
Juan Mario Haut\textsuperscript{(a), Student Member, IEEE, Ruben Fernandez-Beltran\textsuperscript{(a), Mercedes E. Paoletti\textsuperscript{(a), Student Member, IEEE, Javier Plaza\textsuperscript{(a), Senior Member, IEEE, Antonio Plaza\textsuperscript{(a), Fellow, IEEE, and Filiberto Pla

Abstract—Super-resolution (SR) brings an excellent opportunity to improve a wide range of different remote sensing applications. SR techniques are concerned about increasing the image resolution while providing finer spatial details than those captured by the original acquisition instrument. Therefore, SR techniques are particularly useful to cope with the increasing demand remote sensing imaging applications requiring fine spatial resolution. Even though different machine learning paradigms have been successfully applied in SR, more research is required to improve the SR process without the need of external high-resolution (HR) training examples. This paper proposes a new convolutional generator model to super-resolve low-resolution (LR) remote sensing data from an unsupervised perspective. That is, the proposed generative network is able to initially learn relationships between the LR and HR domains through several convolutional, downsampling, batch normalization, and activation layers. Then, the data are symmetrically projected to the target resolution while guaranteeing a reconstruction constraint over the LR input image. An experimental comparison is conducted using 12 different unsupervised SR methods over different test images. Our experiments reveal the potential of the proposed approach to improve the resolution of remote sensing imagery.

Index Terms—Convolutional neural networks (CNNs), remote sensing, super-resolution (SR).

I. INTRODUCTION

REMOTE sensing image acquisition technology is under constant development and now provides improved imagery that are useful to tackle new challenges and needs [1]. Nonetheless, the increasing demand of highly accurate remote sensing imaging applications, such as fine-grained classification [2], [3], target recognition [4], [5], object tracking [6], [7], and detailed land monitoring [8], still makes the spatial resolution of optical sensors one of the most important limitations affecting remotely sensed imagery. In general, the spatial resolution of an instrument defines the pixel size covering the Earth surface, and, therefore, it describes the ability of the sensor to capture small image details. Even though the most technologically advanced satellites are able to discern spatial information within a squared meter on the Earth surface [9], the high cost of this acquisition technology, together with the light physical limitations when substantially decreasing the sensor pixel size, is usually important constraints that make algorithmic-based resolution enhancement techniques an excellent tool for remote sensing imaging applications [10].

The general objective in super-resolution (SR) [11]–[14] is to improve the image resolution beyond the sensor limits, that is, increasing the number of image pixels while providing finer spatial details than those captured by the original acquisition instrument. Depending on the number of input images, it is possible to distinguish between two kinds of SR methods, single-image [15] and multi-image [16]. Single-image SR techniques use a single image of the target scene to obtain the super-resolved output, whereas multi-image SR methods require several scene shots simultaneously acquired at different positions. In remote sensing, the single-image approach is usually adopted, because it provides a more general scheme to super-resolve any kind of imaging sensor without the need for a satellite constellation [17], [18].

The single-image SR approach can be considered as an ill-posed problem, since there is not a single solution for any given low-resolution (LR) pixel, i.e., the solution is not unique. This fact has been traditionally mitigated by constraining the space of possible solutions using a strong prior information extracted from a specific set of images. In this sense, artificial neural networks (ANNs) have become a powerful tool due to their ability to learn image priors from any given data set. Traditionally used in the pattern recognition field [19], ANNs...
have also been intensively used for the analysis of remotely sensed imagery [20]–[22], reaching a good performance without prior knowledge on the input data distribution and offering multiple training techniques.

With the great evolution of deep learning [23], [24] (DL) techniques, the ANN architecture has evolved from the simple linear perceptron classifier to deeper architectures (multilayer stack of simple modules) called deep neural networks (DNNs), allowing to create more complex models which can extract more abstract information (features) from the data than shallow ones [25]. DNNs are currently able to perform SR in a successfully way [26]. In particular, convolutional neural networks (CNNs) [23] stand out as a powerful image processing tool due their effectiveness, especially for the analysis of large sets of 2-D images. CNNs have proven to produce high performance in a great variety of tasks, such as image analysis and target detection [27]–[30], pan-sharpening [31], [32], reconstruction of remote sensing imagery [33], and also image SR [34]–[38]. However, these supervised techniques require sufficient high-resolution (HR) training examples in order to perform properly and generalize well. In addition, they usually tend to overfit quickly due to the models’ complexity and the lack of training data. Note that obtaining the relevant remote sensing training data is expensive and time-consuming. Besides, the number of available training remote sensing data sets is rather limited, and normally, they suffer from a lack of image variations and diversity. For these reasons, supervised learning is difficult to carry out, while unsupervised learning methods do not need any external data to train. On the other hand, the CNN is a very flexible model that can be adapted to different learning models, such as convolutional autoencoders [39], [40], convolutional deep belief networks [41], convolutional generative adversarial neural networks [42], convolutional recurrent neural networks [43], and fully convolutional networks [44], among others. In particular, we highlight the hourglass network [45], [46], whose topology is symmetric, related to the convolution–deconvolution architecture and also to the encoder–decoder, characterized by a first step of pooling down to a lower resolution (composed of convolutional and max pooling layers) and a second step of upsampling to a higher resolution and combining features across multiple resolutions.

Following the hourglass approach, a new unsupervised neural network model is proposed in this paper in order to super-resolve remote sensing images. The novelty of the proposed approach lies on using a generative random noise to introduce a higher variety of spatial patterns, which can be adapted to different learning models, such as convolutional autoencoders [39], [40], convolutional deep belief networks [41], convolutional generative adversarial neural networks [42], convolutional recurrent neural networks [43], and fully convolutional networks [44], among others. In particular, we highlight the hourglass network [45], [46], whose topology is symmetric, related to the convolution–deconvolution architecture and also to the encoder–decoder, characterized by a first step of pooling down to a lower resolution (composed of convolutional and max pooling layers) and a second step of upsampling to a higher resolution and combining features across multiple resolutions.

Following the hourglass approach, a new unsupervised neural network model is proposed in this paper in order to super-resolve remote sensing images. The novelty of the proposed approach lies on using a generative random noise to introduce a higher variety of spatial patterns, which can be adapted to different learning models, such as convolutional autoencoders [39], [40], convolutional deep belief networks [41], convolutional generative adversarial neural networks [42], convolutional recurrent neural networks [43], and fully convolutional networks [44], among others. In particular, we highlight the hourglass network [45], [46], whose topology is symmetric, related to the convolution–deconvolution architecture and also to the encoder–decoder, characterized by a first step of pooling down to a lower resolution (composed of convolutional and max pooling layers) and a second step of upsampling to a higher resolution and combining features across multiple resolutions.

The remainder of this paper is organized as follows. Section II presents an overview of single-image SR methods and their limitations. Section III describes the methodology used by the proposed convolutional generator model. Section IV validates the proposed approach by performing comparisons with different single-image SR methods. Finally, Section V concludes this paper with some remarks and hints at plausible future research lines.

II. BACKGROUND

A. Brief Single-Image SR Overview

Broadly speaking, single-image SR algorithms can be categorized into three different groups [53], [54]: image reconstruction (RE), image learning (LE), and hybrid (HY) methods. RE methods aim at reconstructing HR details in the super-resolved output assuming a specific degradation model along the image acquisition process, which is typically defined by the concatenation of three operators: blurring, decimation, and noise. Therefore, RE methods can be usually defined in terms of the three following stages (see Fig. 1): Stage 1, where the LR input image (I_{LR}) is upscaled to the target resolution (I_{HR}) using a regular interpolation kernel function. In Stage 2, some physical features are extracted from I_{LR} to estimate the singularities of the spatial details. Finally, Stage 3 aggregates both the interpolated image (I_{LR}) and the extracted LR features to obtain the final reconstructed result f_{SR}.

Each particular RE method makes its own assumptions about the imaging model and the reconstruction process to relieve the ill-posed nature of the SR problem. Some of the most popular RE approaches are iterative back projection (IBP) [55], gradient profile prior (GPP) [56], and point spread function (PSF) deconvolution [57]–[59]. The rationale behind IBP is based on iteratively refining an initial interpolation result by means of minimizing the reconstruction error between the LR input image and a simulated LR version of the

1The use of high-performance computing methods, including parallelization with accelerators such as field-programmable gate arrays and GPUs [47]–[49], or the distribution with clusters and clouds [50], [51], has demonstrated great utility for the classification of remote images [52].
super-resolved result. GP takes advantage of the fact that the shape of the gradient profiles tends to remain invariant across scales, and therefore, LR gradient can be used to reconstruct the output image sharpness. PSF deconvolution methods tackle the upsampling problem from a deblurring point of view, that is, they initially estimate the imaging model PSF and then they try to remove the interpolated image blur.

Regarding LE methods, this type of techniques are able to provide a more powerful SR scheme, because they learn the relationships between LR and HR domains from an external training set containing ground-truth HR images. As shown in Fig. 2, RE methods can be divided into three stages: in Stage 1, the relations between LR and HR components are learned from a specific training set. Stage 2 aims at estimating the HR components that are related to the LR input image structures. Finally, Stage 3 combines the estimated HR components to generate the final super-resolved result.

Over the past years, different machine learning paradigms have been successfully applied in LE-based SR. Sparse coding [60], neighborhood embedding [61], and mapping functions [62], [63] are among the most popular methods. In a nutshell, sparse coding-based techniques take advantage of the fact that natural images tend to be sparse when they are characterized as a linear combination of small patches. The neighborhood embedding approach assumes that small image patches of LR images describe a low-dimensional nonlinear manifold with a similar local geometry to their HR counterparts. Mapping-based techniques cope with the SR task as a regression problem between the HR and LR domains.

Finally, HY techniques work toward reaching an agreement between the RE and LE approaches. In particular, they perform a training process but only using the LR input image. The rationale behind the HY methods is based on the patch redundancy property pervading natural images, which assumes that natural images tend to contain repetitive structures within the same scale and over scales as well. Taking this principle into account, it is possible to find patches which appear in a lower scale, without any blurring or decimation, and then extracting their corresponding HR counterparts from the higher scale image. Eventually, the super-resolved image can be generated using the LR/HR relationships learned across scales. In particular, HY methods generally follow the scheme shown in Fig. 3: in Stage 1, the self-learning process is conducted, that is, several lower scale images are initially generated from \( I_{LR} \) and then those patches which tend to appear across scales are extracted. Stage 2 projects the input LR image to the target resolution using the relations previously learned. Finally, the final super-resolved result is generated in Stage 3 considering some sort of reconstruction constraint.

Logically, each specific HY approach defines its own assumptions about the imaging model and the patch searching criteria. For example, the work presented in [64] approximates the blur operator by a Gaussian kernel, and the patch redundancy process is conducted by an approximation of the nearest neighbor search. Other works propose different kinds of modifications over this scheme. It is the case of [65], which introduces a model extension to enable patch geometric transformations across scales. Therefore, the number of patch matches can be increased and consequently the amount of learned LR/HR relationships. In other works, such as in [66], the blur operator is estimated at the same time as the SR output is generated through an optimization process.

B. SR Limitations in Remote Sensing

Each single-image SR methodology has shown to be particularly effective under specific conditions [15], [54]. RE methods are able to reduce the noise as well as the blur and aliasing inherent to interpolation kernel functions. However, the lack of relevant high-frequency information in the LR input image limits their effectiveness to small magnification factors, which can be an important limitation for many of the currently operational (moderate) resolution satellites [67].

LE-based techniques potentially overcome these drawbacks by learning the relationships between LR and HR domains from an external training set. Nonetheless, the availability of suitable HR training examples can also be a serious constraint.
for many satellites. Note that the ground-truth HR images are usually not available in real scenarios, and this may lead to an unrepresentative training phase with a biased super-resolved result. Eventually, the application of LE-based SR methods in actual ground segment production environments is rather limited [68].

HY methods offer the advantage of not requiring any external training set to learn the LR/HR relationships by taking advantage of the patch redundancy property over scales. However, the probability of finding patches satisfying this property decreases with the input resolution, and therefore, the amount of useful LR/HR connections over scales highly depends on the input image.

With all these considerations in mind, unsupervised RE and HY methods are especially attractive to remote sensing. While supervised approaches use a training set of HR images to learn the relationships between the LR and HR domains [35], [69], [70], unsupervised approaches only make use of the target LR image to generate the corresponding super-resolved output result. Moreover, supervised network architectures implement a regressor function to project general LR image patches onto the HR domain. However, in a real-life remotely sensed data production environment, there is no actual HR captured by the sensor. In this sense, unsupervised methods do not require the availability of HR images to train a general SR model, super-resolving each specific LR input image without using any other external data and providing the opportunity to offer new super-resolved data products in satellite and airborne missions that use relatively inexpensive sensors without the need of using any external HR training set. Nevertheless, the number of works in the remote sensing literature dealing with the unsupervised SR problem is rather constrained, and this is precisely the gap that motivates this paper.

Mianji et al. [71] proposed an SR approach using a backpropagation neural network as a regression function and basing on: 1) spectral unmixing; 2) SR mapping; and 3) self-training, which is exploited taking advantage of the embedding provided by the spectral unmixing process itself. However, this approach could be highly affected by the spectral simplex geometry of the input image [72]. In contrast, an HY (also called self-learning) SR scheme has been proposed in this paper to super-resolve remote sensing data from an unsupervised perspective, basing on a new end-to-end convolutional generator model. The rationale behind the proposed approach is based on learning the relationships between the LR and HR domains by downsampling the original input image to a lower scale and then using the learned relations at a lower scale to project the LR input image to the target resolution. However, the amount of spatial information that it is possible to retrieve from a downsampled LR image may be limited, so a random generative noise has been additionally introduce together with a global reconstruction constraint to activate a higher amount of consistent spatial variations along the SR process. That means, random spatial variations are initially generated to be introduced in the self-learning process in order to mitigate the ill-posed nature of the SR problem. Regarding the proposed network global scheme, it provides a similar end-to-end framework to other DL-based approaches, e.g., [35], [69], [70], where the original LR image is used to learn the downsampling filters at the same time that they are also used to generate the super-resolved output.

III. METHODOLOGY

Traditionally, a generator network is an algorithm for image generation, where given a random variable \( z \), the model is able to learn internal relationships (represented by the model parameters \( \theta \)) to generate an image \( X = f_\theta(z) \), i.e., a regression problem. This allows us to learn the distribution of the data and the correlations between \( z \) and \( X \). We can follow this approach in order to perform SR over remote sensing images, where \( z \in \mathbb{R}^{C \times W \times H} \) is a random noise and \( X \in \mathbb{R}^{3 \times W \times H} \) is the desired RGB HR image.

Given an LR image \( X^{LR} \in \mathbb{R}^{3 \times W \times H} \), the SR’s goal is to improve the image resolution beyond the sensor limits obtaining an HR version \( X^{HR} \in \mathbb{R}^{3 \times W \times H} \) from \( X^{LR} \), where \( t \) is the resolution factor and \( W < t \cdot W, H < t \cdot H \). In order to do this, a deep model based on CNNs has been implemented. This kind of networks is composed of the layers that are applied over defined regions of the input data, i.e., they are locally connected to the input, transforming the input volume to an output volume of neuron activations which will serve as an input to the next layer. The fact that each layer is not completely connected to the previous layer (only with a patch/window defined as the receptive field) is a great advantage for data analysis, thus reducing the number of connections in the network, where each layer composes feature extraction stages working as a filter or kernel over patches of the input volume.

Depending on the treatment of the data, CNNs can be classified into three categories. Supposing that \( x^{(i)} \in \mathbb{R}^{C} = [x^{(i)}_1, x^{(i)}_2, \ldots, x^{(i)}_C] \) is a pixel with \( C \) spectral bands of image \( X \in \mathbb{R}^{C \times W \times H} \), with \( i = 1, 2, \ldots, W \cdot H \), while \( p^{(j)} \in \mathbb{R}^{b \times p \times p} \) is a patch of \( X \), where \( p \) is the width and height (with \( p \leq W \) and \( p \leq H \)) and \( b \) is the number of spectral bands of the patch (with \( b \leq C \)). 1-D-CNN models take separate as input data each pixels vector \( x^{(i)} \), extracting only spectral information [73]. On the other hand, 2-D-CNNs extract spatial information, taking as input data the entire image \( X \) [74] or image patches \( p^{(j)} \) [75], where \( C \) and \( b \) are set to small values, i.e., the spectral information is not very relevant compared with the spatial information. Finally, 3-D-CNNs extract spectral–spatial information, taking normally as input data patches \( p^{(j)} \) of the original image \( X \) [29], [30], where \( C \) and \( b \) are set to large values, i.e., the spectral information is very relevant and it is combined with spatial information. Usually, for panchromatic and RGB remote sensing images, a 2-D-CNN approach is taken, while 1-D- and 3-D-CNNs are usually for multispectral and hyperspectral images. This paper works with the RGB remote sensing data sets, so the 2-D-CNN architecture has been implemented to take advantage of the spatial information contained in the images. It is composed of five different kinds of layers, as described in the following.

1) Convolution (CONV) Layer: This kind of layer is composed of a block of neurons where each slice (also called
a filter or a kernel) shares its weights and biases between all the neurons that compose it. Given a CONV layer $C^{(i)}$, its output volume $O^{(i)}$ (also called feature maps) can be calculated by the following (1) as the dot product between the $n^{(i)}$ slices’ weights $W^{(i)}$ and biases $B^{(i)}$ (where $n^{(i)}$ is the number of depth slices, also known as number of filters or kernels) and a small region of the input volume $O^{(i-1)}$, i.e., a rectangular section of the previous layer $C^{(i-1)}$ defined by the kernel size $k^{(i)}$ of the current layer $C^{(i)}$:

$$O^{(i)} = (O^{(i-1)} \cdot W^{(i)})_{f,l} + B^{(i)}$$

$$= \sum_{m=1}^{k^{(i)}} \sum_{n=1}^{k^{(i)}} (O^{(i-1)}_{(f-m),l-n} \cdot w_{m,n}^{(i)}) + B^{(i)}$$

where $O^{(i-1)}_{f,l}$ is the feature $(f,l)$ of the feature map $O^{(i-1)} \in \mathbb{R}^{W,H}$, with $f = 1, 2, \ldots, W$ and $l = 1, 2, \ldots, H$, and $w_{m,n}^{(i)}$ is the weight $(m,n)$ of the weight matrix $W^{(i)} \in \mathbb{R}^{k^{(i)},k^{(i)}}$. As a result, $O^{(i)} \in \mathbb{R}^{n^{(i)},W,H}$ forms a data cube whose depth is defined by the number of kernels $n^{(i)}$ (that indicates the number of output feature maps) and its width and height are calculated as

$$W' = \frac{(Wk + 2P)}{S} + 1$$

$$H' = \frac{(Hk + 2P)}{S} + 1$$

respectively, where $P$ indicates the padding (zeros) added to the input data borders and $S$ indicates the stride of the kernel over the data. $W$ and $H$ are, respectively, the width and height of the previous feature maps $O^{(i-1)} \in \mathbb{R}^{n^{(i-1)},W,H}$.

2) **Batch Normalization (BATCH-NORM) Layer:** Normally, it is placed behind the convolution layer, and it applies the normalization defined by (2) over the batch data

$$y = \frac{x - \text{mean}[x]}{\sqrt{\text{Var}[x] + \epsilon}} \cdot \gamma + \beta$$

where $\gamma$ and $\beta$ are the learnable parameter vectors, and $\epsilon$ is a parameter for numerical stability.

3) **Activation Layer:** After CONV and BATCH-NORM layers, the activation layer or the nonlinearity layer embeds a nonlinear function that is applied over the output of the previous layer as the rectified linear unit (ReLU) [76], [77]. In this case, the LeakyReLU function is implemented [78]

$$f(x) = \begin{cases} 
        x & \text{if } x > 0 \\
        ax & \text{if } x \leq 0
\end{cases}$$

where $\alpha$ is a small nonzero parameter, normally 0.001.

4) **Downsampling/Upsampling Layer:** The proposed model also implements downsampling and upsampling layers at certain locations of the architecture. The first one reduces the spatial resolution of the input volumes by reducing the width and height with a resolution factor $t$. A max pool function is generally implemented to perform the downsampling, and however, the proposed model downsamples the input data setting the strides of certain CONV layers to $S = 2$. In addition, the upsampling layers try to reconstruct the data size using the bilinear function given a scaling factor.

The proposed methodology provides a novel approach to effectively super-resolve remote sensing data from an unsupervised perspective. Specifically, our model receives the random noise vector $z$ as input data, which is resized into a cube matrix $\mathbb{R}^{C \times t \times W \times H}$ in order to feed the network, where $W$ and $H$ are the width and height of the original LR remote sensing image, $C = 3$ is the number of spectral channels, and $t$ is the resolution factor. Following a fully connected hourglass architecture [45], [79], $z$ goes through two main steps composed of several blocks as follows.

1) The downsampling step is composed of $N$ blocks of layers, called $d^{(i)}$ ($i = 1, 2, \ldots, N$), where the input of each one is the feature maps of the previous one. Each $d^{(i)}$ is composed of an initial CONV layer $C^{(1)}_{d^{(i)}}$ that performs the downsampling step by its stride $S = 2$, dividing the output volume size by two. This output volume feeds the BATCH-NORM layer and the non-linear LeakyReLU activation function. The output of the neuron activations feeds the second CONV layer $C^{(2)}_{d^{(i)}}$ without downsampling (i.e., $S = 1$) and also followed by a BATCH-NORM layer and the LeakyReLU activation function. $C^{(1)}_{d^{(i)}}$ and $C^{(2)}_{d^{(i)}}$ have their own number of filters ($n^{(1)}_{d^{(i)}}$ and $n^{(2)}_{d^{(i)}}$) and their own kernel size ($k^{(1)}_{d^{(i)}}$ and $k^{(2)}_{d^{(i)}}$). In fact, each block $d^{(i)}$ is reducing the space information, i.e., generating a low spatial resolution data that will feed the second upsampling step.

2) The upsampling step is symmetric to downsampling one, and it is also composed of $N$ blocks of layers, called $u^{(i)}$ ($i = N, \ldots, 2, 1$), where the input of each one is the output of the previous one. In this case, each $u^{(i)}$ is composed of several stacked layers. The first one is a BATCH-NORM layer, followed by the first CONV layer $C^{(1)}_{u^{(i)}}$ (which maintains the size of the data, i.e., $S = 1$) and its BATCH-NORM and LeakyReLU function. The output of the neuron activations feeds the second convolutional layer $C^{(2)}_{u^{(i)}}$ (which also maintains the size of the data). After the BATCH-NORM and the activation function, the output will finally feed the bilinear upsampling layer with factor equal to 2. Again, $C^{(1)}_{u^{(i)}}$ and $C^{(2)}_{u^{(i)}}$ have their own number of filters ($n^{(1)}_{u^{(i)}}$ and $n^{(2)}_{u^{(i)}}$) and their own kernel size ($k^{(1)}_{u^{(i)}}$ and $k^{(2)}_{u^{(i)}}$). Both the steps, downsampling and upsampling, are symmetrical and connected by skip connections, i.e., the input of each upsampling block $u^{(i)}$ is combined with the corresponding $d^{(i)}$ through the skip connection $s^{(i)}$ ($i = 1, 2, \ldots, N$) composed of a CONV layer $C^{(i)}_{s^{(i)}}$, with its number of filters $n^{(i)}_{s^{(i)}}$ and its kernel size $k^{(i)}_{s^{(i)}}$, a BATCH-NORM layer, and the activation function, LeakyReLU. In fact, the output of $s^{(i)}$ is concatenated to the output of $u^{(i)}$. The chosen topology is shown in Fig. 4. At the end of the topology, an output block is added, composed of a CONV layer and a sigmoid function at the end. As a result, an HR image $X^{HR}_{o} \in \mathbb{R}^{3 \times t \times W \times H}$ is generated as an output of the network.
In particular, the SR’s goal is to generate an HR image from an LR one, minimizing the following cost function:

\[
\min || \phi(X_{HR}) - X_{LR} ||^2. \tag{4}
\]

In fact, our remote sensing data sets are composed of HR images. However, we cannot use them, because they cannot be considered as ground truth to perform SR. In order to solve this, an LR version is generated from each HR image by a downsampler \( \phi : [R^{3 \times W \times H}] \rightarrow [R^{3 \times W \times H}] \), so \( X_{LR} = \phi(X_{HR}) \). In our case, the downsampler \( \phi \) has been implemented using Lanczos3 resampling [80], where pixels of the original image \( X_{HR} \) are passed into an algorithm that averages their color/alpha using sinc functions. With this LR version, we can perform the SR task. However, the model is generating an HR image, \( X_{HR}' \). In order to solve this, the downsampler function \( \phi \) is applied over \( X_{HR}' \). At the end, (4) can be rewritten as

\[
\min || \phi(X_{HR}) - \phi(X_{HR}') ||^2 \rightarrow \min || X_{LR} - X_{HR}' ||^2. \tag{5}
\]

The cost function defined by (5) is optimized iteratively by the model via an Adam optimizer [81]. The proposed method is summarized in Algorithm 1. Also, in Fig. 8, we can observe the \( X_{HR}' \) image generated by the model at each epoch.

In order to test the proposed model, two networks have been implemented. The first one performs a two times SR over an LR image \( X_{LR} \in [R^{3 \times W \times H}] \), i.e., the resolution factor is set to \( t = 2 \), obtaining a \( X_{HR} \in [R^{3 \times 2 \times W \times 2 \times H}] \) HR image, and the second one performs a four times SR, i.e., \( t = 4 \) obtaining a \( X_{HR} \in [R^{3 \times 4 \times W \times 4 \times H}] \) HR image. Following the scheme presented in Fig. 4, both the models have been implemented with the topology described in Tables I and II.

Algorithm 1: Unsupervised Remote Sensing Single-Image SR Algorithm

1: \( \text{procedure} \ SR\_\text{MODEL}(X_{LR}, t) \) \( \triangleright \)
2: \( X_{LR} \in [R^{C \times W \times H}] \) original low resolution remote sensing image, \( t \) resolution factor
3: \( z \leftarrow \text{Random noise with size } C \times t \cdot W \times t \cdot H \)
4: \( \text{repeat} \)
5: \( X_{HR}' \leftarrow \text{model\_net}(z) \)
6: \( X_{LR} \leftarrow \phi(X_{HR}') \) \( \triangleright \phi \) is Lanczos3
7: \( \text{loss} = \text{MSE}(X_{HR}' \times X_{LR}) \)
8: \( \text{ADAM\_Optimizer(loss)} \)
9: \( z \leftarrow X_{HR}' \)
10: \( \text{until} \) Reach maximum epoch
11: \( \text{return} X_{HR}' \)
12: \( \text{end procedure} \)

A. Metrics

In order to compare the properties of the obtained \( X_{HR}' \) image with regard to the original remote sensing image \( X_{HR} \), several evaluation metrics have been used. For the sake of simplicity, we rename \( X_{HR}' = X_0 \) and \( X_{HR} = X \), where \( x_0 \) and \( x \) are the \( i \)th pixels of \( X_0 \) and \( X \), respectively.

Following (6), where \( n_{\text{samples}} \) is the number of pixels of \( X \) and \( X_{\text{max}} \) and \( X_{\text{min}} \) are the maximum and minimum
TABLE I
NETWORK TOPOLOGY FOR TWO-TIME SR. THE UPSAMPLING PHASE HAS BEEN PERFORMED WITH A SCALE-FACTOR SET TO 2

<table>
<thead>
<tr>
<th>Block ID</th>
<th>CONV ID</th>
<th>Kernel size</th>
<th>Number of kernels</th>
<th>Stride</th>
</tr>
</thead>
<tbody>
<tr>
<td>d(1)</td>
<td>C4d(1)</td>
<td>3 x 3</td>
<td>256</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>C4d(2)</td>
<td>3 x 3</td>
<td>256</td>
<td>1</td>
</tr>
<tr>
<td>d(2)</td>
<td>C4d(1)</td>
<td>3 x 3</td>
<td>256</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>C4d(2)</td>
<td>3 x 3</td>
<td>256</td>
<td>1</td>
</tr>
<tr>
<td>Bottle-neck connection</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>u(3)</td>
<td>C4u(1)</td>
<td>3 x 3</td>
<td>256</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>C4u(2)</td>
<td>3 x 3</td>
<td>256</td>
<td>1</td>
</tr>
<tr>
<td>Up-sampling connections</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>u(1)</td>
<td>C4u(1)</td>
<td>5 x 5</td>
<td>256</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>C4u(2)</td>
<td>1 x 1</td>
<td>256</td>
<td>1</td>
</tr>
<tr>
<td>u(0)</td>
<td>C4u(1)</td>
<td>5 x 5</td>
<td>256</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>C4u(2)</td>
<td>1 x 1</td>
<td>256</td>
<td>1</td>
</tr>
<tr>
<td>Output connections</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>u(1)</td>
<td>C4u(1)</td>
<td>3 x 3</td>
<td>256</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>C4u(2)</td>
<td>3 x 3</td>
<td>256</td>
<td>1</td>
</tr>
<tr>
<td>Skip connections</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>z(1)</td>
<td>C4z(1)</td>
<td>1 x 1</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>z(2)</td>
<td>C4z(2)</td>
<td>1 x 1</td>
<td>3</td>
<td>1</td>
</tr>
</tbody>
</table>

values of image X, respectively, the normalized root mean square error (NRMSE) measures the distance between the data predicted by a model \(X_o\) and the original data observed from the environment X that we want to model

\[
\text{NRMSE}(X, X_o) = \sqrt{\frac{1}{n_{\text{samples}}} \sum_{i=0}^{n_{\text{samples}}} (x^{(i)} - x_o^{(i)})^2} \left( X_{\text{max}} - X_{\text{min}} \right). \tag{6}
\]

Peak signal-to-noise ratio (PSNR) \([82]\) represents a better image quality than NRMSE. This metric is defined as the standard index for SR, where \(\text{MAX}_f\) is the maximum signal value that exists in the original X image. A higher PSNR value indicates that the reconstructed image \(X_o\) is of higher quality

\[
\text{PSNR}(X, X_o) = 20 \cdot \log_{10} \frac{\text{MAX}_f}{\text{RMSE}(X, X_o)}. \tag{7}
\]

Spectral angle mapper (SAM) \([83]\) calculates the angle between the corresponding pixels of the super-resolved image \(X_o\) and the original image X in the domain \([0, \pi]\]

\[
\text{SAM}(X, X_o) = \frac{1}{n_{\text{samples}}} \sum_{i=0}^{n_{\text{samples}}} \arccos \frac{x^{(i)} \cdot x_o^{(i)}}{\|x^{(i)}\| \cdot \|x_o^{(i)}\|}. \tag{8}
\]

The universal image quality index, also called Q-index, gathers three different properties in the image evaluation: 1) correlation; 2) luminance; and 3) contrast

\[
Q(X, X_o) = \sum_{j} \left( \frac{a}{\sigma_{X} \sigma_{X_o}} + \frac{b}{2 \sigma_{X} \sigma_{X_o}} (x^{(i)} - x_o^{(i)})^2 + \frac{c}{(\sigma_{X}^2 - \sigma_{X_o}^2)} \right). \tag{9}
\]

An extension of Q-index is the structural similarity (SSIM) \([84]\), a well-known quality metric used to measure the similarity between two images. It is a combination of three factors (loss correlation, luminance distortion, and contrast distortion)

\[
\text{SSIM}(X, X_o) = \frac{(2\mu_X \mu_{X_o} + c_1) \cdot (2\sigma_{X} x_{X_o} + c_2)}{(\mu_{X^2} + \mu_{x_{X_o}^2} + c_1) \cdot (\sigma_{X}^2 + \sigma_{X_o}^2 + c_2)}. \tag{10}
\]

Erreur relative globale adimensionnelle de synthèse (ERGAS) \([85]\) measures the quality of obtained \(X_o\) taking into account the scaling factor to evaluate the super-resolved image

\[
\text{ERGAS}(X, X_o) = \frac{100}{n_{\text{bands}}} \frac{1}{n_{\text{samples}}} \sum_{i=0}^{n_{\text{samples}}} \left( \frac{\text{RMSE}(x^{(i)}, x_o^{(i)})}{x^{(i)}}, \right)^2. \tag{11}
\]

TABLE II
NETWORK TOPOLOGY FOR FOUR-TIME SR. THE UPSAMPLING PHASE HAS BEEN PERFORMED WITH A SCALE-FACTOR SET TO 2

<table>
<thead>
<tr>
<th>Block ID</th>
<th>CONV ID</th>
<th>Kernel size</th>
<th>Number of kernels</th>
<th>Stride</th>
</tr>
</thead>
<tbody>
<tr>
<td>d(1)</td>
<td>C4d(1)</td>
<td>3 x 3</td>
<td>256</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>C4d(2)</td>
<td>3 x 3</td>
<td>256</td>
<td>1</td>
</tr>
<tr>
<td>d(2)</td>
<td>C4d(1)</td>
<td>3 x 3</td>
<td>256</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>C4d(2)</td>
<td>3 x 3</td>
<td>256</td>
<td>1</td>
</tr>
<tr>
<td>Bottle-neck connection</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>u(3)</td>
<td>C4u(1)</td>
<td>3 x 3</td>
<td>256</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>C4u(2)</td>
<td>3 x 3</td>
<td>256</td>
<td>1</td>
</tr>
<tr>
<td>Up-sampling connections</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>u(1)</td>
<td>C4u(1)</td>
<td>3 x 3</td>
<td>256</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>C4u(2)</td>
<td>3 x 3</td>
<td>256</td>
<td>1</td>
</tr>
<tr>
<td>u(0)</td>
<td>C4u(1)</td>
<td>3 x 3</td>
<td>256</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>C4u(2)</td>
<td>3 x 3</td>
<td>256</td>
<td>1</td>
</tr>
<tr>
<td>Output connections</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>u(1)</td>
<td>C4u(1)</td>
<td>3 x 3</td>
<td>256</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>C4u(2)</td>
<td>3 x 3</td>
<td>256</td>
<td>1</td>
</tr>
<tr>
<td>Skip connections</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>z(1)</td>
<td>C4z(1)</td>
<td>3 x 3</td>
<td>256</td>
<td>2</td>
</tr>
<tr>
<td>z(2)</td>
<td>C4z(2)</td>
<td>3 x 3</td>
<td>256</td>
<td>1</td>
</tr>
</tbody>
</table>

Erreur relative globale adimensionnelle de synthèse (ERGAS) \([85]\) measures the quality of obtained \(X_o\) taking into account the scaling factor to evaluate the super-resolved image

\[
\text{ERGAS}(X, X_o) = \frac{100}{n_{\text{bands}}} \frac{1}{n_{\text{samples}}} \sum_{i=0}^{n_{\text{samples}}} \left( \frac{\text{RMSE}(x^{(i)}, x_o^{(i)})}{x^{(i)}}, \right)^2. \tag{11}
\]
IV. Experiments

A. Experimental Configuration and Data Sets

In order to test the performance of the proposed model, several experiments have been conducted using two different hardware environments as follows.

1) A GPU environment composed of a sixth Generation Intel Core i7-6700K processor with 8 M of Cache and up to 4.20 GHz (4 cores/8-way multitask processing), 40 GB of DDR4 RAM with a serial speed of 2400 MHz, a GPU NVIDIA GeForce GTX 1080 with 8-GB GDDR5X of video memory and 10 Gb/s of memory frequency, a Toshiba DT01ACA HDD with 7200 rpm and 2 TB of capacity, and an ASUS Z170 pro-gaming motherboard. The software environment is composed of Ubuntu 16.04.4 x64 as an operating system, Pytorch [86] 0.3.0, and compute device unified architecture (CUDA) 8 for GPU functionality.

2) A CPU environment composed of Intel Core i7-4790 @ 3.60 GHz, 16 GB of DDR3 RAM with a serial speed of 800 MHz, and a Western Digital HDD with 7200 rpm and 1 TB of capacity. The software environment is composed of Windows 7 as an operating system and MATLAB R2013a.

It should be noted that our proposed method has been executed on the GPU environment, while the other methods have been executed in the CPU environment. Although our method uses Pytorch and CUDA, its parallelization can still be further optimized, and therefore, the difference in computation times with regard to the other methods was not very significant.

In addition, the employed database is composed of multiple RGB images from three different remote sensing repositories with the aim of testing the SR approach process under different sensor’s acquisition conditions and including different kinds of small perturbations. No additional levels of noise have been considered due to the design of the proposed SR approach, given by the noise-free scheme of (4), presented in other approaches, such as [35], [69], [70] and [87]. The employed repositories are described in the following and are publicly available on this repository (see Fig. 5).

1) UCMERED [88]: It is composed of 21 land-use classes, including agricultural, airplane, baseball diamond, beach, buildings, chaparral, dense residential, forest, freeway, golf course, harbor, intersection, medium density residential, mobile home park, overlap, parking lot, river, runway, sparse residential, storage tanks, and tennis courts images. Each class consists of 100 images with 256 × 256 pixels and a pixel resolution of 30.

2) RSCNN7 [89]: This data set contains 2800 images with seven different classes. The data set is rather challenging due to the wide differences of the scenes, which have been captured under changing seasons and varying weathers and sampled with different scales. The resolution of individual images is 400 × 400 pixels.

3) NWPU-RESIS45 [90]: The remote sensing image scene classification data set has been created by Northwestern Polytechnical University. This data set has 45 scenes with a total number of 31500 images, 700 per class.

From these images, an LR version has been generated from their corresponding HR counterparts following a two-step procedure [91]: 1) an initial blurring step and 2) a final decimation process. In particular, a Lanczos3 windowed sinc filter has been used for blurring the corresponding HR images, and then, these images have been downsampled according to the considered scaling factors (2 and 4, respectively). Regarding the blurring step, it should be noted that the Lanczos3 kernel size has been adapted to the scaling factor using the following expression, \( w = (4s + 1) \), where \( w \) represents the filter width and \( s \) is the considered scaling factor. For the downsampling process, image rows and columns have been selected from the top-left corner using a stride equal to the considered scaling factor. The goal behind this preprocessing step is to generate LR images from ground-truth HR ones maintaining the acquisition sensor properties but considering a lower spatial resolution. In this way, it has been possible to conduct a full-reference assessment protocol in experiments.

Fig. 5. Data set used in the experiments, comprising the following images: agricultural, agricultural2, airplane, baseball, bridge, circular-farmland, harbor, industry, intersection, parking, residential, and road.

<table>
<thead>
<tr>
<th>Identifier</th>
<th>SR type</th>
<th>Method description</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>BCI</td>
<td>Baseline</td>
<td>Bi-cubic interpolation</td>
<td>[81]</td>
</tr>
<tr>
<td>IBP</td>
<td>Reconstruction</td>
<td>Iterative back projection</td>
<td>[55]</td>
</tr>
<tr>
<td>GPP</td>
<td>Reconstruction</td>
<td>Gradient profile prior</td>
<td>[56]</td>
</tr>
<tr>
<td>SRI</td>
<td>Hybrid</td>
<td>Scale redundancy image</td>
<td>[64]</td>
</tr>
<tr>
<td>LSE</td>
<td>Hybrid</td>
<td>Local self-examples</td>
<td>[93]</td>
</tr>
<tr>
<td>GPR</td>
<td>Reconstruction</td>
<td>Gaussian process regression</td>
<td>[94]</td>
</tr>
<tr>
<td>BDB</td>
<td>Hybrid</td>
<td>Blind deblurring</td>
<td>[66]</td>
</tr>
<tr>
<td>DLU</td>
<td>Reconstruction</td>
<td>Deconvolution using Lucy-Richardson method</td>
<td>[57]</td>
</tr>
<tr>
<td>DRE</td>
<td>Reconstruction</td>
<td>Deconvolution using regularized filters</td>
<td>[58]</td>
</tr>
<tr>
<td>PSR</td>
<td>Reconstruction</td>
<td>Fast super-resolution</td>
<td>[95]</td>
</tr>
<tr>
<td>TSE</td>
<td>Hybrid</td>
<td>Transformed self-exemplars</td>
<td>[65]</td>
</tr>
<tr>
<td>UMK</td>
<td>Reconstruction</td>
<td>Unsharp masking</td>
<td>[59]</td>
</tr>
<tr>
<td>Ours</td>
<td>Generative-HY</td>
<td>The proposed approach</td>
<td>-</td>
</tr>
</tbody>
</table>

TABLE III

METHODS CONSIDERED FOR THE EXPERIMENTS. FURTHER DETAILS CAN BE FOUND IN THE CORRESPONDING REFERENCES

https://github.com/mhaut/images-superresolution
The performance of the proposed approach has been compared with the results obtained by 11 different unsupervised SR methods available in the literature, as well as the bicubic interpolation kernel function [80] used as an upscaling baseline. These SR methods have been considered for the experimental discussion, because they provide an unsupervised SR scheme in the same way that the proposed approach does, using the LR input image to generate a super-resolved output result. In addition, two different scaling factors, two times and four times, have been tested over the considered image data set (see Section IV-A). Table III provides a brief description of the SR techniques considered in the experimental part of this paper.

All the tested methods have been downloaded from the following website, and they have been used considering the default settings suggested by the methods’ authors for each particular scaling ratio [54]. Note that this configuration provides the most general scenario to super-resolve a wide range of image types taking into account the tested image diversity.

### C. Discussion

According to the quantitative assessment reported in Tables IV and V, it is possible to rank the global performance of the tested SR methods into three different categories:
1. high performance: for the proposed approach; TSE and SRI,
2. moderate performance: for IBP, DLU, DRE, and UMK; and
3. low performance: for GPP, LSE, GPR, BDB, and FSR.

When considering a two-time scaling factor, the proposed approach (together with the HY methods TSE and SRI) provides a significant improvement with respect to the BCI baseline. Specifically, the proposed approach obtains the best performance for NRMSE, PSNR, and ERGAS metrics, whereas TSE exhibits the best result for $Q$-index, SSIM, and SAM. Although TSE and SRI also achieve, on average, a remarkable improvement over the baseline, the proposed approach provides a more consistent performance, because it obtains the best average result for NRMSE, PSNR, and ERGAS metrics, and the second best value for $Q$-index, SSIM, and SAM. It can be observed that the average PSNR gain provided by the proposed approach is 0.39 dB for two times and 0.48 dB for four times. Regarding the methods providing a moderate improvement 2), the PSF deconvolution-based techniques, DLU, DRE, and UMK, provide a similar average performance, and IBP is able to obtain a slightly better quantitative result over all the considered metrics. Within the low-performance method group 3), it is possible to see that GPP and LSE methods provide a result similar to the one obtained by the baseline, and GPR, BDB, and FSR obtain even a worse result.

A similar trend can be observed when considering a four-time scaling factor. In this case, the proposed approach is, on average, the best method according to NRMSE, PSNR, and ERGAS metrics. TSE obtains the best $Q$-index and SSIM results, and both the methods obtain a similar average result for the SAM metric. It should be noted that SRI performance has worsened when using a four-time ratio, and however, it still obtains the third best $Q$-index and SSIM results. With respect to the rest of the moderate 2) and low-performance methods 3), they obtain similar results with regard to the ones obtained with a two-time factor. Overall, the proposed approach and

### TABLE IV

<table>
<thead>
<tr>
<th>Image</th>
<th>Method</th>
<th>Ratio 2x</th>
<th>Ratio 4x</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>TIME</td>
<td>NRMSE</td>
<td>PSNR</td>
</tr>
<tr>
<td>BCI</td>
<td>0.01</td>
<td>0.0506</td>
<td>28.11</td>
</tr>
<tr>
<td>IBP</td>
<td>0.15</td>
<td>0.0455</td>
<td>29.01</td>
</tr>
<tr>
<td>GPP</td>
<td>0.30</td>
<td>0.0591</td>
<td>28.20</td>
</tr>
<tr>
<td>SRI</td>
<td>33.77</td>
<td>0.0393</td>
<td>30.23</td>
</tr>
<tr>
<td>LSE</td>
<td>1015.26</td>
<td>0.0510</td>
<td>27.83</td>
</tr>
<tr>
<td>GPR</td>
<td>227.82</td>
<td>0.0693</td>
<td>25.29</td>
</tr>
<tr>
<td>BDB</td>
<td>189.08</td>
<td>0.0904</td>
<td>22.80</td>
</tr>
<tr>
<td>DLU</td>
<td>0.10</td>
<td>0.0458</td>
<td>28.96</td>
</tr>
<tr>
<td>DRE</td>
<td>0.05</td>
<td>0.0458</td>
<td>28.96</td>
</tr>
<tr>
<td>FSR</td>
<td>0.69</td>
<td>0.0575</td>
<td>28.65</td>
</tr>
<tr>
<td>TSE</td>
<td>0.17</td>
<td>0.0397</td>
<td>30.18</td>
</tr>
<tr>
<td>UMK</td>
<td>0.01</td>
<td>0.0457</td>
<td>28.97</td>
</tr>
<tr>
<td>Ours</td>
<td>204.19</td>
<td>0.0376</td>
<td>30.57</td>
</tr>
</tbody>
</table>

3http://www.vision.uji.es/srtoolbox/
TSE have shown to obtain the best quantitative performance followed some way behind by SRI. However, the differences among these methods are relatively small, which motivates a thorough discussion over qualitative results to find out each method’s singularity.

According to the visual results presented in Figs. 6 and 7, each SR method tends to foster a particular kind of visual feature on the super-resolved output. Some methods, like TSE or SRI, are able to obtain sharper edges, while others, like DLU or UMK, seem more robust to noise by generating smoother super-resolved textures. In terms of visual perceived quality, the proposed approach achieves a remarkable performance. For instance, the boat detail in Fig. 6(h) is certainly the most similar to its HR counterpart in Fig. 6(a).
Even though the result provided by SRI [see Fig. 6(d)] seems to obtain a slightly better contrast on some parts of the image, the proposed approach is able to introduce more high-frequency information in the boat structure. In addition, it is possible to see that the proposed approach also introduces some shadow fine details, which are not present in the others methods’ results.

When considering a four-time ratio, the proposed approach shows even better capability to recover high-frequency information while preserving HR details to avoid undesirable visual artifacts in the super-resolved output. For instance, it is the case of the result provided by SRI in Fig. 7(d), which provides a remarkable sharpness on edges, and however, it generates a kind of ghosting effect and also alters several shapes in the image. Despite the fact that TSE [see Fig. 7(g)] is able to overcome some of these limitations, the proposed approach certainly provides a more competitive visual result. That is, the proposed approach generates a super-resolved image with sharper edges, and it is also able to reduce the aliasing effect present in the TSE result. Another illustrative difference can be found in the asphalt surface, where the proposed approach removes the noise appearing in other output results.

Regarding computational time, we can observe some important differences among the tested methods. In particular, three groups can be identified when super-resolving LR input images: 1) BCI, IBP, DLU, DRE, FSK, and UMK, with an average time consumption per image under a second; 2) GPP and TSE, with a time between 10 and 120 s; and 3) the proposed approach, SRI, LSE, GPR, and BDB, which require more than 120 s per image. Even though the proposed approach is not one of the most computationally efficient methods, it shows a computational cost comparable to that of SRI, which, on average, has shown to be among the best methods together with TSE and the proposed approach.

D. Advantages and Limitations of the Proposed Approach

When comparing the proposed approach performance with respect to the best ones obtained in the experiments, we can observe the high potential of the proposed deep generative network to super-resolve the remote sensing data. To date, the HY approach used by SRI and TSE has shown to be one of the most effective ways to learn useful LR/HR patch relationships under an unsupervised SR scheme. However, this straightforward approach of searching patches across scales is rather constrained to the quality of the spatial information appearing in the LR input image. That is, the super-resolved result often tends to suffer from ghosting artifacts and watering effects as the magnification factor increases (see Fig. 7).

Even though TSE deals with this issue by allowing patch geometric transformation on the searching patch criteria, i.e., patches can occur in a lower scale as they are or even transformed, this process does not actually introduce any new spatial information in the output result which eventually may limit the SR process, especially in the remote sensing field. Note that remotely sensed imagery is usually a highly complex kind of data, because they are usually fully focused Multiband shots with plenty of different spatial details within the same image. As a result, the generation of a consistent spatial
variability becomes a key factor to improve the unsupervised remote sensing SR process.

Precisely, this is the objective of the proposed approach. In particular, the presented deep generative network learns the relationships between the LR and HR domains throughout several convolutional and downsampling layers starting from the LR input image. However, this process is affected by random noise, which is also restricted by the cost function, that is (5), to guarantee a global reconstruction constraint over the LR input image. That is, the random noise generates new spatial variations as possible solutions to relieve the ill-posed nature of the SR problem, while the cost optimizer controls that only these variations consistent with respect to the input LR image are promoted through the network to generate the
final SR result. Fig. 8 shows the SR process conducted by the proposed network over the parking test image considering a four-time scaling factor. As we can see, the reconstructed super-resolved result is initially noise; however, the spatial structures are recovered from a coarser to finer level of details as the network iterates.

In a sense, the proposed approach is able to recover a richer variety of high-frequency patterns for a given LR image due to its generative nature. In other words, the proposed deep generative network provides a more flexible unsupervised SR scheme than the current HY techniques, because it is able to introduce some spatial variations that are impossible to retrieve from the LR input image. In fact, it is possible to better appreciate the proposed approach effectiveness when only considering the PSNR metric, which is the most widely used quality index in SR. Figs. 9 and 10 show the PSNR gain obtained by the three best methods, i.e., the proposed approach, TSE, and SRI, with respect to the BCI baseline. As we can appreciate, the proposed approach provides some remarkable PSNR improvements in two times; however,
the PSNR gain is consistently higher when considering a four-time ratio. Note that, with this scaling factor, the level of uncertainty significantly increases, and it is then when the generative process of the proposed approach becomes more effective by introducing a higher variety of spatial details.

Although the results obtained by the proposed approach are encouraging, there are two points which deserve to be mentioned when comparing the proposed approach performance to the one obtained by the most effective unsupervised SR methods: the performance on some metrics and the computational cost.

On the one hand, the proposed approach performances on some metrics, specifically $Q$-index, SSIM, and SAM, seem not to be superior to the corresponding TSE results. For instance,
Fig. 9. PSNR (dB) results when considering a two-time scaling factor.

Fig. 10. PSNR (dB) results when considering a four-time scaling factor.

Table VII shows that the TSE obtains the best SSIM result for the four-time road image (0.8290), whereas the proposed approach achieves the second best SSIM value (0.8247). However, the proposed approach provides the best PSNR result (25.69 dB), which is substantially higher than the TSE one (23.86 dB). In spite of the small SSIM differences, it is possible to see the proposed approach advantages when considering the quantitative results. That is, Fig. 10 certainly shows that TSE magnifies the aliasing effect in the first line of pedestrian crossing and also generates a kind of watering effect on surfaces, whereas the proposed approach is able to obtain a more natural as well as reliable result, even though some image materials seem less contrasted. For the proposed approach, we adopt a cost function based on the mse in the way many other DL-based SR methods do in the supervised scheme (see [35], [69], [70]). Logically, our model has a different nature because of its unsupervised scheme; however, it seems reasonable to make this consideration, because the PSNR index, which is based on the mse, is one of the most commonly used metric in SR. Somehow, this definition of the cost function may constrain the performance on some metrics, because the network optimizer works for minimizing the mse and other kinds of metric features are not taken into account in this optimization process, which eventually may lead to a super-resolved solution with an excellent PSNR performance but with some small divergences in other figures of merit.

On the other hand, the computational cost of the proposed approach may also become a limitation in some specific scenarios. According to the quantitative results shown in Table IV, the proposed approach takes over 300 and 150 s to process each input image considering a two-time and four-time ratios, respectively. Even though the proposed approach has not shown to be one of the most computationally efficient methods, three important considerations have to be done to this extent. First, the computational burden is not only a drawback of the proposed approach but also of any DL architecture, because this kind of technology usually provides a more powerful framework to cope with new challenges and tasks. Second, the implementation of our model has not been optimized to really exploit the GPU hardware resources in order to substantially reduce the resulting computational time. That is, we make use of standard functions but further efforts could be addressed to generate a more optimized version of the code. Third, we use a general configuration of 4000 iterations as a security margin to guarantee a good network convergence; however, this value could be reduced in order to significantly improve the proposed approach computational efficiency. Fig. 11 shows the evolution of the PSNR metric with respect to the number of iterations for harbor, circular-farmland, industry, and road test images with a four-time ratio. As it is possible to see, the network is able to achieve a PSNR result that is very close to the optimal value after 2000 iterations, and therefore, it would be possible to reduce the number of iterations in order to significantly decrease the proposed approach computational time. In Fig. 12, we also show the PSNR evolution over time to highlight the fact that the proposed approach is able to rapidly converge to the optimal PSNR value. It should be noted that we use a unique network settings in this paper, and therefore, 4000 iterations are used to guarantee a good general parameter convergence, that is, without adapting the network to each input image.
V. CONCLUSION AND FUTURE LINES

In this paper, we have presented a new convolutional generator model to super-resolve the LR remote sensing data from an unsupervised perspective. Specifically, the proposed approach is initially able to learn relationships between the LR and HR domains while generating consistent random spatial variations. Then, the data are symmetrically projected to the target resolution, guaranteeing a reconstruction constraint over the LR input image. Our experiments, conducted using several test images, two scaling factors, and 12 different SR methods available in the literature, reveal the competitive performance of the proposed approach when super-resolving remotely sensed images.

One of the main conclusions that arises from this paper is the potential of deep generative models to cope with the unsupervised SR problem because of their capabilities to introduce new spatial details not present in the input LR image. As opposed to the common (HY) SR trend, which only relies on the patch relationships learned across scales, the proposed approach extends this scheme by introducing some spatial variations that allow the network to retrieve new spatial patterns that are consistent with the input LR image.

According to the conducted experiments, the proposed approach obtains a competitive global performance over the considered remote sensing test images in terms of both quantitative and qualitative SR results. Regarding the NRMSE, PSNR, and ERGAS metrics, the SR framework proposed in this paper obtains, on average, the best performance. When considering Q-index, SSIM, and SAM, TSE tends to provide the best average result, but the proposed approach is still able to perform among the best methods, especially when considering a four-time scaling factor.

Although the proposed approach results are encouraging as a generative SR model in remote sensing, the method still has some limitations, which provide room for improvement by conducting additional research on supervised SR. Specifically, our future work will be aimed at the following directions: 1) extending the cost function to simultaneously take into account several image quality metrics and also to extend it with the aim of implementing a noise reduction scheme for a different kind of input data; 2) adapting the convolutional kernel size to each specific input image; and 3) reducing the model computational cost by designing new strategies to actively control the number of iterations depending on the input image.

ACKNOWLEDGMENT

The authors would like to thank the editors and reviewers for their outstanding comments and suggestions, which greatly helped us to improve the technical quality and presentation of this paper.

REFERENCES


[92] Mercedes E. Pauletí (S’17) received the B.Sc. and M.Sc. degrees in computer engineering from the University of Extremadura, Cáceres, Spain, in 2014 and 2016, respectively, where she is currently pursuing the Ph.D. degree with a University Teacher Training Program from the Spanish Ministry of Education. She is also a member of the Hyperspectral Computing Laboratory, Department of Technology of Computers and Communications, University of Extremadura. Her research interests include remote sensing and analysis of very high spectral resolution with the current focus on deep learning and high performance computing.

[93] Ruben Fernandez-Beltran received the B.Sc. and M.Sc. degrees in computer science, the M.Sc. degree in intelligent systems, and the Ph.D. degree in computer science from Universitat Jaume I, Castellón de la Plana, Spain, in 2007, 2011, and 2016, respectively. He has been a Visiting Scientist with the University of Bristol, Bristol, U.K. Since 2017, he has been a Post-Doctoral Researcher with the Hyperspectral Computing Laboratory, University of Extremadura, Cáceres, Spain. He is currently a Post-Doctoral Researcher with the Computer Vision Group, University Jaume I, as a member of the Institute of New Imaging Technologies. His research interests include multimedia retrieval, spatio-spectral image analysis, pattern recognition techniques applied to image processing, and remote sensing.

Dr. Fernandez-Beltran is a member of the Spanish Association for Pattern Recognition and Image Analysis, which is part of the International Association for Pattern Recognition. He received the Outstanding Ph.D. Dissertation Award at Universitat Jaume I in 2017.

[94] Javier Plaza (M’09–SM’15) received the B.Sc. and Ph.D. degrees from the Hyperspectral Computing Laboratory, Department of Technology of Computers and Communications, University of Extremadura, Cáceres, Spain, in 2004 and 2008, respectively. He is currently a member of the Hyperspectral Computing Laboratory, Department of Technology of Computers and Communications, University of Extremadura. He has authored more than 150 publications, including over 50 Journal Citations Reports journal papers, ten book chapters, and 90 peer-reviewed conference proceeding papers. His research interests include hyperspectral data processing and parallel computing of remote sensing data.

Dr. Plaza was a recipient of the Outstanding Ph.D. Dissertation Award at the University of Extremadura in 2008 and the Best Column Award of the IEEE Signal Processing Magazine in 2015 and the most highly cited paper (2005–2010) in the *Journal of Parallel and Distributed Computing*. He received the best paper awards at the IEEE International Conference on Space Technology and the IEEE Symposium on Signal Processing and Information Technology. He has guest edited four special issues on hyperspectral remote sensing for different journals. He is an Associate Editor of the IEEE *Geoscience and Remote Sensing Letters* and the IEEE Remote Sensing Code Library. http://www.umbc.edu/rssipl/people/plaza.

Juan Mario Haut (S’17) received the B.Sc. and M.Sc. degrees in computer engineering from the University of Extremadura, Cáceres, Spain, in 2011 and 2014, respectively, where he is currently pursuing the Ph.D. degree with a University Teacher Training Program from the Spanish Ministry of Education. He is also a member of the Hyperspectral Computing Laboratory, Department of Computers and Communications, University of Extremadura. His research interests include remote sensing and analysis of very high spectral resolution with the current focus on deep learning and cloud computing. https://mhaust.github.io/
Antonio Plaza (M’05–SM’07–F’15) received the M.Sc. and Ph.D. degrees in computer engineering from the Hyperspectral Computing Laboratory, Department of Technology of Computers and Communications, University of Extremadura, Cáceres, Spain, in 1999 and 2002, respectively.

He is currently the Head of the Hyperspectral Computing Laboratory, Department of Technology of Computers and Communications, University of Extremadura. He has authored more than 600 publications, including over 200 Journal Citations Reports journal papers (over 160 in IEEE journals), 23 book chapters, and around 300 peer-reviewed conference proceeding papers. His research interests include hyperspectral data processing and parallel computing of remote sensing data.

Dr. Plaza is a fellow of the IEEE for his contributions to hyperspectral data processing and parallel computing of Earth observation data. He was also a member of the Steering Committee of the IEEE JOURNAL OF SELECTED TOPICS IN APPLIED EARTH OBSERVATIONS AND REMOTE SENSING (JSTARS). He was a recipient of the Best Column Award of the IEEE Signal Processing Magazine in 2015 and the 2013 Best Paper Award of the JSTARS journal and the most highly cited paper (2005–2010) in the Journal of Parallel and Distributed Computing. He received the best paper awards at the IEEE International Conference on Space Technology and the IEEE Symposium on Signal Processing and Information Technology. He served as the Director of Education Activities for the IEEE Geoscience and Remote Sensing Society (GRSS) from 2011 to 2012 and the President of the Spanish Chapter of the IEEE GRSS from 2012 to 2016. He was a recipient of the recognition of Best Reviewers of the IEEE GEOSCIENCE AND REMOTE SENSING LETTERS in 2009 and the recognition of Best Reviewers of the IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING in 2010, for which he served as an Associate Editor from 2007 to 2012. He is also an Associate Editor of the IEEE ACCESS (receiving a recognition as an outstanding Associate Editor of the journal in 2017) and was a member of the Editorial Board of the IEEE GEOSCIENCE AND REMOTE SENSING NEWSLETTER from 2011 to 2012 and the IEEE Geoscience and Remote Sensing Magazine in 2013. He has reviewed more than 500 manuscripts for over 50 different journals. He has guest edited ten special issues on hyperspectral remote sensing for different journals. He served as an Editor-in-Chief of the IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING from 2013 to 2017. http://www.umbc.edu/rssipl/people/aplaza.

Filiberto Pla received the B.Sc. and Ph.D. degrees in physics from the Universitat de València, Valencia, Spain, in 1989 and 1993, respectively.

He has been a Visiting Scientist with the Silsoe Research Institute, Bedford, U.K., the University of Surrey, Guildford, U.K., the University of Bristol, Bristol, U.K., the Institut national de recherche en sciences et technologies pour l’environnement et l’agriculture, Montpellier, France, the University of Genoa, Genoa, Italy, the Instituto Superior Técnico of Lisbon, Lisbon, Portugal, the Swiss Federal Institute of Technology–ETH Zürich, Zürich, Switzerland, the Idiap Research Institute, Martigny, Switzerland, the Technical University of Delft, Delft, The Netherlands, and Mid Sweden University, Sundsvall, Sweden. He has also been the Director of the Institute of New Imaging Technologies, University Jaume I, Castellón de la Plana, Spain, where he is currently a Full Professor with the Departament de Llenguatges i Sistemes Informàtics. His research interests include color and spectral image analysis, visual motion analysis, 3-D image visualization, and pattern recognition techniques applied to image processing.

Dr. Pla is a member of the Spanish Association for Pattern Recognition and Image Analysis, which is a member of the International Association for Pattern Recognition.