that the cropped images are mostly vegetation (more than 70% of the image is vegetation) and only a small fraction corresponds to water, urban, or barren areas. In the end, we gathered 10,000 images.

Similarly, for the barren domain, we relied on OECD³⁰ to pick areas mostly covered by barren soil, with a small percentage of water, vegetation, and urban areas. Specifically, we chose images from South and Central America. As for the vegetation domain, we used the LDA classifier to make sure that the cropped images are mostly barren (more than 70% barren) with small percentages of water, urban, or vegetation. In Fig. 4, we show an RGB example of the images we got for the two different domains.

4.2.2 Architecture

As we already explained, for the landcover transfer, we used a CycleGAN architecture. For the generator, we chose a residual ResNet network³¹ consisting of six residual blocks with skip connection. Each residual block has a convolutional layer, a batch normalization layer, and a leaky ReLU activation function layer. For the discriminator, we used seven convolutional layers, each followed by batch normalization and leaky ReLU activation. For both networks, the Adam optimizer²⁹ was used.

While training the model, we observed that even after 600 epochs, the quality of the transferred images was not improving, and the resulting images were blurry. Our conjecture is that this is due to the different ground resolutions of the 13 bands. To overcome this problem, we split the identity loss $L_{identity}$ and the cyclic loss L_{cycle} in three parts, each referring to a group of bands with the same spatial resolution, by modifying the loss terms as follows:

$$L_{identity_mod} = \beta_1 L_{identity}[2,3,4,8] + \beta_2 L_{identity}[5,6,7,8a,11,12] + \beta_3 L_{identity}[1,9,10], \quad (9)$$

$$L_{cycle_mod} = \beta_1 L_{cycle}[2,3,4,8] + \beta_2 L_{cycle}[5,6,7,8a,11,12] + \beta_3 L_{cycle}[1,9,10], \quad (10)$$

where β_1 is the weight for the 10 m spatial resolution bands, β_2 is the weight for the 20 m, and β_3 is the weight for 60 m.

Then we trained the network by substituting the L_{cycle} loss with the L_{cycle_mod} loss and the $L_{identity}$ loss with the $L_{identity_mod}$ losses for additional 150 epochs. In this way, we were able to reduce the blurriness of the generated images. In the following, we refer to this architecture as “weighted_cycleGAN.” Yet, the domain transfer was not very evident. Assuming that the reason is the different spatial resolutions of the bands, we split the discriminator into three discriminators focusing on the bands at a specific spatial resolution, each one with its specific loss:

$$\begin{aligned} L_{D_1} &= L_D[2,3,4,8] \\ L_{D_2} &= L_D[5,6,7,8a,11,12] \\ L_{D_3} &= L_D[1,9,10]. \end{aligned} \quad (11)$$

This model was trained for 60 epochs. The initial weights used for the generator were the weights obtained after training the previous 750 epochs; however, the discriminators were initialized randomly. After 60 epochs, we stopped training since the losses plateaued without further improvement and according to our visual assessment, the translated images exhibited satisfactory quality but showed no further enhancement with additional training. In the following, we call this model “3dis_cycleGAN.”

4.2.3 Training

We used 8000 images for training the model and 2000 were kept for testing. For each network in the model, we used the Adam optimizer²⁹ with $\beta_1 = 0.5$, $\beta_2 = 0.999$, and the learning rate equal to 0.0001. The number of filters used is 32, and the slope of the leaky ReLU was set to 0.2. The batch size was constrained to 1 due to GPU limitation. The weight α_1 of the GAN adversarial loss was set to 1 and the cyclic consistency weight α_2 to 5. The identity loss weight α_3 was set to 3, β_1 to 13/16, β_2 to 1/8, and β_3 to 1/16.

5 Improving Image Quality via Pansharpener

Most generators’ architectures adopt an encoder–decoder structure,³ due to which the generated images look slightly blurred and with a certain lack of fine spatial details. In Fig. 5, we show an example of the above effect when a vegetation image [Fig. 5(a)] is remapped into barren soil [Fig. 5(b)]. The sharpness of the details of the buildings contained in the synthetic images is visibly lower than that in the original image.

In this section, we propose an algorithm to improve the sharpness of the synthetic images generated by the image-transfer architectures. The algorithm is inspired by pansharpener algorithms usually applied to improve the sharpness of multispectral images.⁴ Pansharpener uses the availability of a panchromatic image (PAN) to sharpen the corresponding multispectral images since they usually contain fewer details than their panchromatic counterpart. Component

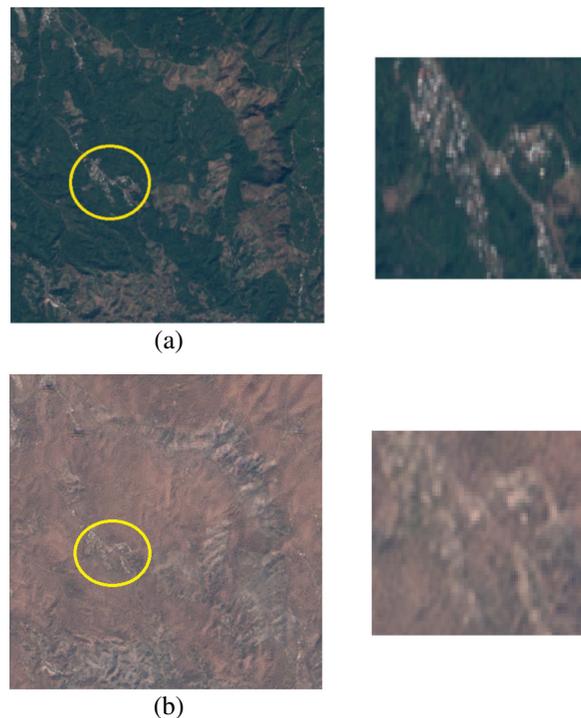


Fig. 5 Example of the lack of details typical of synthetically generated images: (a) real vegetation and (b) generated barren images.

substitution is a popular class of pansharpener algorithms. It works by transforming the low-resolution multispectral images into a different domain, where the spatial and spectral structures are separated. The spatial structures are then replaced by the corresponding components of the PAN image. After the replacement, the multispectral image is brought back into the original domain. Pansharpener relies on the details contained in a high-resolution PAN image, which is not available in image-transfer applications. Since in our case, the spatial structures, such as buildings or roads, contained in the source images must also be present, with the same resolution, in the synthetically generated images, we propose to use the source image to improve the sharpness of the images generated by the GANs. In other words, we first build an artificial PAN image by starting from the source multispectral image used to drive the image-transfer architecture. Then the GSA pansharpener algorithm³² is applied to improve the quality of the synthetic images produced by the network.

5.1 Sharpening of Synthetic Images by GSA

In the following, we indicate with x the source multispectral image and with y the synthetic image generated by the GAN. By drawing an analogy with pansharpener algorithms, y represents the low-resolution image, and the high-resolution image I_{pan} is estimated from the source multispectral image x . The spectral bands of both images are indicated, respectively, by x_i and y_i , $i \in \{1, 2, \dots, n\}$, and where n is the number of bands ($n = 13$ for Sentinel images). The pixel position is indicated by (j, h) , with j and $h \in \{1, 2, \dots, m\}$, where m is the width and height of each band ($m = 512$ in our case). We divide both x and y into three subsets where each subset represents a set of bands belonging to the same GSD resulting in a subset for 10, 20, and 60 m GSD. The pansharpener algorithm described in the following that is applied separately to each subset then the image bands are put together again to form the 13 bands image. In this way, m remains fixed to 512, however, n varies based on which subset we are processing (4 for 10 m, 6 for 20 m, and 3 for 60 m). To avoid heavy symbolism, in the following, we use x to represent the multispectral image for the subset of bands being processed and y to indicate the synthetic image generated by the GAN for the same subset of bands.

As a first step, we estimate the high-resolution image I_{pan} whose spatial details will be used to improve the sharpness of y . In particular, the estimated I_{pan} image is obtained as a linear combination of the source image bands x_i 's. In order to determine the coefficients of the linear combination, hereafter indicated by $\alpha = (\alpha_1, \dots, \alpha_n)$, we first compute a panchromatic version y_{av} of the synthetic image by spectral averaging all the bands of y . The coefficient vector α is then computed by applying a linear regression between the AC components (The AC component of an image is obtained by removing from the image its spatial average.) of the image bands of the source image and y_{av} . Eventually, the AC component of I_{pan} will be used to enrich the details of the synthetic image by means of GSA pansharpener.

More specifically, the exact procedure we used to build the I_{pan} image is described by the following steps.

- Compute the spectral average of the synthetic multispectral image by averaging all the bands of y :

$$y_{\text{av}} = \sum_i^n y_i/n. \quad (12)$$

- Remove the spatial mean from y_{av} :

$$\hat{y}_{\text{av}} = y_{\text{av}} - \sum_{j=1}^m \sum_{h=1}^m y_{\text{av}}(j, h)/m^2. \quad (13)$$

- Remove the spatial mean from each band of the high-resolution source image:

$$\hat{x}_i = x_i - \sum_{j=1}^m \sum_{h=1}^m x_i(j, h)/m^2. \quad (14)$$

- Compute a set of weights α_i by applying a linear regression between \hat{y}_{av} and the bands of \hat{x} . The linear regression aims at finding the coefficients α_i 's that permit to better approximate \hat{y}_{av} starting from \hat{x}_i 's:

$$\alpha = \arg \min_{\alpha} \left\| \hat{y}_{av} - \sum_i \alpha_i \hat{x}_i \right\|^2. \quad (15)$$

- Use the weights obtained in the previous step to build the I_{pan} image:

$$I_{pan} = \sum_{i=1}^n \alpha_i x_i. \quad (16)$$

- Extract the high-resolution content of the I_{pan} image, by removing from it the spatial mean:

$$\hat{I}_{pan} = I_{pan} - \sum_{j=1}^m \sum_{h=1}^m I_{pan}(j, h) / m^2. \quad (17)$$

After computing \hat{I}_{pan} , we proceed by applying the classical GSA algorithm³² depicted in Fig. 6 and detailed below.

To start with, the spatial mean of each band of y is subtracted from the corresponding band, as shown in section I in Fig. 6, where \bar{y}_i in this figure corresponds to $\sum_{j=1}^m \sum_{h=1}^m y_i(j, h) / m^2$:

$$\hat{y}_i = y_i - \sum_{j=1}^m \sum_{h=1}^m y_i(j, h) / m^2. \quad (18)$$

Then

- The spectral weights w_i are computed by applying a linear regression between \hat{y} and $I_{LR\ pan}$, which is obtained by applying wavelet transform to compute a low-pass version of the I_{pan} image.
- The low-resolution approximations of the I_{pan} image is computed by starting from the zero mean low-resolution image bands (section II in Fig. 6):

$$\hat{y}_0 = \sum_{i=1}^n w_i \hat{y}_i. \quad (19)$$

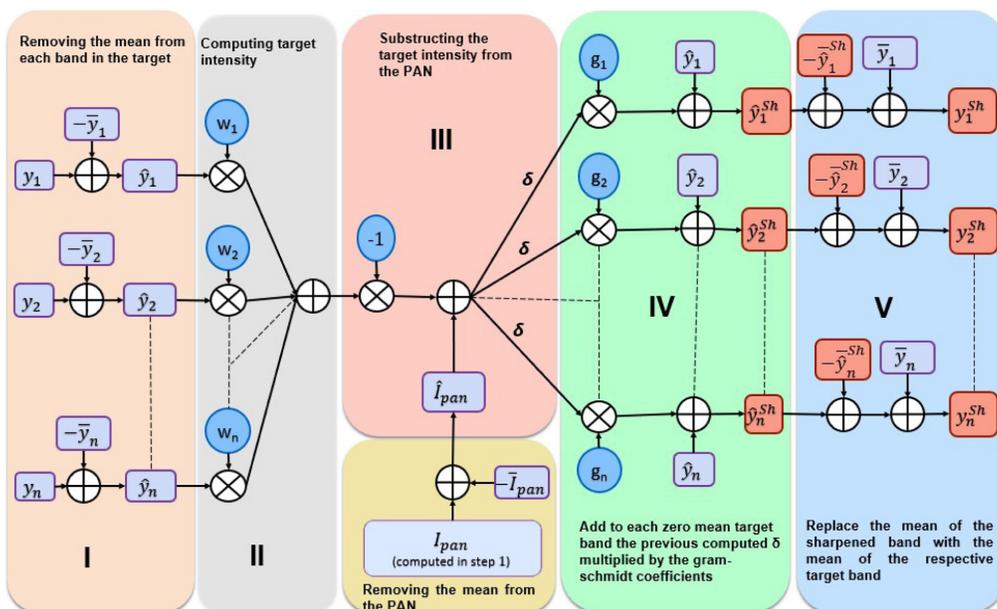


Fig. 6 GSA workflow.

- \hat{y}_0 is subtracted from the zero-mean \hat{I}_{pan} to obtain the details δ that are lacking from the low-resolution y image (section III in Fig. 6):

$$\delta = \hat{I}_{\text{pan}} - \hat{y}_0. \quad (20)$$

- Following Ref. 32, we compute the gain injection coefficients g_i as

$$g_i = \text{cov}(\hat{y}_0, \hat{y}_i) / \text{var}(\hat{y}_0), \quad (21)$$

where cov is the covariance matrix between the two matrices, and var is the variance.

- Add each zero mean low-resolution band to the previously computed details multiplied by the respective gain injection coefficients (section IV in Fig. 6):

$$\hat{y}_i^{\text{sh}} = \hat{y}_i + g_i \delta. \quad (22)$$

- Replace the mean of the sharpened bands with the mean of the respective low-resolution band y_i (section V in Fig. 6):

$$\hat{y}_i^{\text{sh}} = \hat{y}_i - \sum_{j=1}^m \sum_{h=1}^m \hat{y}_i(j, h)^{\text{sh}} / m^2 + \sum_{j=1}^m \sum_{h=1}^m y_i(j, h)^{\text{sh}} / m^2. \quad (23)$$

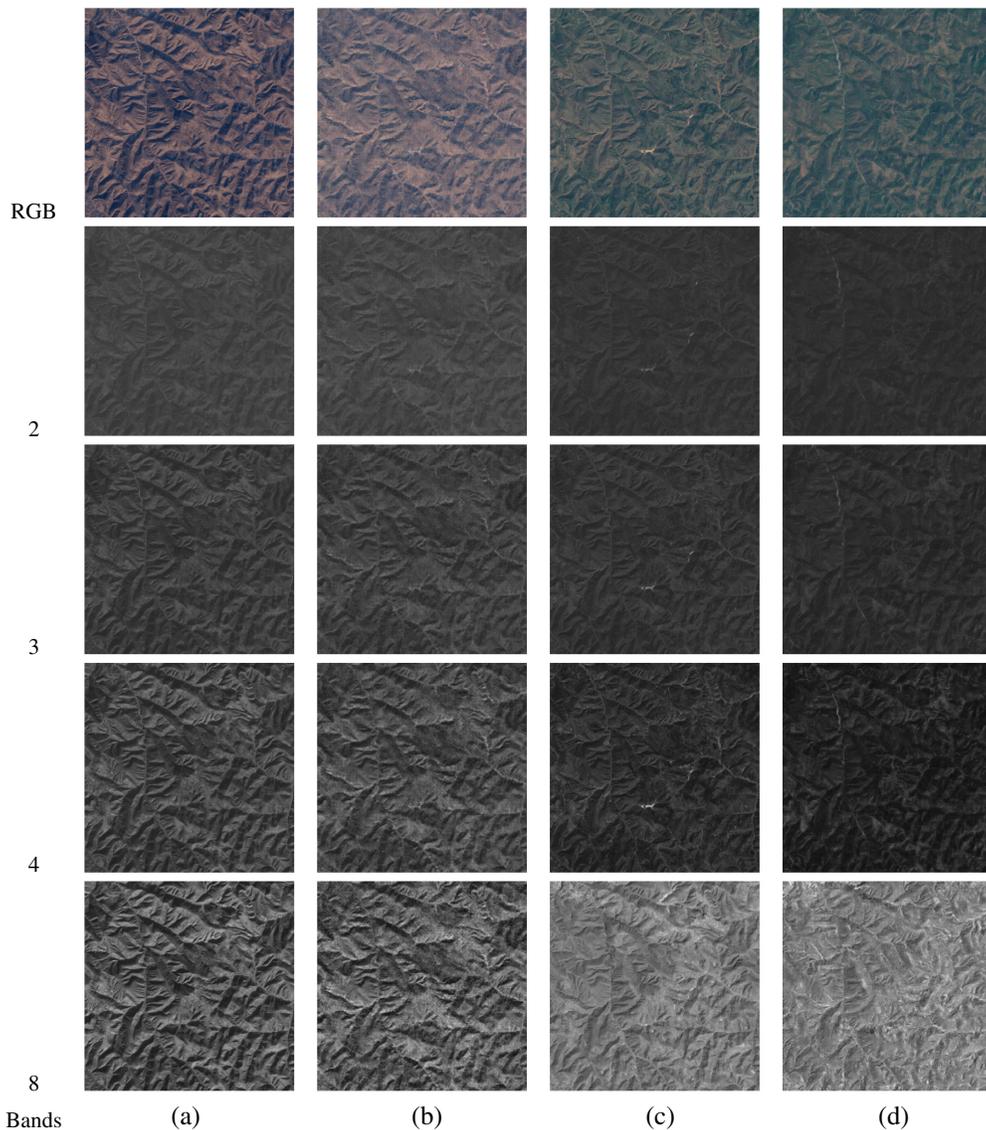


Fig. 7 Example of season transfer for the China dataset: 10 m bands. (a) Real winter, (b) generated winter, (c) real summer, and (d) generated summer.

6 Results

In this section, we present the results we got by applying the procedures described in the previous sections to generate synthetic Sentinel by means of image transfer. We first present the results we got by applying the basic image transfer models, then we show the improvement obtained by means of pansharpening.

6.1 Season Transfer

In Figs. 7–9, we show an example of season transfer for the China dataset for the 10, 20, and 60 m bands, respectively. Similarly for the Scandinavian dataset, Figs. 10–12 show the translated 10,

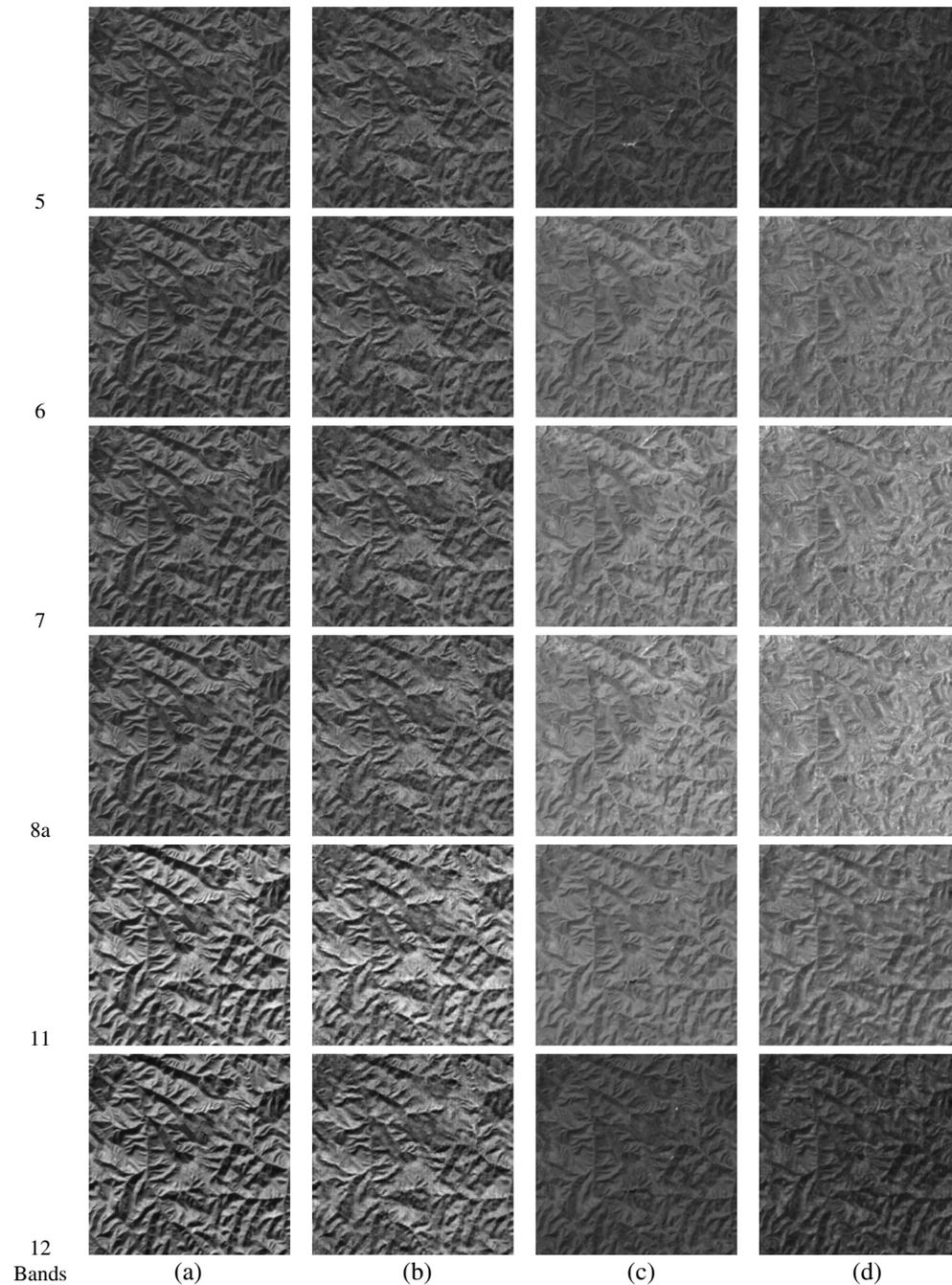


Fig. 8 Example of season transfer for the China dataset: 20 m bands. (a) Real winter, (b) generated winter, (c) real summer, and (d) generated summer.

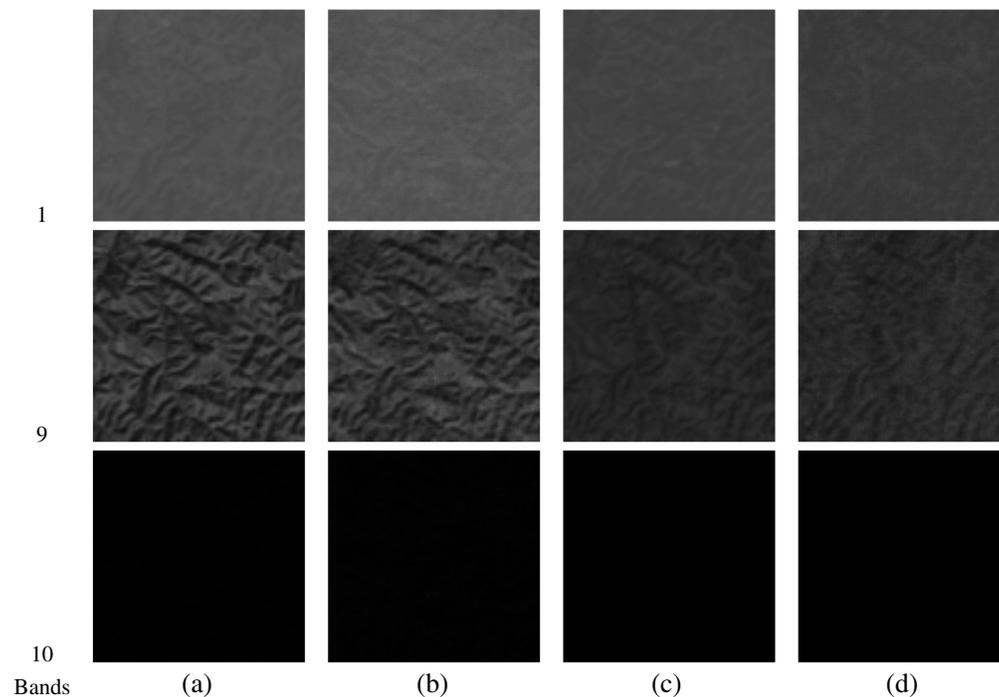


Fig. 9 Example of season transfer for the China dataset: 60 m Bands. (a) Real winter, (b) generated winter, (c) real summer, and (d) generated summer.

20, and 60 m bands, respectively. For each dataset, the transfer is applied in both directions: from summer to winter and from winter to summer. The generated winter (b) image is generated starting from the real summer image (c). By visual inspection, we can see that for all the bands, the synthetic images are very close to real winter images (a). The overall brightness and spatial properties are conserved, with the 60 m bands having a lower spatial resolution and thus mimicking the content of real images. The 20 m resolution bands are still a bit blurry, even if less than the 60 m ones, and the 10 m resolution bands have a better ground resolution. Similar observations hold for the synthetic summer image (d), created starting from the real winter image (a). We also note that for the China dataset, the land cover in winter images is mostly dry, and in summer images it is prominently greenish. For the Scandinavian dataset, summer images are greenish, and the synthetic winter images are dark and snowy (as they should be).

6.2 Land-Cover Transfer

In Fig. 13, we show a comparison of the images generated with the three cycleGAN models we have developed (see Sec. 4.2.2), namely: cycleGAN, weighted_cycleGAN, and 3dis_cycleGAN. To ease the comparison, we applied the vegetation-to-barren transformation to the same image using the three models. In Fig. 13(b), we apply weights for the cycle loss and the identity loss based on the spatial resolution, assuming that the higher the spatial resolution, the more importance we need to give to these losses. In comparison to Fig. 13(a) where the basic loss is applied, the generated image is sharper and contains more details but the transfer is not strong enough. In the third variant of the model (3dis_cycleGAN), we split the discriminator into three discriminators, one for each spatial resolution. The image produced by this model is shown in Fig. 13(c). The stronger transfer effect achieved by this network can be appreciated easily, given the more diffused presence of barren terrain than in the images produced by the other two models. A complete, 13-band, example of barren to vegetation and vegetation to barren transformation using the 3dis_cycleGAN model is shown in Fig. 14–16. In Fig. 13, we show a comparison of the images generated with the three cycleGAN models we have developed (see Sec. 4.2.2), namely: cycleGAN, weighted_cycleGAN, and 3dis_cycleGAN. To ease the comparison,

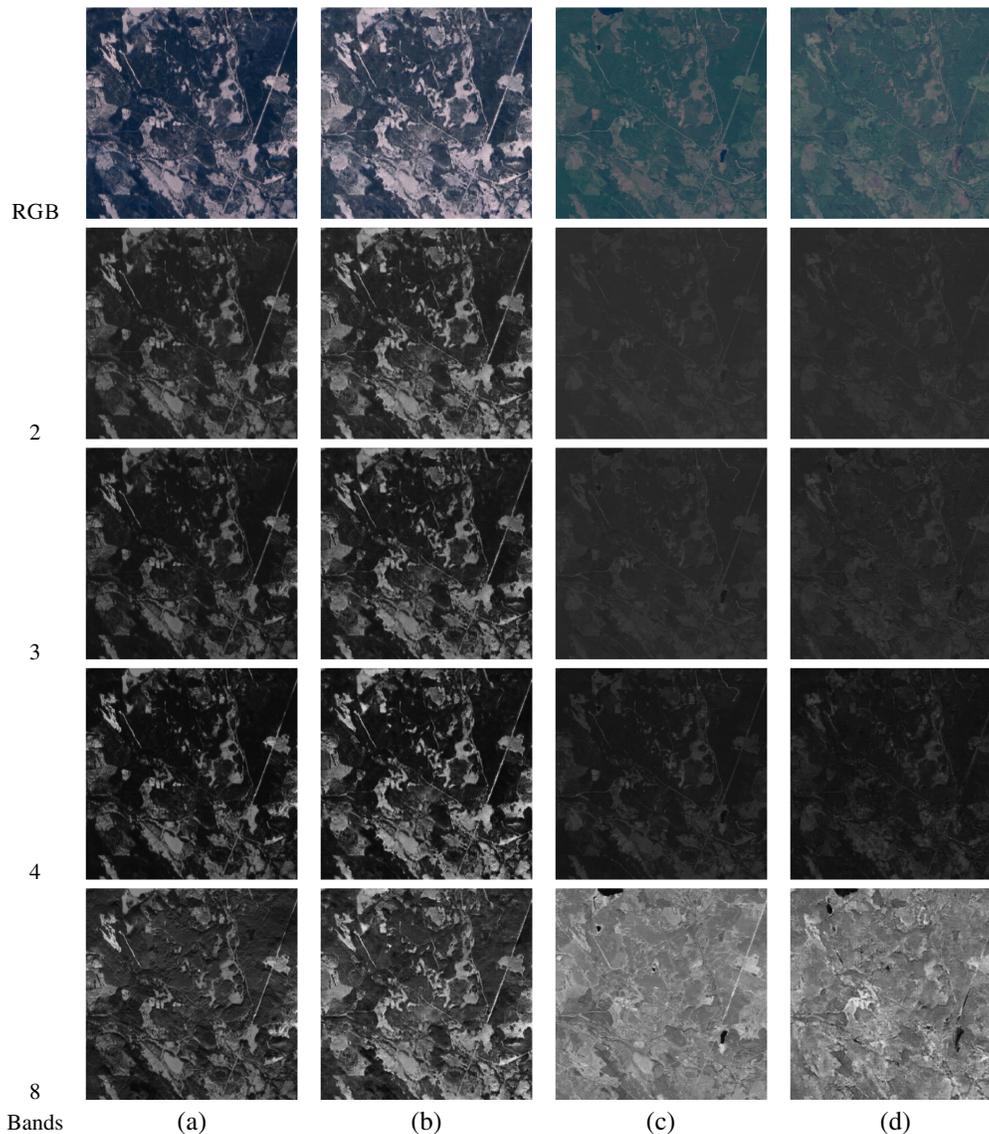


Fig. 10 Example of season transfer for the Scandinavian dataset: 10 m bands. (a) Real winter, (b) generated winter, (c) real summer, and (d) generated summer.

we applied the vegetation-to-barren transformation to the same image using the three models. In Fig. 13(b), we apply weights for the cycle loss and the identity loss based on the spatial resolution, assuming that the higher the spatial resolution, the more importance we need to give to these losses. In comparison to Fig. 13(a), where the basic loss is applied, the generated image is sharper and contains more details but the transfer is not strong enough. In the third variant of the model (3dis_cycleGAN), we split the discriminator into three discriminators, one for each spatial resolution. The image produced by this model is shown in Fig. 13(c). The stronger transfer effect achieved by this network can be appreciated easily, given the more diffused presence of barren terrain in the images produced by the other two models. A complete, 13-band, example of barren to vegetation and vegetation to barren transformation using the 3dis_cycleGAN model is shown in Figs. 14–16 for the 10, 20, and 60 m bands, respectively.

To better quantify the quality of the transferred images, we classified the image pixels into four classes (high vegetation, low vegetation, barren, and water) using the classifier based on the normalized difference vegetation index (NDVI) described in Ref. 3. So we computed the NDVI for the pixels belonging to the various classes. We assumed that an NDVI value below

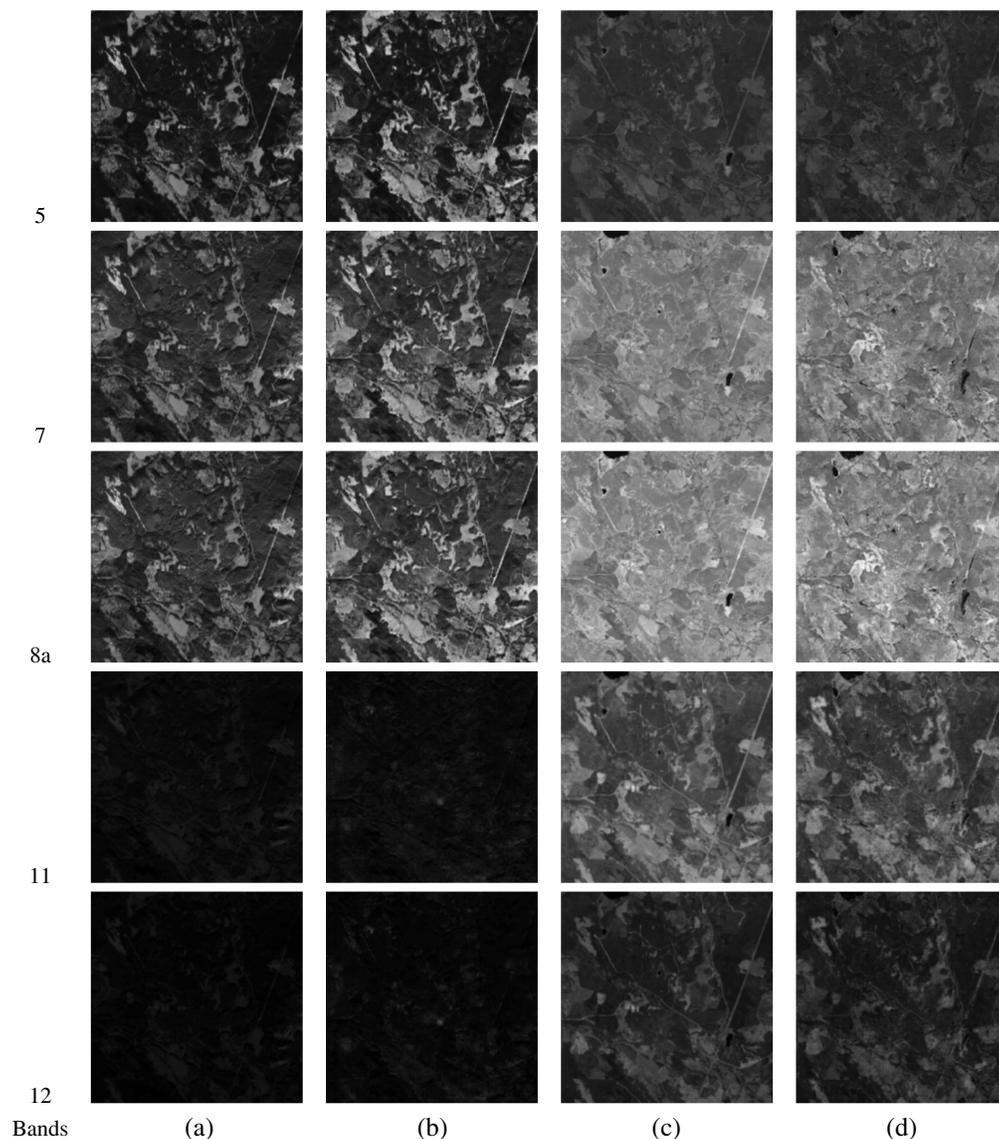


Fig. 11 Example of season transfer for the Scandinavian dataset: 20 m bands. (a) Real winter, (b) generated winter, (c) real summer, and (d) generated summer.

-0.1 corresponds to water pixels, barren in the $[-0.1, 0.1]$ range, low vegetation in $[0.1, 0.4]$, and high vegetation when $NDVI > 0.4$. In Table 2, we report the result of the pixel classification for the different datasets. We computed the percentage of the correctly classified pixels in both types of images (real and synthetic). We ran the classifier on the 2000 real vegetation images in the test dataset, and we got confirmation that the majority of the pixels have a high vegetation index. Then we repeated the same procedure for the 2000 real barren images, and we found that, as expected, the majority of the pixels belong to barren terrain. Then we classified the pixels of the images obtained by the three cycleGAN variants. In all cases and for both transformations, the overall content of the synthetic images corresponds to the content of the real images of the same class. The notable exception is when we move from vegetation to barren. We can see that the standard cycleGAN and weighted_cycleGAN do not transfer totally into barren but instead reduce the vegetation into low vegetation, whereas 3dis_cycleGAN converts the vegetation pixels into barren ones, yielding a stronger transfer, and that is also quite evident in Fig. 13, where the image generated by the 3dis_cyclegan contains more barren soil than the other two images.

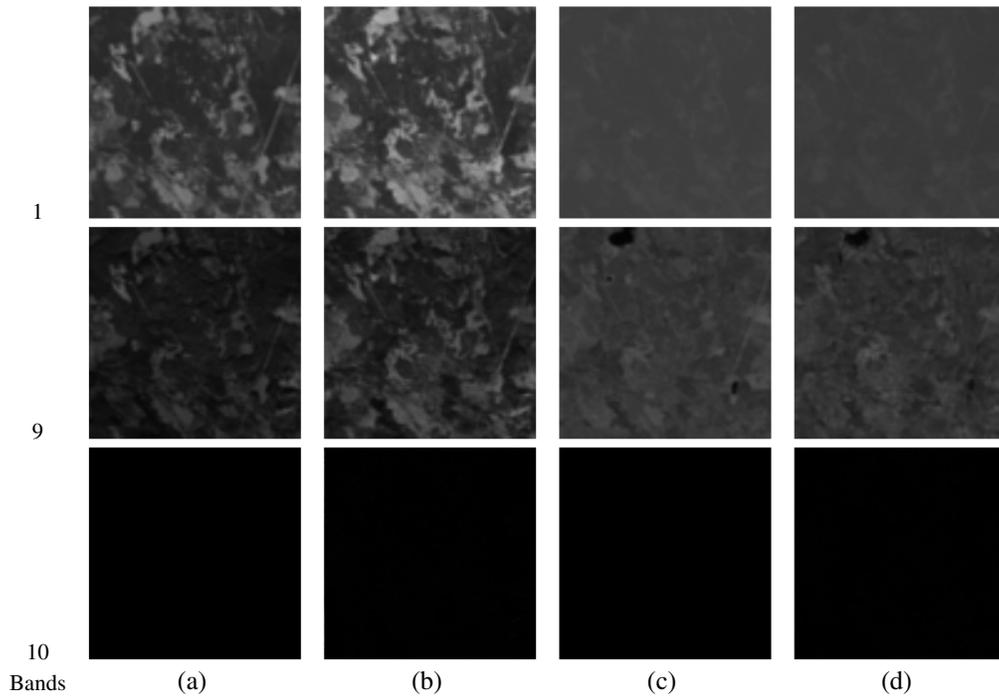


Fig. 12 Example of season transfer for the Scandinavian dataset: 60 m bands. (a) Real winter, (b) generated winter, (c) real summer, and (d) generated summer.

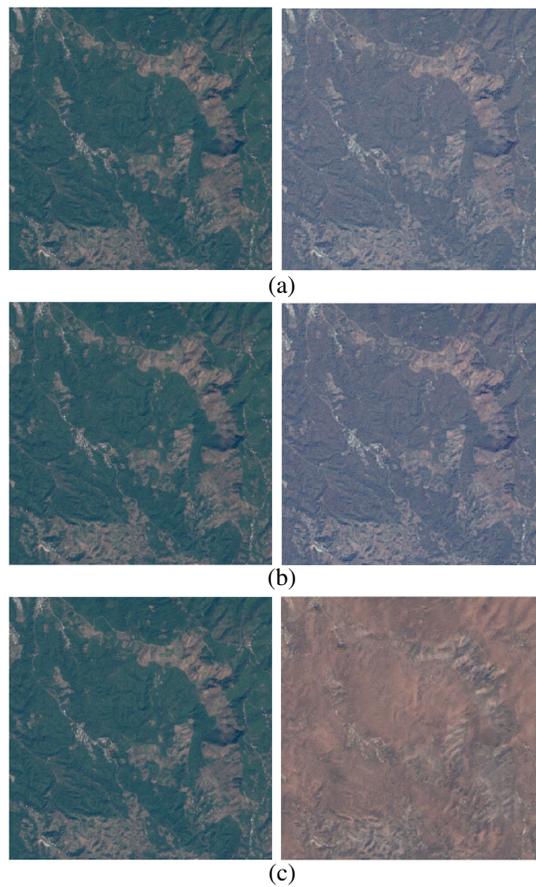


Fig. 13 Comparison of vegetation-to-barren images obtained with the different models described in Sec. 4.2.2. To ease the visual analysis, only the RGB bands are shown. Land cover translation with (a) standard cycleGAN, (b) weighted cycleGAN, and (c) 3dis cycleGAN.

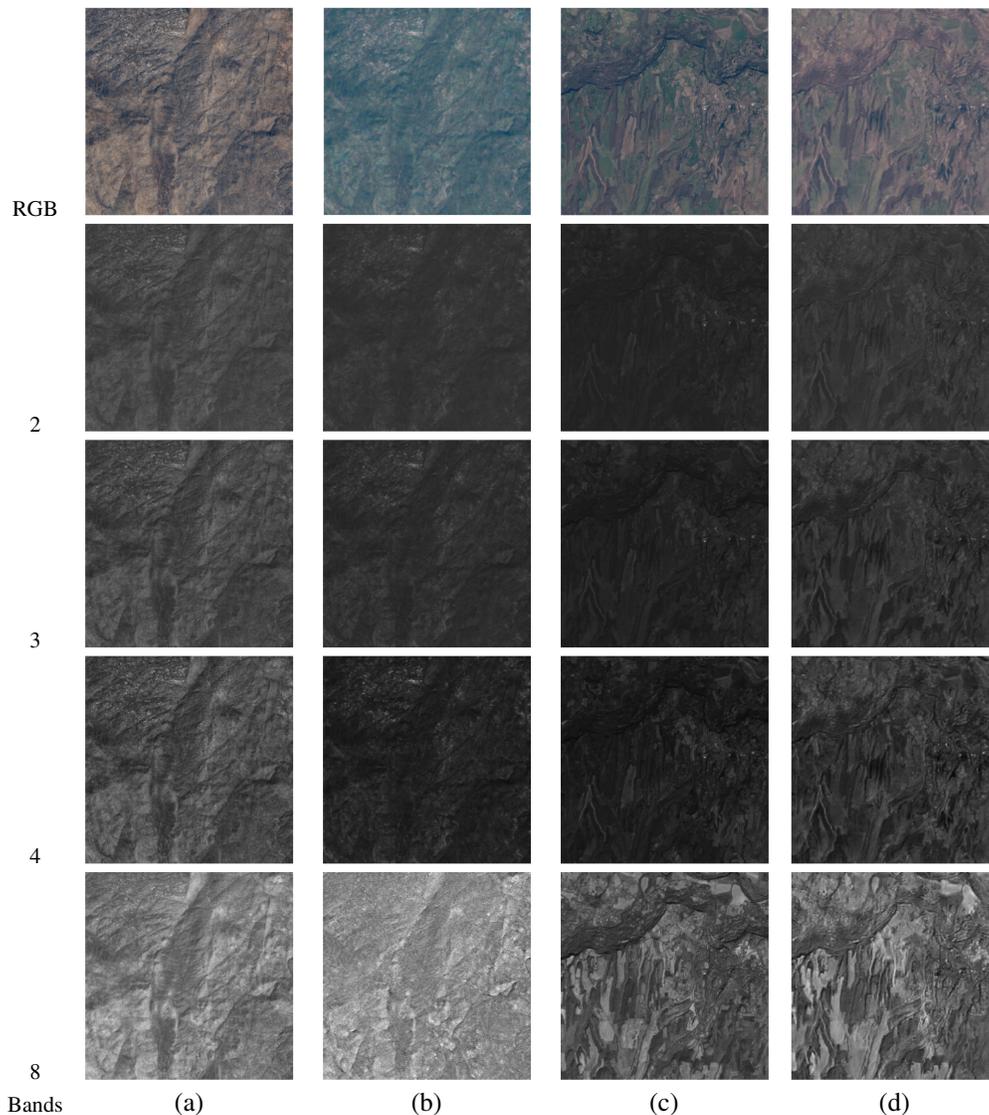


Fig. 14 Land cover transfer example using 3dis_cycleGAN: 10 m bands. (a) Real winter, (b) generated winter, (c) real summer, and (d) generated summer.

6.3 Sharpness Improvement by Pansharpening

In Secs. 6.1 and 6.2, we discussed the quality of the images obtained by applying season transfer and land cover transfer. Despite the good similarity to real images, the synthetic images are visibly less sharp than the pristine images used to generate them. In Fig. 17, we show some examples of source images from the vegetation domain (resp. barren domain) and their respective barren (resp. vegetation) GAN-generated image obtained by applying the 3dis_cyclegan model and their sharpened counterpart after applying GSA. It is evident that the sharpened images have much more spatial details than the GAN-generated ones. Similarly, Fig. 18 shows a couple of RGB examples obtained by applying the season transfer task to the China dataset. For both winter-to-summer and summer-to-winter transfers, we show the input image, the respective GAN image, and the sharpened image. Figures 19–21 shows the 10, 20, and 60 m bands, respectively, of an image of the China dataset after winter to summer transfer with and without sharpening.

Giving a quantitative measure of the quality of the sharpening algorithm is not easy since we do not actually have a reference image to compare our results with. For this reason, we employed a general no-reference image quality metric capable of quantifying the quality of the generated

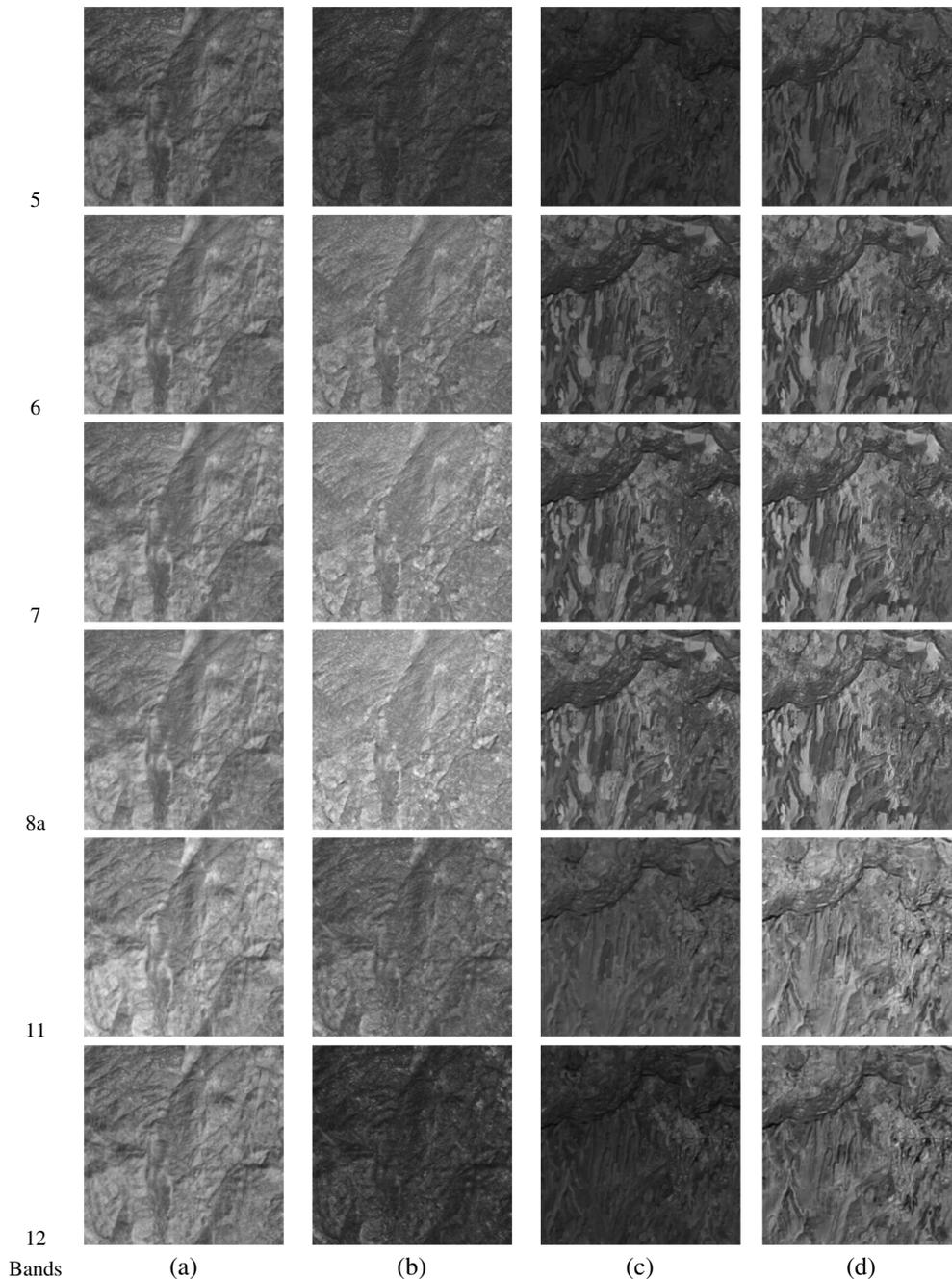


Fig. 15 Land cover transfer example using 3dis_cycleGAN: 20 m bands. (a) Real winter, (b) generated winter, (c) real summer, and (d) generated summer.

images without a reference and without resorting to opinion-based supervised learning. In particular, we used the PIQUE metric,³³ whose aim is to estimate the amount of distortion contained in an image based on local, block-level, features. The lower the score is, the better the image quality. We refer the reader to Ref. 33 for a detailed description of the metric. In the following, we describe the results we have obtained by applying the PIQUE metric to the examples reported in Fig. 17.

For the image, in Fig. 17(a) where we translated an input vegetation image into a barren image using the 3dis_cycleGAN model, PIQUE resulted in a score of 16.1 for the source vegetation image and a score of 9.64 for the sharpened translated image considering only the RGB

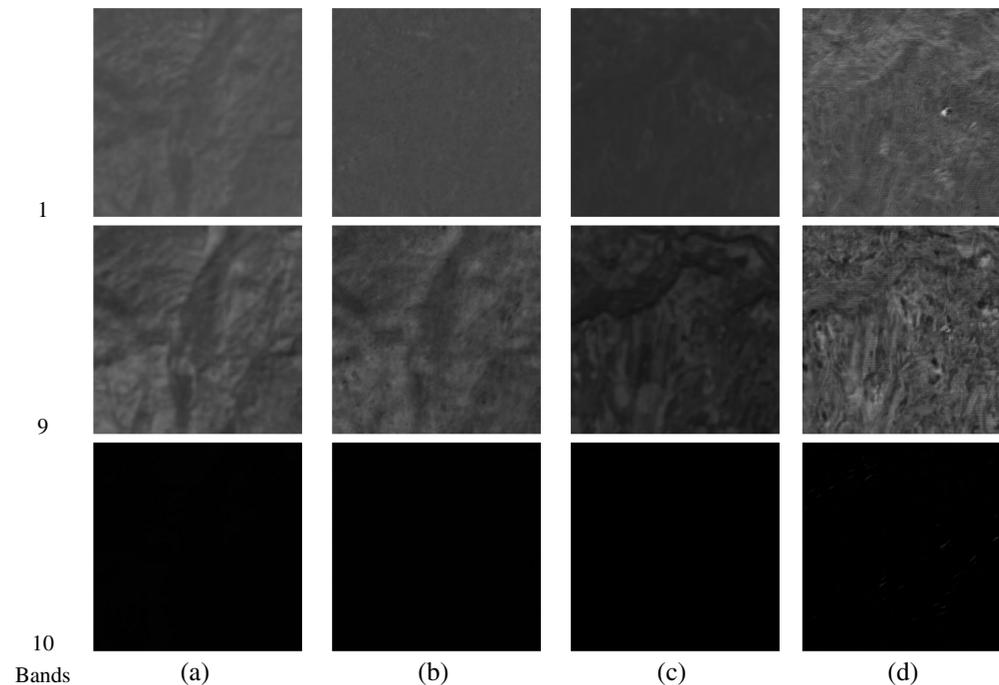


Fig. 16 Land cover transfer example using 3dis_cycleGAN: 60 m bands. (a) Real winter, (b) generated winter, (c) real summer, and (d) generated summer.

Table 2 Percentage of the pixels classified correctly in real and synthetic images based on NDVI.

Class	Dataset			
	High vegetation (%)	Low vegetation (%)	Barren (%)	Water (%)
Real vegetation	77.87	21.81	0.29	0.01
GAN vegetation (base_cycleGAN)	99	1	0	0
GAN vegetation (weighted_cycleGAN)	94.2	5.79	0	0
GAN vegetation (3dis_cycleGAN)	99.99	0.001	0	0
Real barren	0	0	100	0
GAN barren (base_cycleGAN)	3.64	94.92	1.4	0.01
GAN barren (weighted_cycleGAN)	3.7	94.41	1.86	0.01
GAN barren (3dis_cycleGAN)	0	6.87	93.12	0.01

bands. Hence, in that case, the image quality actually improved after translation and sharpening. On the other hand, for the image in Fig. 17(b), the score is 7.3 for the input barren image and 7.29 for the vegetation sharpened counterpart (again considering only the RGB bands). In that case, the quality of the image remained with no significant deterioration. We conclude that adding spatial details using the method described in Sec. 5 improved the image sharpness, producing images whose quality is comparable to that of the original source image. With reference to the image shown in Fig. 19, we also computed the PIQUE score band by band for the original image, the translated image, and the sharpened image. The results are shown in Table 3, where the sharpened score for each band is generally smaller (or much smaller) than the synthetic image without sharpening.

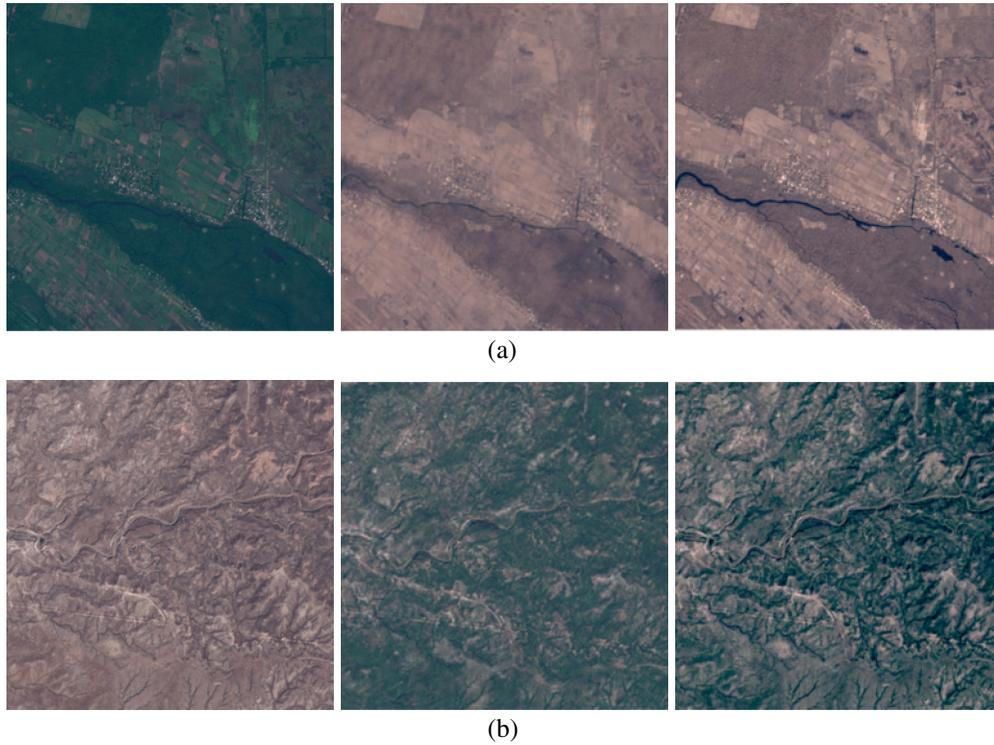


Fig. 17 Sharpened images after postprocessing sharpening from the land cover dataset. Source image, the generated output by (a) vegetation to barren and (b) barren to vegetation 3dis cycleGAN model and its sharpened counterpart.

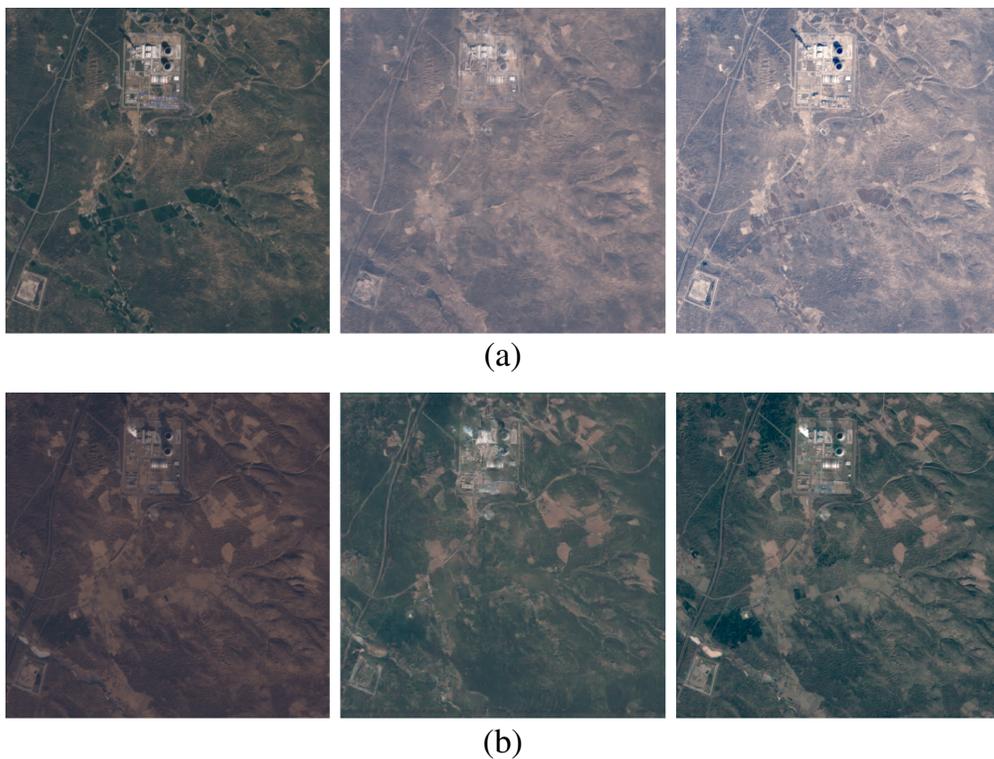


Fig. 18 Effect of sharpening on season transfer images of the China dataset: (a) summer to winter and (b) winter to summer. In both cases, the original image is shown in the left, the transferred image in the center, and the transferred image after pansharpening in the right.

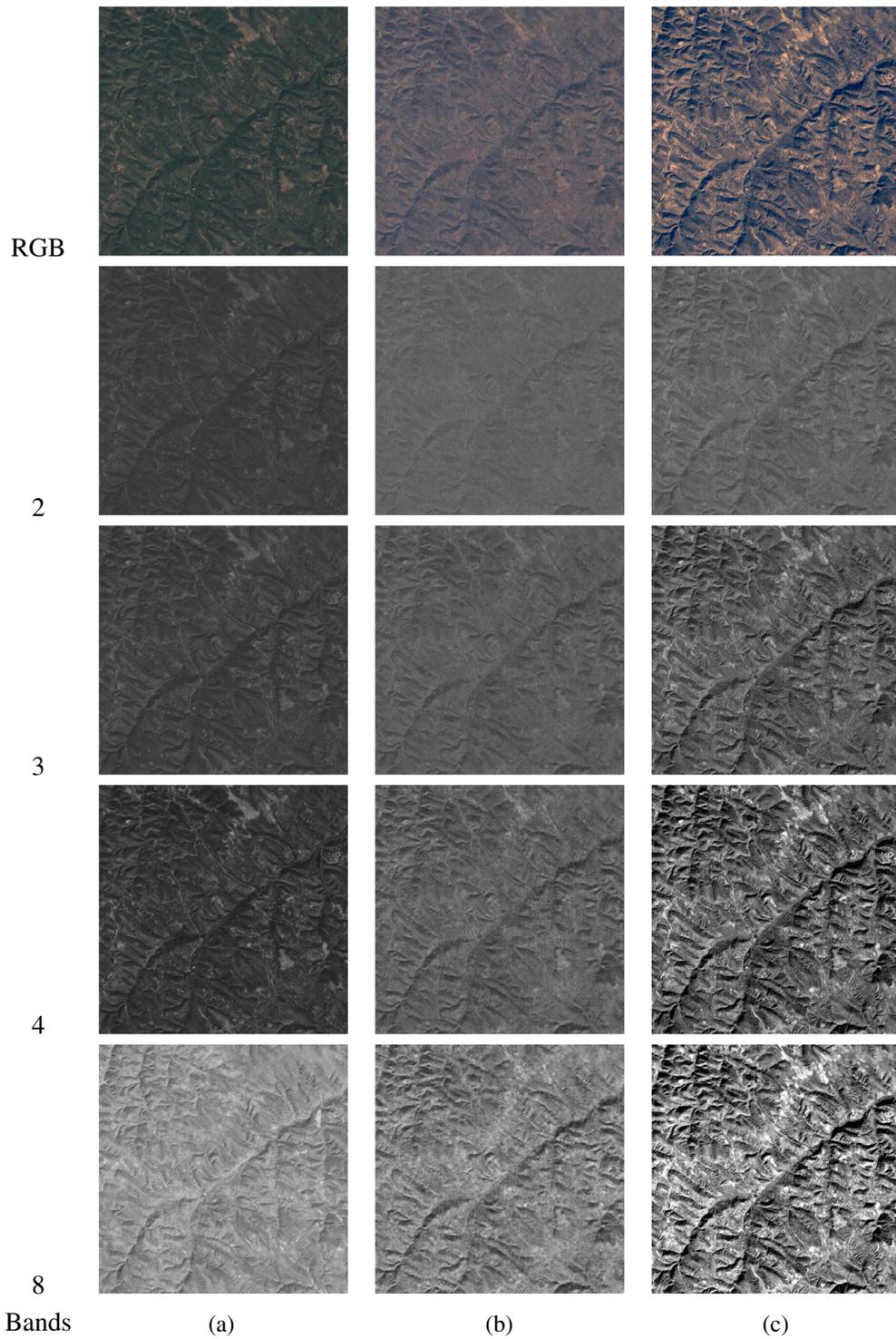


Fig. 19 Sharpening season transfer example on China dataset: 10 m bands. (a) Real, (b) generated, and (c) sharpened China datasets.

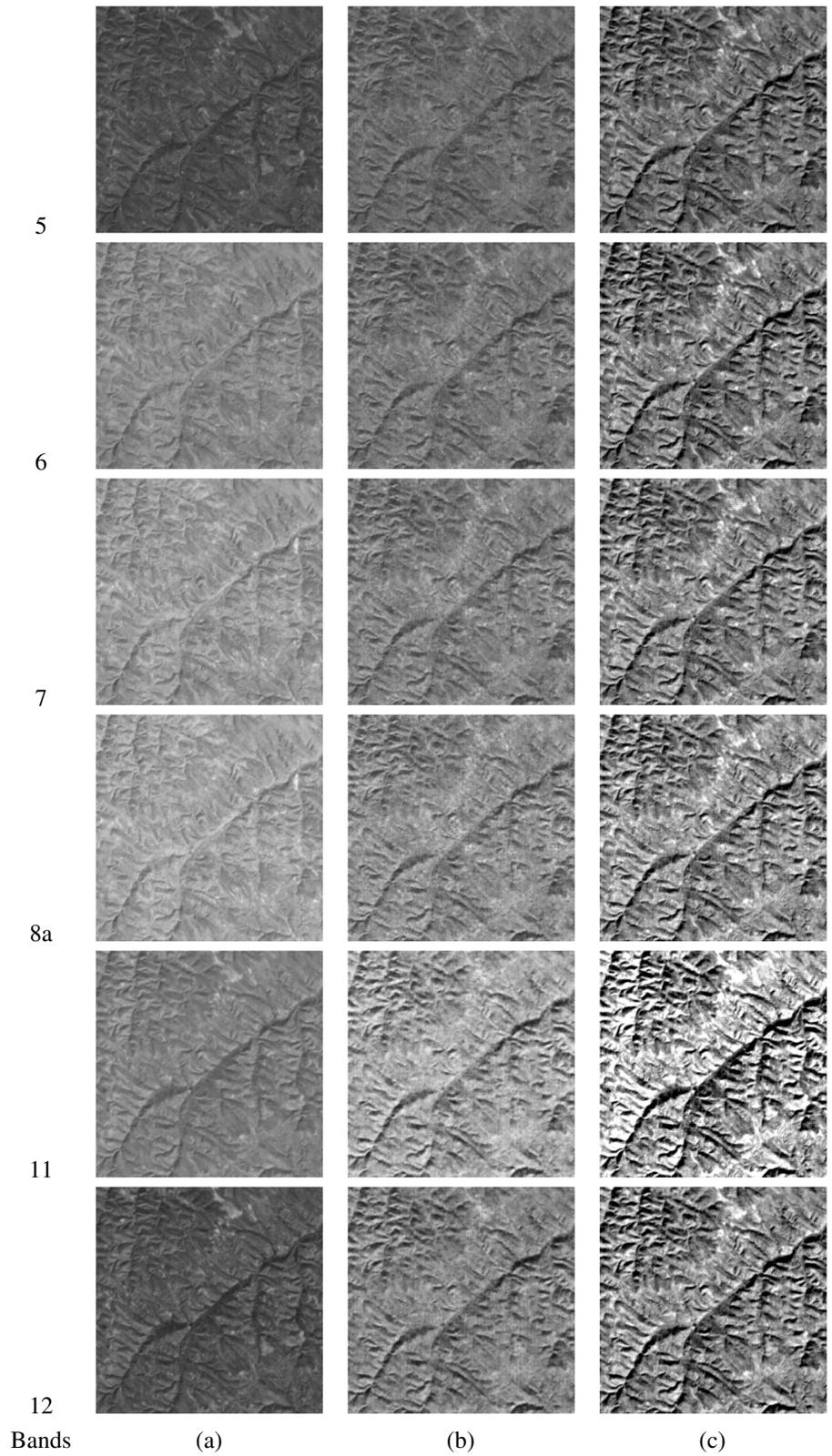


Fig. 20 Sharpening season transfer example on China dataset: 20 m bands. (a) Real, (b) generated, and (c) sharpened China datasets.

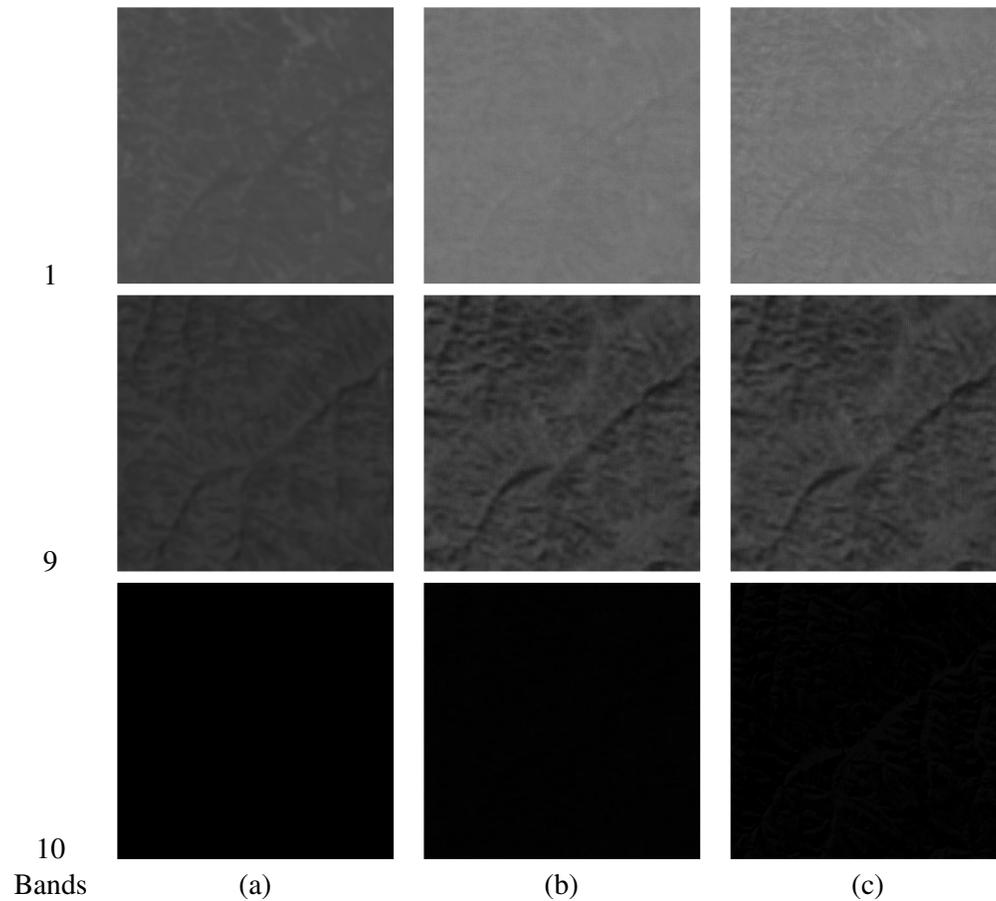


Fig. 21 Sharpening season transfer example on China dataset: 60 m bands. (a) Real, (b) generated, and (c) sharpened China datasets.

Table 3 No reference visual quality per band³³ (PIQUE) for the example shown in Fig. 19.

Band	Original input image	GAN image	Sharpened image
1	90.2	60.6	52.6
2	9.5	24.7	20.2
3	12.3	12.6	11.7
4	14.4	16.8	8.3
5	5	59.7	8.8
6	6.4	59.9	9.6
7	5.7	61.9	11.3
8	5.7	9.5	22
8a	7.9	61.1	11.3
9	39.4	87.4	39.6
10	44.2	79.9	44.6
11	11	71.9	32.5
12	8.5	67.3	17.8

7 Conclusion

In this paper, we have proposed two GANs specifically thought to generate synthetic multispectral satellite images consisting of 13 bands with sharpened spatial details. The proposed architectures have been applied to two image transfer tasks, namely, (i) land cover transfer, whereby the land cover of the source image is changed from vegetation to barren and vice versa and (ii) season transfer, according to which the image season is changed from summer to winter and vice versa. To cope with the blurriness of the images produced by the generative networks, we have introduced a pansharpening-like postprocessing step, whereby the spatial structures of the input image are transferred to the style-transferred images. The quality of the generated images is evaluated both visually and by applying a no-reference image quality measure. In the case of land cover transfer, we also applied a classifier based on NDVI to make sure that the pixels of the generated images belong to the target terrain type. The novel task addressed in this paper is the application of modified cycleGAN to produce 13 bands while incorporating pansharpening techniques aimed at optimizing the overall image quality.

A possible direction for the future work is considering the application of the proposed techniques to other transfer types, such as day to night and cloud to no-cloud. Another interesting research direction is to directly include the sharpening step within the generative network. This can be done either by training the network with sharpened images or by introducing within the network some *ad hoc* layers in charge of sharpening. The development of a detector capable of distinguishing synthetic images from genuine ones is also worth further investigation.

Disclosures

No conflicts of interest exist.

Code and Data Availability

Data were downloaded from ESA Copernicus hub and further processed. It is available upon request to the corresponding author.

Acknowledgments

This paper is based on the research sponsored by the Defense Advanced Research Projects Agency and the Air Force Research Laboratory (Grant No. FA8750-20-2-1004). The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of DARPA or AFRL, or the U.S. Government

References

1. Mashable, "Satellite images show clearly that Russia faked its MH17 report," (May 2015). <http://mashable.com/2015/05/31/russia-fake-mh17-report> (accessed 11 March 2022).
2. G. Rannard, "Australia fires: misleading maps and pictures go viral," (January 2020). <https://www.bbc.com/news/blogs-trending-51020564> (accessed June 2022).
3. X. Yuan, J. Tian, and P. Reinartz, "Generating artificial near infrared spectral band from RGB image using conditional generative adversarial network," *ISPRS Ann. Photogramm. Remote Sens. Sp. Inf. Sci.* **V-3-2020**, 279–285 (2020).
4. G. Vivone et al., "A critical comparison among pansharpening algorithms," *IEEE Trans. Geosci. Remote Sens.* **53**(5), 2565–2586 (2015).
5. X. Meng et al., "Review of the pansharpening methods for remote sensing images based on the idea of meta-analysis: practical discussion and challenges," *Inf. Fusion* **46**, 102–113 (2019).
6. C. Ko, "On the performance of adaptive Gram–Schmidt algorithm for interference cancelling arrays," *IEEE Trans. Antenn. Propag.* **39**(4), 505–511 (1991).
7. I. J. Goodfellow et al., "Generative adversarial nets," in *Proc. 27th Int. Conf. Neural Inf. Process. Syst. - Vol. 2, NIPS'14*, Cambridge, MA, USA, MIT Press, pp. 2672–2680 (2014).
8. T. Karras et al., "Progressive growing of GANs for improved quality, stability, and variation," in *6th Int. Conf. Learn. Represent., ICLR 2018*, April 30–May 3, 2018, Vancouver, BC, Canada, OpenReview.net (2018).

9. T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *IEEE Conf. Comput. Vis. and Pattern Recognit., CVPR*, 16-20 June 2019, Long Beach, CA, USA, Computer Vision Foundation/IEEE, pp. 4401–4410 (2019).
10. T. Karras et al., "Analyzing and improving the image quality of styleGAN," in *IEEE/CVF Conf. Comput. Vis. and Pattern Recognit., CVPR*, 13-19 June 2020, Seattle, WA, USA, Computer Vision Foundation/IEEE, pp. 8107–8116 (2020).
11. P. Isola et al., "Image-to-image translation with conditional adversarial networks," in *IEEE Conf. Comput. Vis. and Pattern Recognit., CVPR*, 21–26 July 2017, Honolulu, HI, USA, IEEE Computer Society, pp. 5967–5976 (2017).
12. J. Zhu et al., "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *IEEE Int. Conf. Comput. Vis., ICCV*, 22–29 October 2017, Venice, Italy, IEEE Computer Society, pp. 2242–2251 (2017).
13. A. Odena, C. Olah, and J. Shlens, "Conditional image synthesis with auxiliary classifier GANs," in *Proc. 34th Int. Conf. Mach. Learn., ICML*, 6–11 August 2017, D. Precup and Y. W. Teh, Eds., Sydney, NSW, Australia, PMLR, Vol. 70, pp. 2642–2651 (2017).
14. S. A. Israel et al., "Generative adversarial networks for classification," in *IEEE Appl. Imagery Pattern Recognit. Workshop, AIPR*, 10–12 October 2017, Washington, DC, USA, IEEE Computer Society, pp. 1–4 (2017).
15. Q. H. Le et al., "GAN mask R-CNN: instance semantic segmentation benefits from generative adversarial networks," CoRR abs/2010.13757 (2020).
16. M. F. Reyes et al., "SAR-to-optical image translation based on conditional generative adversarial networks: optimization, opportunities and limits," *Remote. Sens.* **11**(17), 2067 (2019).
17. T. Vandal et al., "Spectral synthesis for geostationary satellite-to-satellite translation," *IEEE Trans. Geosci. Remote. Sens.* **60**, 1–11 (2022).
18. H. J. A. Andrade and B. J. T. Fernandes, "Synthesis of satellite-like urban images from historical maps using conditional GAN," *IEEE Geosci. Remote. Sens. Lett.* **19**, 1–4 (2022).
19. G. Baier et al., "Building a parallel universe image synthesis from land cover maps and auxiliary raster data," CoRR abs/2011.11314 (2020).
20. Z. Wang et al., "Ultra-dense GAN for satellite imagery super-resolution," *Neurocomputing* **398**, 328–337 (2020).
21. X. Liu, Y. Wang, and Q. Liu, "PSGAN: a generative adversarial network for remote sensing image pan-sharpening," in *IEEE Int. Conf. Image Process., ICIP*, 7–10 October 2018, Athens, Greece, IEEE, pp. 873–877 (2018).
22. Y. Zhan et al., "Semisupervised hyperspectral image classification based on generative adversarial networks," *IEEE Geosci. Remote. Sens. Lett.* **15**(2), 212–216 (2018).
23. L. Abady et al., "GAN generation of synthetic multispectral satellite images," *Proc. SPIE* **11533**, 122–133 (2020).
24. R. Chen et al., "Reusing discriminators for encoding: towards unsupervised image-to-image translation," in *IEEE/CVF Conf. Comput. Vis. and Pattern Recognit., CVPR*, 13-19 June 2020, Seattle, WA, USA, Computer Vision Foundation/IEEE, pp. 8165–8174 (2020).
25. L. Abady et al., "Manipulation and generation of synthetic satellite images using deep learning models," *J. Appl. Remote Sens.* **16**(4), 046504 (2022).
26. The European Space Agency, "Copernicus open access hub," (2021). <https://scihub.copernicus.eu/dhus/#/home>.
27. F. Warmerdam, "Gdal," The Open Source Geospatial Foundation, <https://gdal.org/> (2020).
28. O. Ronneberger, P. Fischer, and T. Brox, "U-Net: convolutional networks for biomedical image segmentation," *Lect. Notes Comput. Sci.* **9351**, 234–241 (2015).
29. D. P. Kingma and J. Ba, "Adam: a method for stochastic optimization," in *3rd Int. Conf. Learn. Represent., ICLR*, 7–9 May 2015, Y. Bengio and Y. LeCun, Eds., San Diego, CA, USA (2015).
30. The Organisation for Economic Co-operation and Development, "OECD," <https://stats.oecd.org/Index.aspx> (2020).
31. K. He et al., "Deep residual learning for image recognition," in *IEEE Conf. Comput. Vis. and Pattern Recognit., CVPR*, 27–30 June 2016, Las Vegas, NV, USA, IEEE Computer Society, pp. 770–778 (2016).
32. B. Aiazzi, S. Baronti, and M. Selva, "Improving component substitution pansharpening through multivariate regression of MS + PAN data," *IEEE Trans. Geosci. Remote. Sens.* **45**(10), 3230–3239 (2007).
33. N. Venkatanath et al., "Blind image quality evaluation using perception based features," in *Twenty First Natl. Conf. Commun., NCC*, 27 February–1 March 2015, Mumbai, India, IEEE, pp. 1–6 (2015).

Lydia Abady is currently a postdoctoral fellow in the Department of Information Engineering and Mathematics of the University of Siena. Her research interests are generative models, multimedia forensics, and security.

Mauro Barni is a full professor at the University of Siena. In the last two decades, he has been studying the application of image and signal processing for security applications. His current research interests include multimedia forensics, adversarial machine learning, and DNN watermarking. He published about 350 papers in international journals and conference proceedings. He is a fellow member of the IEEE and the AAIA, and a member of EURASIP.

Andrea Garzelli is a professor of telecommunications in the Department of Information Engineering and Mathematics, University of Siena, Italy. His main research interests include remote sensing image processing from optical and SAR sensors, change detection, and multi-sensor image fusion.

Benedetta Tondi is currently an assistant professor in the Department of Information Engineering and Mathematics of the University of Siena. Her research interest focuses on the multimedia forensics and counter-forensics and more in general adversarial signal processing, adversarial machine learning, and the security of deep learning techniques.