A New Deep Generative Network for Unsupervised Remote Sensing Single-Image Super-Resolution

Juan M. Haut, *Student Member, IEEE,* Ruben Fernandez-Beltran, Mercedes E. Paoletti, *Student Member, IEEE,*
Javier Plaza, *Senior Member, IEEE,* Antonio Plaza, *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 work 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 throughout several convolutional, down-sampling, 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 twelve 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*—Remote sensing, super-resolution, convolutional neural networks.**

This paper has been supported by Ministerio de Educación (Resolución de 26 de diciembre de 2014 y de 19 de noviembre de 2015, de la Secretaría de Estado de Educación, Formación Profesional y Universidades, por la que se convocan ayudas para la formación de profesorado universitario, de los subprogramas de Formación y de Movilidad incluidos en el Programa Estatal de Promoción del Talento y su Empleabilidad, en el marco del Plan Estatal de Investigación Científica y Técnica y de Innovación 2013-2016. This work has also been supported by Junta de Extremadura (decreto 297/2014, ayudas para la realización de actividades de investigación y desarrollo tecnológico, de divulgación y de transferencia de conocimiento por los Grupos de Investigación de Extremadura, Ref. GR15005). This work has been additionally supported by the Generalitat Valenciana through the contract APOSTD/2017/007 and by the Spanish Ministry of Economy under the project ESP2016-79503-C2-2-P. (Corresponding author: J.M. Haut.)

J. M. Haut, M. E. Paoletti, J. Plaza and A. Plaza are with the Hyperspectral Computing Laboratory, Department of Technology of Computers and Communications, Escuela Politécnica, University of Extremadura, PC-10003 Cáceres, Spain.(e-mail: juanmariohaut@unex.es; mpaoletti@unex.es; jplaza@unex.es; aplaza@unex.es).

R. Fernandez-Beltran and F.Pla are with the Institute of New Imaging Technologies, University Jaume I, 12071 Castellón, Spain. (e-mail: rufernan@uji.es; pla@uji.es).

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] or 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, are 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]. Whereas single-image SR techniques use a single image of the target scene to obtain the super-resolved output, 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 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 dataset. Traditionally used in the pattern recognition fied [19], ANNs have been also 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 two-dimensional 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 over-fit quickly due to the models' complexity and the lack of training data. Note that obtaining relevant remote sensing training data is expensive and time consuming. Besides, the amount of available training remote sensing datasets 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 (AEs) [39], [40], convolutional deep belief networks (DBNs) [41], convolutional generative adversarial neural networks (GANs) [42], convolutional recurrent neural networks (CRNN) [43] or fully convolutional networks (FCN) [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 low resolution (composed by 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 work 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 promoted to a higher scale throughout the network according to a global reconstruction constraint. Even though the relevance of generating new spatial variations when super-resolving remotely sensed data in a unsupervised manner, this is, to the best of our knowledge, the first time an unsupervised generative network model has been successfully formulated to super-resolve remote sensing imagery. Specifically, a convolutional generator network has been adopted, where from a given image $X^{LR} \in \mathbb{R}^{C\times W\times H}$, a higher resolution version $X^{HR} \in \mathbb{R}^{C\times t\cdot W\times t\cdot H}$ is generated (being $W < t \cdot W$ and $H < t \cdot H$, with t being a factor of resolution).

In addition, the algorithm has been adapted to be efficiently executed in parallel on graphics processing units (GPUs)[1] and presents some methodological improvements to make the model more efficient and effective. To summarize, the main contributions of this work can be highlighted as follows:

- An *hourglass* convolutional neural network model is developed to perform unsupervised super-resolution.
- In particular, a convolutional generator model has been implemented to super-resolve low-resolution remote sensing images.
- Starting from generative random noise, the model is able to reconstruct the image, promoting it to a higher scale according to a global reconstruction constraint.
- Experiments over three datasets, with 2 scaling factors and 12 different SR methods, reveal the competitive performance of the proposed model when super-resolving remotely sensed images.

The remainder of the paper is organized as follows. Section II presents an overview of single-image SR methods and their limitations. Section III describes the methodology employed 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 the 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

[1]The use of high performance computing methods (HPC), including parallelization with accelerators such as field programmable gate arrays (FPGAs) and GPUs [47]–[49], or the distribution with clusters and clouds [50], [51], have demonstrated great utility for the classification of remote images [52].

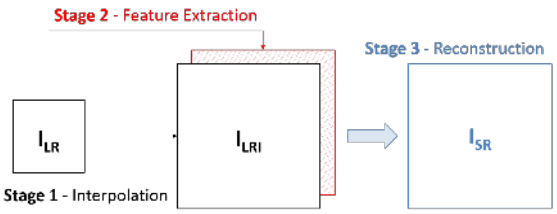


Fig. 1. Super-resolution based on image reconstruction (RE).

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 (Fig. 1): Stage 1, where the LR input image (I_{LR}) is upscaled to the target resolution (I_{LRI}) 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_{LRI}) and the extracted LR features to obtain the final reconstructed result I_{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 low-resolution version of the super-resolved result. GP takes advantage of the fact that the shape of the gradient profiles tends to remain invariant across scales, therefore LR gradient can be used to reconstruct the output image sharpness. PSF deconvolution methods tackle the upscaling 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 Fig. 2 shows, 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 non-linear 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.

Lastly, HY techniques work towards reaching an agreement between RE and LE approaches. In particular, they perform a training process but only using the LR input image. The rationale behind 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,

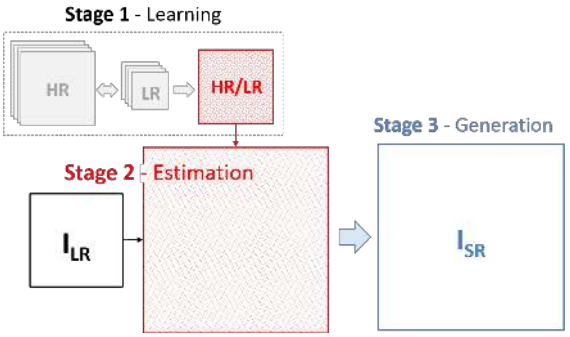


Fig. 2. Super-resolution based on image learning (LE).

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 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 sat-

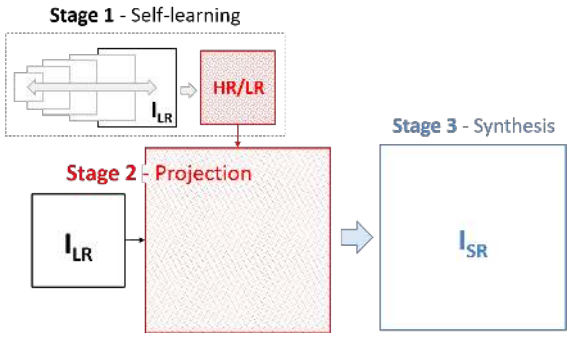


Fig. 3. Super-resolution based on hybrid algorithms (HY).

isfying 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 [69]–[71], 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 not 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 work.

In [72], authors propose a SR approach using a back-propagation neural network as a regression function, and basing on (i) spectral unmixing, (ii) super-resolution mapping and (iii) 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 [73]. In contrast, a hybrid (also called self-learning) SR scheme has been proposed in this work 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 down-sampling 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 down-sampled 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 deep learning-based approaches, e.g. [69]–[71], where the original LR image is used to learn the down-sampling 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 θ) 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 random noise and $X \in \mathbb{R}^{3\times W'\times H'}$ is the desired RGB high resolution image.

Given a 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 a HR version $X^{HR} \in \mathbb{R}^{3\times t\cdot W\times t\cdot H}$ 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 are composed

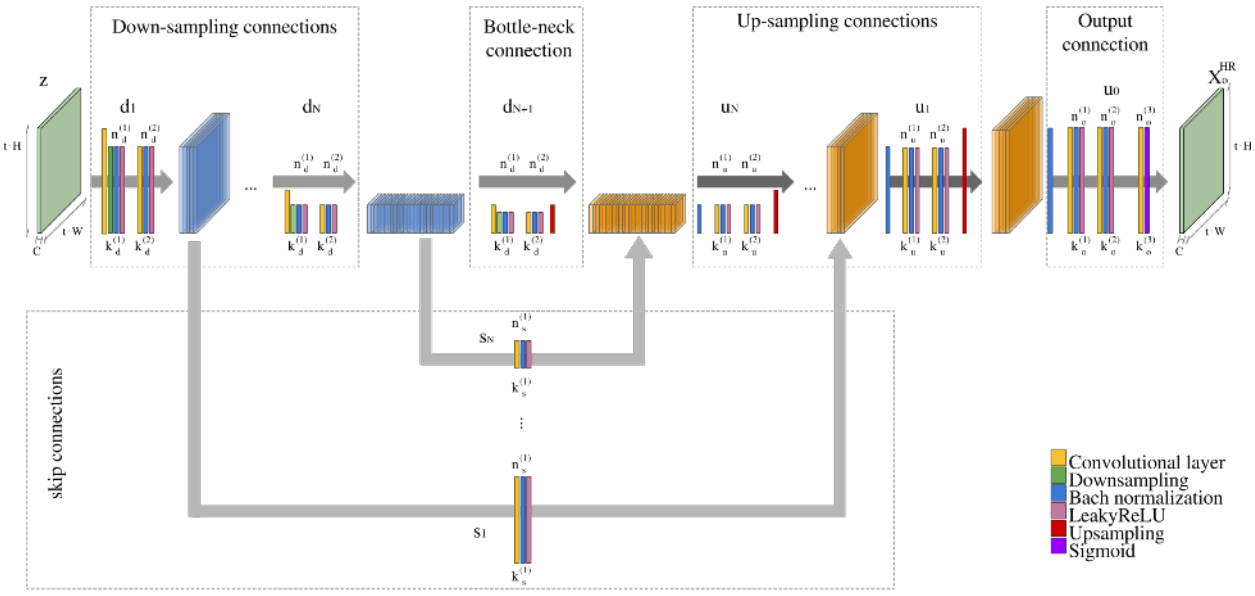


Fig. 4. The proposed 2D-CNN architecture model follows a symmetric topology. The input image z goes through a first step of down-sampling composed by blocks $(d^{(1)}, d^{(2)}, ..., d^{(N)})$ of several CONV, down-sampling, BATCH-NORM and activation layers, where each $n_d^{(j)}$ and $k_d^{(j)}$ (with $j = 1, 2$) are the number of filters and kernel sizes of each down-sampling connection $d^{(i)}$. Then, symmetrically, data goes through the up-sampling step, where the output of each block $u^{(i)}$ (with number of filters $n_u^{(j)}$ and kernel size $k_u^{(j)}$, $j = 1, 2$ and composed by CONV, BATCH-NROM, up-sampling and activation layers) is combined with the features of the corresponding d_i through a skip connection s_i, which also has a number of filters $n_s^{(1)}$ and a kernel size $k_s^{(1)}$ and is composed by a CONV, a BATCH-NORM and an activation layers.

by layers that are applied over defined regions of the input data, i.e. they are local-connected to the input, transforming the input volume to an output volume of neuron activations which will serve as 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, 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 $x^{(i)} \in \mathbb{R}^C = [x_1^{(i)}, x_2^{(i)}, ..., x_C^{(i)}]$ is a pixel with C spectral bands of image $X \in \mathbb{R}^{C \times W \times H}$, with $i = 1, 2, ..., W \cdot H$, 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 the number of spectral bands of the patch (with $b \leq C$). 1D-CNN models take separately as input data each pixels vector $x^{(i)}$, extracting only spectral information [74]. On the other hand, 2D-CNNs extract spatial information, taking as input data the entire image X [75] or image patches $P^{(j)}$ [76], where C and b are set to small values, i.e. the spectral information is not very relevant compared to the spatial information. Finally, 3D-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 2D-CNN approach is taken while 1D- and 3D-CNNs are usually for multi- and hyperspectral images. This paper works with RGB remote sensing datasets, so a 2D-CNN architecture has been implemented to take

advantage of the spatial information contained in the images. It is composed by five different kinds of layers, described below:

- *Convolution layer* (CONV): this kind of layer is composed by a block of neurons where each slice (also called filter or 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 following equation 1 as the dot product between the $n^{(i)}$ slices' weights $W^{(i)}$ and biases $B^{(i)}$ (being $n^{(i)}$ 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 $C^{(i)}$:

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

  being $o_{f,l}^{(i-1)}$ the feature (f,l) of the feature map $O^{(i-1)} \in \mathbb{R}^{W,H}$, with $f = 1,2,...,W$ and $l = 1,2,...,H$, and $w_{m,n}^{(i)}$ the weight (m,n) of weight matrix $W^{(i)} \in \mathbb{R}^{k^{(i)},k^{(i)}}$.
  As 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:

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

  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 $O^{(i-1)} \in \mathbb{R}^{n^{(i-1)},W,H}$.
- *Batch normalization layer* (BATCH-NORM): normally it is placed behind the convolution layer and it applies the normalization defined by equation 2 over the batch data:

$$y = \frac{x - \text{mean}[x]}{\sqrt{\text{Var}[x] + \epsilon}} \cdot \gamma + \beta \quad (2)$$

  where γ and β are learnable parameter vectors, and ϵ is a parameter for numerical stability.
- *Activation layer*: after CONV and BATCH-NORM layers, the activation layer or non-linearity layer embeds a non-linear function that is applied over the output of previous layer, as the rectified linear unit (ReLU) [77], [78]. In this case, the LeakyReLU function is implemented [79]:

$$f(x) = \begin{cases} x & \text{if } x > 0 \\ \alpha x & \text{if } x \leq 0 \end{cases} \quad (3)$$

  where α is a small non-zero parameter, normally 0.001.
- *Down-sampling/Up-sampling layer*: the proposed model also implements down-sampling and up-sampling 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 down-sampling, however the proposed model down-samples the input data setting the strides of certain CONV layers to $S = 2$. Additionally, the up-sampling 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 $\mathbb{R}^{C \times t \cdot W \times t \cdot H}$ 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], [80], z goes through two main steps composed by several blocks:

1) The down-sampling step is composed by N blocks of layers, called $d^{(i)}$ ($i = 1, 2, ...N$), where the input of each one is the feature maps of the previous one. Each $d^{(i)}$ is composed by an initial CONV layer $C_d^{(1)}$ that performs the down-sampling 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 $C_d^{(2)}$ without down-sampling (i.e. $S = 1$) and also followed by a BATCH-NORM layer and the LeakyReLU activation function. $C_d^{(1)}$ and $C_d^{(2)}$ have their own number of filters ($n_d^{(1)}$ and $n_d^{(2)}$) and their own kernel size ($k_d^{(1)}$ and $k_d^{(2)}$).
   In fact, each block $d^{(i)}$ is reducing the space information, i.e. generating a low spatial resolution data that will feed the second up-sampling step.
2) The up-sampling step is symmetric to down-sampling one and it is also composed by N blocks of layers, called $u^{(i)}$ ($i = N, ..., 2, 1$), where the input of each one is the output of the previous one. In this case, each $u^{(i)}$ is composed by several stacked layers. The first one is a BATCH-NORM layer, followed by the first CONV layer $C_u^{(1)}$ (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 $C_u^{(2)}$ (which also maintains the size of the data). After the BATCH-NORM and the activation function, the output will finally feed the bilinear up-sampling layer with factor equal to 2. Again, $C_u^{(1)}$ and $C_u^{(2)}$ have their own number of filters ($n_u^{(1)}$ and $n_u^{(2)}$) and their own kernel size ($k_u^{(1)}$ and $k_u^{(2)}$).

Both steps, down-sampling and up-sampling, are symmetrical and connected by skip connections, i.e. the input of each up-sampling block $u^{(i)}$ is combined with the corresponding $d^{(i)}$ through the skip connection $s^{(i)}$ ($i = 1, 2..., N$) composed by a CONV layer $C_s^{(1)}$, with its number of filters $n_s^{(i)}$ and its kernel size $k_s^{(i)}$, a BATCH-NORM layer and the activation function, LeakyReLU. In fact, the output of $s^{(i)}$ is concatenated to the input of $u^{(i)}$. The chosen topology is depicted in Fig. 4. At the end of the topology, an output block is added, composed with a CONV layer and a sigmoid function at the end. As result, a HR image $X_o^{HR} \in \mathbb{R}^{3\times t\cdot W\times t\cdot H}$ is generated as output of the network.

In particular, the SR's goal is to generate a HR image from a LR one, minimizing the following cost function:

$$\min \parallel \phi(X^{HR}) - X^{LR} \parallel^2 \tag{4}$$

In fact, our remote sensing datasets are composed by HR images. However, we cannot use them because they cannot be considered as ground-truth to perform SR. In order to solve this, a LR version is generated from each HR image by a down-sampler $\phi : \mathbb{R}^{3\times t\cdot W\times t\cdot H} \rightarrow \mathbb{R}^{3\times W\times H}$, so $X^{LR} = \phi(X^{HR})$. In our case the down-sampler ϕ has been implemented using Lanczos3 resampling [81], 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 a HR image, X_o^{HR}. In order to solve this, the down-sampler function ϕ is applied over X_o^{HR}. At the end, equation 4 can be rewritten as:

$$\min \parallel \phi(X^{HR}) - \phi(X_o^{HR}) \parallel^2 \rightarrow \min \parallel X^{LR} - X_o^{LR} \parallel^2 \tag{5}$$

The cost function defined by equation 5 is optimized iteratively by the model via Adam optimizer [82]. The

proposed method is summarized in Algorithm 1. Also, in Fig. 8 we can observe the X_o^{HR} image generated by the model at each epoch.

Algorithm 1 Unsupervised remote sensing single-image super-resolution algorithm

procedure SR_MODEL(X^{LR}, t) ▷ $X^{LR} \in \mathbb{R}^{C\times W\times H}$ original low resolution remote sensing image, t resolution factor
  $z \leftarrow$ Random noise with size $C \times t \cdot W \times t \cdot H$
  repeat
    $X_o^{HR} \leftarrow$model_net(z)
    $X_o^{LR} \leftarrow \phi\left(X_o^{HR}\right)$ ▷ ϕ is Lanczos3
    $loss = \text{MSE}(X^{LR}, X_o^{LR})$
    ADAM_Optimizer($loss$)
    $z \leftarrow X_o^{HR}$
  until Reach maximum epoch
  return X_o^{HR}
end procedure

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

TABLE I
NETWORK TOPOLOGY FOR 2X SUPER-RESOLUTION. THE UP-SAMPLING PHASE HAS BEEN PERFORMED WITH A SCALE-FACTOR SET TO 2.

Block ID	CONV ID	Kernel size $k_d^{(j)}/k_u^{(j)}/k_s^{(j)}$	Number of kernels $n_d^{(j)}/n_u^{(j)}/n_s^{(j)}$	Stride
Down-sampling connections				
$d^{(1)}$	$C_d^{(1)}$	3×3	256	2
	$C_d^{(2)}$	3×3	256	1
$d^{(2)}$	$C_d^{(1)}$	3×3	256	2
	$C_d^{(2)}$	3×3	256	1
Bottle-neck connection				
$d^{(3)}$	$C_d^{(1)}$	3×3	256	2
	$C_d^{(2)}$	3×3	256	1
Up-sampling connections				
$u^{(2)}$	$C_u^{(1)}$	5×5	256	1
	$C_u^{(2)}$	1×1	256	1
$u^{(1)}$	$C_u^{(1)}$	5×5	256	1
	$C_u^{(2)}$	1×1	256	1
Output connections				
$u^{(0)}$	$C_u^{(1)}$	5×5	256	1
	$C_u^{(2)}$	1×1	256	1
	$C_u^{(3)}$	1×1	3	1
Skip connections				
$s^{(1)}$	$C_s^{(1)}$	1×1	3	1
$s^{(2)}$	$C_s^{(1)}$	1×1	3	1

A. Metrics

In order to compare the properties of the obtained X_o^{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 $X_o^{HR} = X_o$ and $X^{HR} = X$, being $x_o^{(i)}$ and $x^{(i)}$ the i-th pixels of X_o and X respectively.

Following equation 6, where $n_{samples}$ is the number of pixels of X and X_{max} and X_{min} are the maximum and minimum 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.

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

Peak signal-to-noise ratio (PSNR) [83] represents a better image quality than NRMSE. This metric is defined as the standard index for SR, being MAX_f the maximum signal value that exists in the original X image. A higher PSNR value indicates that the reconstructed image X_o

TABLE II
NETWORK TOPOLOGY FOR 4X SUPER-RESOLUTION. THE UP-SAMPLING PHASE HAS BEEN PERFORMED WITH A SCALE-FACTOR SET TO 2.

Block ID	CONV ID	Kernel size $k_d^{(j)}/k_u^{(j)}/k_s^{(j)}$	Number of kernels $n_d^{(j)}/n_u^{(j)}/n_s^{(j)}$	Stride
Down-sampling connections				
$d^{(1)}$	$C_d^{(1)}$	3×3	256	2
	$C_d^{(2)}$	3×3	256	1
$d^{(2)}$	$C_d^{(1)}$	3×3	256	2
	$C_d^{(2)}$	3×3	256	1
$d^{(3)}$	$C_d^{(1)}$	3×3	256	2
	$C_d^{(2)}$	3×3	256	1
$d^{(4)}$	$C_d^{(1)}$	3×3	256	2
	$C_d^{(2)}$	3×3	256	1
$d^{(5)}$	$C_d^{(1)}$	3×3	256	2
	$C_d^{(2)}$	3×3	256	1
Bottle-neck connection				
$d^{(6)}$	$C_d^{(1)}$	3×3	256	2
	$C_d^{(2)}$	3×3	256	1
Up-sampling connections				
$u^{(5)}$	$C_u^{(1)}$	3×3	256	1
	$C_u^{(2)}$	1×1	256	1
$u^{(4)}$	$C_u^{(1)}$	3×3	256	1
	$C_u^{(2)}$	1×1	256	1
$u^{(3)}$	$C_u^{(1)}$	3×3	256	1
	$C_u^{(2)}$	1×1	256	1
$u^{(2)}$	$C_u^{(1)}$	3×3	256	1
	$C_u^{(2)}$	1×1	256	1
$u^{(1)}$	$C_u^{(1)}$	3×3	256	1
	$C_u^{(2)}$	1×1	256	1
Output connections				
$u^{(0)}$	$C_u^{(1)}$	3×3	256	1
	$C_u^{(2)}$	1×1	256	1
	$C_u^{(3)}$	1×1	3	1
Skip connections				
$s^{(1)}$	$C_s^{(1)}$	1×1	3	1
$s^{(2)}$	$C_s^{(1)}$	1×1	3	1
$s^{(3)}$	$C_s^{(1)}$	1×1	3	1
$s^{(4)}$	$C_s^{(1)}$	1×1	3	1
$s^{(5)}$	$C_s^{(1)}$	1×1	3	1

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) [84] calculates the angle between the corresponding pixels of the super-resolved image X_o and original image X in the domain $[0, \pi]$.

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

The *universal image quality index*, also called Q-index, gathers three different properties in the image evaluation: (a) correlation, (b) luminance and (c) contrast.

$$Q(X, X_o) = \sum_{j}^{n_{bands}} \left(\overbrace{\frac{\sigma_{IR}}{\sigma_X \ \sigma_{X_o}}}^{a} \ \overbrace{\frac{2 \ \overline{X} \ \overline{X_o}}{(\overline{X})^2 \ (\overline{X_o})^2}}^{b} \ \overbrace{\frac{2 \ \sigma_X \ \sigma_{X_o}}{(\sigma_X)^2 \ (\sigma_{X_o})^2}}^{c} \right)_j \tag{9}$$

An extension of Q-index is the *structural similarity* (SSIM) [85], 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).

$$\text{SSIM}(X, X_o) = \frac{(2\mu_X \mu_{X_o} + c_1) * (2\sigma_{X X_o} + c_2)}{\left(\mu_X^2 + \mu_{X_o}^2 + c_1\right) * \left(\sigma_X^2 + \sigma_{X_o}^2 + c_2\right)} \tag{10}$$

Erreur relative globale adimensionnelle de synthese (ERGAS) [86] measures the quality of obtained X_o taking into account the scaling factor to evaluate the super-resolved image.

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

IV. EXPERIMENTS

A. Experimental Configuration and Datasets

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

- A GPU environment composed by a 6th Generation Intel® Core™i7-6700K processor with 8M of

Cache and up to 4.20GHz (4 cores/8 way multi-task processing), 40GB of DDR4 RAM with a serial speed of 2400MHz, a GPU NVIDIA GeForce GTX 1080 with 8GB GDDR5X of video memory and 10 Gbps of memory frequency, a Toshiba DT01ACA HDD with 7200RPM and 2TB of capacity, and an ASUS Z170 pro-gaming motherboard. The software environment is composed by Ubuntu 16.04.4 x64 as operating system, Pytorch [87] 0.3.0 and compute device unified architecture (CUDA) 8 for GPU functionality.

- A CPU enviroment composed by Intel Core i7-4790 @ 3.60GHz, 16GB of DDR3 RAM with a serial speed of 800MHz, a Western Digital HDD with 7200RPM and 1TB of capacity. The software environment is composed by Windows 7 as 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.

Additionally, the employed database is composed by 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 Eq. 4, presented in other approaches such as [69]–[71], [88]. The employed repositories are described below, and are publicly available on this repository[2].

[2]https://github.com/mhaut/images-superresolution

1) UCMERCED [89]: It is composed by 21 land use classes, including agricultural, airplane, baseball diamond, beach, buildings, chaparral, dense residential, forest, freeway, golf course, harbor, intersection, mediumdensity residential, mobile home park, overpass, 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 [90]: this data set contains 2800 images with seven different classes. The dataset 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 [91]: the remote sensing image scene classification (RESISC) dataset has been created by Northwestern Polytechnical University (NWPU). This dataset has 45 scenes with a total number of 31500 images, 700 per class. The size of each image is 256×256 pixels.

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

From these images, a LR version has been generated from their corresponding HR counterparts following a

two-step procedure [92]: (i) an initial blurring step and (ii) a final decimation process. In particular, a Lanczos3 windowed sinc filter has been used for blurring the corresponding HR images, then these images have been down-sampled 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, $w = (4 * s + 1)$, where w represents the filter width and s is the considered scaling factor. For the down-sampling 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 pre-processing 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.

The performance of the proposed approach has been compared to the results obtained by 11 different unsupervised SR methods available in the literature, as well as the bi-cubic interpolation kernel function [81] used as a up-scaling baseline. These SR methods have been considered for the experimental discussion because of they provide an unsupervised SR scheme in the same way the proposed approach does, using the LR input image to generate a super-resolved output result. Additionally, two different scaling factors, $2\times$ and $4\times$, have been tested over the considered image dataset (Sec. A). Table III provides a brief description of the SR techniques considered in the experimental part of the work.

All the tested methods have been downloaded from the following website[3] 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.

[3]http://www.vision.uji.es/srtoolbox/

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

Identifier	SR type	Method description	Reference
BCI	Baseline	Bi-cubic interpolation kernel	[81]
IBP	Reconstruction	Iterative back projection	[55]
GPP	Reconstruction	Gradient profile prior	[56]
SRI	Hybrid	Scale patch redundancy	[64]
LSE	Hybrid	Scale patch redundancy	[93]
GPR	Reconstruction	Gaussian Process Regression	[94]
BDB	Hybrid	Scale patch redundancy	[66]
DLU	Reconstruction	Point Spread Funtion deconvolution	[57]
DRE	Reconstruction	Point Spread Funtion deconvolution	[58]
FSR	Reconstruction	Point Spread Funtion deconvolution	[95]
TSE	Hybrid	Transformed scale patch redundancy	[65]
UMK	Reconstruction	Unsharp masking	[59]
Ours	Generative-HY	The proposed approach	-

B. Results

Tables V-VII present the quantitative assessment of the considered SR methods in terms of seven different quality metrics. Specifically, each table contains the super-resolved results of four test images and, for each image, the SR results are provided in rows considering two different scaling factors, $2\times$ and $4\times$, which are shown in columns. Besides, Table IV provides the average results for the whole image collection in order to provide a global view.

In addition to the quantitative evaluation provided by the considered metrics, some visual results are provided as a qualitative evaluation for the tested SR methods. Specifically, Figs. 6-7 show the super-resolved results obtained for harbor and road test images considering $2\times$ and $4\times$ scaling factors, respectively. Besides, Fig. 8

TABLE IV

AVERAGE SR RESULTS. THE BEST RESULT FOR SCALING RATIO AND METRIC IS HIGHLIGHTED IN BOLD FONT.

Image	Method	Ratio 2x							Ratio 4x						
		TIME	NRMSE	PSNR	ERGAS	Qindex	SSIM	SAM	TIME	NRMSE	PSNR	ERGAS	Qindex	SSIM	SAM
Average	**BCI**	**0.01**	0.0506	28.11	5.975	0.7915	0.8406	0.0160	**0.01**	0.0837	23.59	4.913	0.4769	0.6067	**0.0233**
	IBP	0.15	0.0455	29.01	5.353	0.8200	0.8667	0.0174	0.48	0.0793	24.05	4.668	0.5474	0.6575	0.0260
	GPP	25.30	0.0501	28.20	5.934	0.7870	0.8409	0.0178	17.46	0.0823	23.74	4.830	0.4847	0.6155	0.0244
	SRI	337.77	0.0395	30.23	4.599	0.8337	0.8805	0.0167	335.30	0.0823	23.62	4.830	0.5490	0.6631	0.0272
	LSE	1015.26	0.0510	27.83	5.874	0.7995	0.8546	0.0181	345.41	0.0865	23.15	4.925	0.5008	0.6454	0.0293
	GPR	227.82	0.0693	25.29	8.179	0.6330	0.7215	0.0194	100.26	0.0888	23.03	5.202	0.4288	0.5734	0.0250
	BDB	189.08	0.0904	22.80	10.660	0.6143	0.7093	0.0233	302.67	0.1341	19.26	7.873	0.2610	0.4569	0.0316
	DLU	0.10	0.0458	28.96	5.374	0.8171	0.8642	0.0175	0.10	0.0811	23.87	4.767	0.4958	0.6220	0.0246
	DRE	0.05	0.0458	28.96	5.374	0.8171	0.8642	0.0175	0.05	0.0811	23.87	4.767	0.4958	0.6220	0.0246
	FSR	0.69	0.0575	26.85	6.825	0.7462	0.8170	0.0184	1.81	0.1015	21.81	5.974	0.2965	0.5190	0.0265
	TSE	17.64	0.0397	30.18	4.626	**0.8527**	**0.8902**	**0.0150**	17.27	0.0742	24.73	4.386	**0.5695**	**0.6820**	0.0237
	UMK	0.01	0.0457	28.97	5.367	0.8176	0.8647	0.0176	0.01	0.0789	24.11	4.648	0.5318	0.6465	0.0253
	Ours	294.19	**0.0376**	**30.57**	**4.366**	0.8351	0.8836	0.0163	156.71	**0.0704**	**25.21**	**4.193**	0.5483	0.6776	0.0236

presents the visual evolution of the super-resolved result along the network iterations.

C. Discussion

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

When considering a $2\times$ scaling factor, the proposed approach (together with the hybrid 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 $2\times$ and 0.48 dB for $4\times$. Regarding the methods providing a moderate improvement (b), 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 (c), 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 $4\times$ 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 methods obtain a similar average result for the SAM metric. It should be noted that SRI performance has worsened when using a $4\times$ ratio, however it still obtains the third best Q-index and SSIM results. With respect to the rest of the moderate (b) and low performance methods (c), they

(*a*) HR (*b*) BCI (21.73 dB) (*c*) IBP (23.43 dB) (*d*) SRI (25.82 dB)

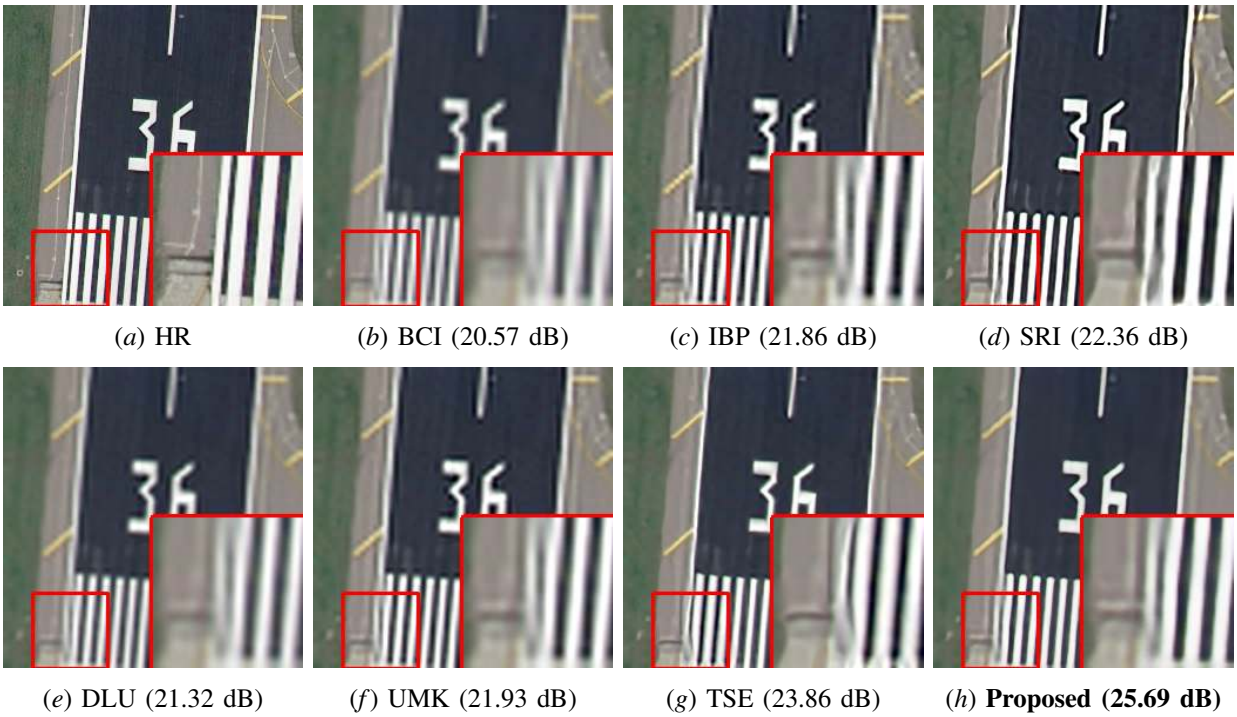
(*e*) DLU (23.40 dB) (*f*) UMK (23.44 dB) (*g*) TSE (25.63 dB) (*h*) **Proposed (26.84 dB)**

Fig. 6. SR results obtained using the methods shown in captions over the test image harbor with a $2\times$ scaling factor. For each result, PSNR (dB) values appear in brackets. The best PSNR value is highlighted in bold.

obtain similar results with regards to the ones obtained with a $2\times$ factor. Overall, the proposed approach and 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 singularities.

According to the visual results presented in Figs. 6-7, 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 (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 $4\times$ 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, however it generates a kind of ghosting effect and also alters several shapes in the image. Despite the fact that TSE (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



(*a*) HR (*b*) BCI (20.57 dB) (*c*) IBP (21.86 dB) (*d*) SRI (22.36 dB)

(*e*) DLU (21.32 dB) (*f*) UMK (21.93 dB) (*g*) TSE (23.86 dB) (*h*) **Proposed (25.69 dB)**

Fig. 7. SR results obtained using the methods shown in captions over the test image road with a $4\times$ scaling factor. For each result, PSNR (dB) values appear in brackets. The best PSNR value is highlighted in bold.

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: (i) BCI, IBP, DLU, DRE, FSK and UMK, with an average time consumption per image under a second, (ii) GPP and TSE, with a time between 10 and 120 seconds, and (iii) the proposed approach, SRI, LSE, GPR and BDB which require more than 120 seconds 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 remote sensing data. To date, the hybrid 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 (Fig. 7).

Even though TSE deals with this issue by allowing patch geometric transformation on the searching patch

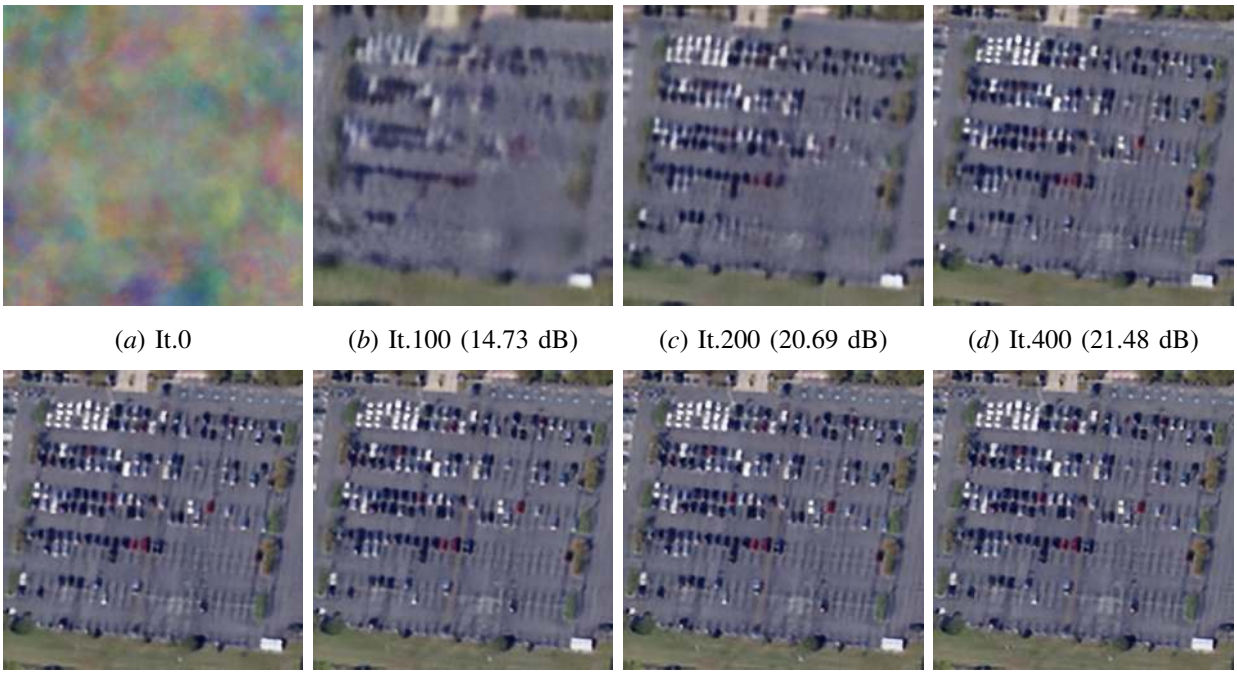
criteria, i.e. patches can occur in a lower scale as they are or even transformed, this process does not actually introduce any new spatial information in the output result which eventually may limit the SR process, especially in the remote sensing field. Note that remotely sensed imagery are usually a highly complex kind of data because they are usually fully-focused multi-band 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, i.e. equation (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 though the network to generate the final SR result. Fig. 8 depicts the SR process conducted by the proposed network over the parking test image considering a $4\times$ 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 hybrid 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. 9-10 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 $2\times$, however the PSNR gain is consistently higher when considering a $4\times$ 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 than the corresponding TSE results. For instance, Table VII shows that the TSE obtains the best SSIM result for the $4\times$ 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 qualitative results. That is, Fig. 10 certainly shows that TSE magnifies the aliasing effect in the fist 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



(*a*) It.0 (*b*) It.100 (14.73 dB) (*c*) It.200 (20.69 dB) (*d*) It.400 (21.48 dB)

(*e*) It.800 (21.89 dB) (*f*) It.1000 (21.97 dB) (*g*) It.2000 (22.13 dB) (*h*) **It.3900 (22.18 dB)**

Fig. 8. SR process conducted by the proposed approach over the parking test image with a $4\times$ scaling factor. Each sub-figure represents the obtained X_o^{HR} images at each epoch of the model, following Algorithm 1.

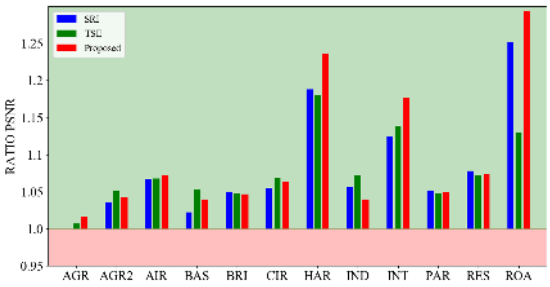


Fig. 9. PSNR (dB) results when considering a $2\times$ scaling factor.

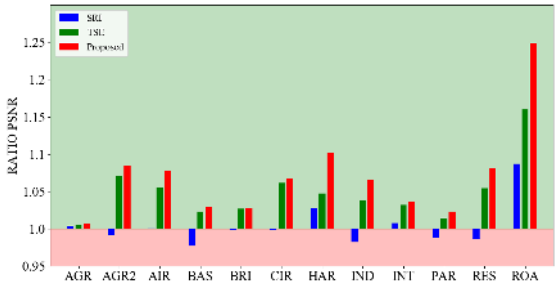


Fig. 10. PSNR (dB) results when considering a $4\times$ scaling factor.

though some image materials seem less contrasted. For the proposed approach, we adopt a cost function based on the mean-squared-error (MSE) in the way many other deep learning-based SR methods do in the supervised scheme, e.g. [69]–[71]. Logically, our model has a different nature because of its unsupervised scheme, however it seem reasonable to make this consideration because the PSNR index, which is based on the MSE, is one 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 led 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 seconds to process each input image considering a $2\times$ and $4\times$ ratios respectively. Even though the proposed approach has shown not 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 deep learning 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 much more optimized version of the code. Third, we use a general configuration of $4,000$ 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 iteration for harbor, circular-farmland, industry and road test images with a $4\times$ 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 2,000 iterations, 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 work, therefore $4,000$ iterations are used to guarantee a good general parameter convergence, that is, without adapting the network to each input image.

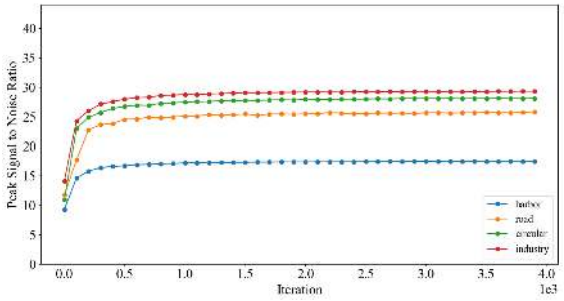


Fig. 11. PSNR evolution for harbor, circular-farmland, industry and road test images considering a $4\times$ scaling ratio versus iteration.

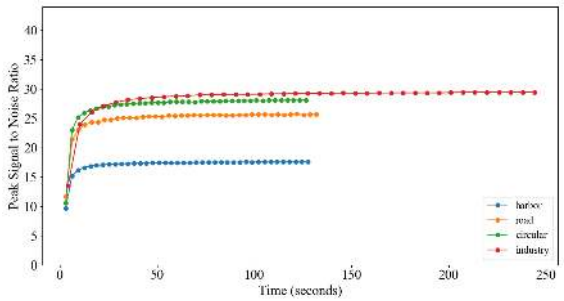


Fig. 12. PSNR evolution for harbor, circular-farmland, industry and road test images considering a $4\times$ scaling ratio versus time.

V. Conclusions and Future Lines

In this work, we have presented a new convolutional generator model to super-resolve 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 is symmetrically projected to the target resolution, guaranteeing a reconstruction constraint over the LR input image. Our experiments, conducted using several test images, 2 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 work 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 (hybrid) 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 work 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 $4\times$ 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 unsupervised SR. Specifically, our future work will be aimed at the following directions: (i) 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, (ii) adapting the convolutional kernel size to each specific input image, and (iii) 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 gratefully acknowledge the Editors and Reviewers for their outstanding comments and suggestions, which greatly helped us to improve the technical quality and presentation of this work.

REFERENCES

[1] J. A. Benediktsson, J. Chanussot, and W. M. Moon, "Very high-resolution remote sensing: Challenges and opportunities," *Proceedings of the IEEE*, vol. 100, no. 6, pp. 1907–1910, 2012.

[2] M. T. Pham, E. Aptoula, and S. Lefvre, "Feature profiles from attribute filtering for classification of remote sensing images," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 11, no. 1, pp. 249–256, 2018.

[3] A. E. Maxwell, T. A. Warner, and F. Fang, "Implementation of machine-learning classification in remote sensing: an applied review," *International Journal of Remote Sensing*, vol. 39, no. 9, pp. 2784–2817, 2018.

[4] G. Sumbul, R. G. Cinbis, and S. Aksoy, "Fine-grained object recognition and zero-shot learning in remote sensing imagery," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 56, no. 2, pp. 770–779, 2018.

[5] X. Kang, Y. Huang, S. Li, H. Lin, and J. A. Benediktsson, "Extended random walker for shadow detection in very high resolution remote sensing images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 56, no. 2, pp. 867–876, 2018.

[6] H. Lin, Z. Shi, and Z. Zou, "Fully convolutional network with task partitioning for inshore ship detection in optical remote sensing images," *IEEE Geoscience and Remote Sensing Letters*, vol. 14, no. 10, pp. 1665–1669, 2017.

[7] T. Wu, J. Luo, J. Fang, J. Ma, and X. Song, "Unsupervised object-based change detection via a weibull mixture model-based binarization for high-resolution remote sensing images," *IEEE Geoscience and Remote Sensing Letters*, vol. 15, no. 1, pp. 63–67, 2018.

[8] Z. Liu, G. Li, G. Mercier, Y. He, and Q. Pan, "Change detection in heterogenous remote sensing images via homogeneous pixel transformation," *IEEE Transactions on Image Processing*, vol. 27, no. 4, pp. 1822–1834, 2018.

[9] A. S. Belward and J. O. Skøien, "Who launched what, when and why; trends in global land-cover observation capacity from civilian earth observation satellites," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 103, pp. 115–128, 2015.

[10] T. S. Unger Holtz, "Introductory digital image processing: A remote sensing perspective," 2007.

[11] S. C. Park, M. K. Park, and M. G. Kang, "Super-resolution image reconstruction: a technical overview," *IEEE signal processing magazine*, vol. 20, no. 3, pp. 21–36, 2003.

[12] P. Milanfar, *Super-resolution imaging*. CRC press, 2010.

[13] L. Yue, H. Shen, J. Li, Q. Yuan, H. Zhang, and L. Zhang, "Image super-resolution: The techniques, applications, and future," *Signal Processing*, vol. 128, pp. 389–408, 2016.

[14] A. Garzelli, "A review of image fusion algorithms based on the super-resolution paradigm," *Remote Sensing*, vol. 8, no. 10, p. 797, 2016.

[15] C.-Y. Yang, C. Ma, and M.-H. Yang, "Single-image super-resolution: A benchmark," in *European Conference on Computer Vision*, 2014.

[16] A. Punnappurath, T. M. Nimisha, and A. N. Rajagopalan, "Multi-image blind super-resolution of 3d scenes," *IEEE Transactions on Image Processing*, vol. 26, no. 11, pp. 5337–5352, 2017.

[17] D. Yang, Z. Li, Y. Xia, and Z. Chen, "Remote sensing image super-resolution: Challenges and approaches," in *Digital Signal Processing (DSP), 2015 IEEE International Conference on*, 2015, pp. 196–200.

[18] L. Alparone, B. Aiazzi, S. Baronti, and A. Garzelli, *Remote sensing image fusion*. Crc Press, 2015.

[19] C. M. Bishop, *Neural Networks for Pattern Recognition*. Clarendon Press, 1995. [Online]. Available: https://books.google.es/books?id=-aAwQO_-rXwC

[20] J. A. Benediktsson, P. H. Swain, and O. K. Ersoy, "Conjugate gradient neural networks in classification of very high dimensional remote sensing data," *International Journal of Remote Sensing*, vol. 14, no. 15, pp. 2883–2903, 1993. [Online]. Available: http://www.tandfonline.com/doi/abs/10.1080/01431169308904316

[21] P. M. Atkinson and A. R. L. Tatnall, "Introduction Neural networks in remote sensing," *International Journal of Remote Sensing*, vol. 18, no. 4, pp. 699–709, 1997. [Online]. Available: http://dx.doi.org/10.1080/014311697218700

[22] H. Yang, "A back-propagation neural network for mineralogical mapping from AVIRIS data," *International Journal of Remote Sensing*, vol. 20, no. 1, pp. 97–110, 1999. [Online]. Available: http://dx.doi.org/10.1080/014311699213622

[23] Y. LeCun, Y. Bengio, and G. Hinton, "Deep Learning," *Nature*, vol. 521, p. 436444, 2015.

[24] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.

[25] Y. Bengio, "Learning Deep Architectures for AI," *Machine Learning*, vol. 2, no. 1, pp. 1–127, 2009.

[26] M. Sharma, S. Chaudhury, and B. Lall, "Deep learning based frameworks for image super-resolution and noise-resilient super-resolution," in *2017 International Joint Conference on Neural Networks (IJCNN)*, May 2017, pp. 744–751.

[27] K. Makantasis, K. Karantzalos, A. Doulamis, and N. Doulamis, "Deep supervised learning for hyperspectral data classification through convolutional neural networks," in *2015 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, 2015, pp. 4959–4962.

[28] M. Castelluccio, G. Poggi, C. Sansone, and L. Verdoliva, "Land use classification in remote sensing images by convolutional neural networks," *CoRR*, vol. abs/1508.00092, 2015. [Online]. Available: http://arxiv.org/abs/1508.00092

[29] Y. Chen, H. Jiang, C. Li, X. Jia, and P. Ghamisi, "Deep Feature Extraction and Classification of Hyperspectral Images Based on Convolutional Neural Networks," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 10, pp. 6232–6251, 2016. [Online]. Available: http://ieeexplore.ieee.org/document/7514991/

[30] M. E. Paoletti, J. M. Haut, J. Plaza, and A. Plaza, "A new deep convolutional neural network for fast hyperspectral image classification," *ISPRS Journal of Photogrammetry and Remote Sensing*, 2017.

[31] Q. Yuan, Y. Wei, X. Meng, H. Shen, and L. Zhang, "A multiscale and multidepth convolutional neural network for remote sensing imagery pan-sharpening," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2018.

[32] R. Dian, S. Li, A. Guo, and L. Fang, "Deep hyperspectral image sharpening," *IEEE Transactions on Neural Networks and Learning Systems*, 2018.

[33] Q. Zhang, Q. Yuan, C. Zeng, X. Li, and Y. Wei, "Missing data reconstruction in remote sensing image with a unified spatial-temporal-spectral deep convolutional neural network," *IEEE Transactions on Geoscience and Remote Sensing*, 2018.

[34] L. Liebel and M. Krner, "Single-image super resolution for multispectral remote sensing data using convolutional neural networks," vol. XLI-B3, pp. 883–890, 06 2016.

[35] C. Dong, C. C. Loy, K. He, and X. Tang, "Image super-resolution using deep convolutional networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 2, pp. 295–307, Feb 2016.

[36] T. Y. Han, Y. J. Kim, and B. C. Song, "Convolutional neural network-based infrared image super resolution under low light environment," in *2017 25th European Signal Processing Conference (EUSIPCO)*, Aug 2017, pp. 803–807.

[37] C. Li, Z. Ren, B. Yang, X. Wan, and J. Wang, "Texture-centralized deep convolutional neural network for single image super resolution," in *2017 Chinese Automation Congress (CAC)*, Oct 2017, pp. 3707–3710.

[38] X. Du, Y. He, J. Li, and X. Xie, "Single image super-resolution via multi-scale fusion convolutional neural network," in *2017 IEEE 8th International Conference on Awareness Science and Technology (iCAST)*, Nov 2017, pp. 544–551.

[39] O. Frat and F. T. Y. Vural, "Representation learning with convolutional sparse autoencoders for remote sensing," in *2013 21st Signal Processing and Communications Applications Conference (SIU)*, April 2013, pp. 1–4.

[40] W. Cui, Q. Zhou, and Z. Zheng, "Application of a Hybrid Model Based on a Convolutional Auto-Encoder and Convolutional Neural Network in Object-Oriented Remote Sensing Classification," *Algorithms*, vol. 11, no. 1, p. 9, 1 2018. [Online]. Available: http://www.mdpi.com/1999-4893/11/1/9

[41] R. Tanase, M. Datcu, and D. Raducanu, "A convolutional deep belief network for polarimetric sar data feature extraction," in *2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, July 2016, pp. 7545–7548.

[42] D. Lin, K. Fu, Y. Wang, G. Xu, and X. Sun, "Marta gans: Unsupervised representation learning for remote sensing image classification," *IEEE Geoscience and Remote Sensing Letters*, vol. 14, no. 11, pp. 2092–2096, Nov 2017.

[43] Z. Zuo, B. Shuai, G. Wang, X. Liu, X. Wang, B. Wang, and Y. Chen, "Convolutional recurrent neural networks: Learning spatial dependencies for image representation," in *2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, June 2015, pp. 18–26.

[44] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015, pp. 3431–3440.

[45] A. Newell, K. Yang, and J. Deng, "Stacked hourglass networks for human pose estimation," *CoRR*, vol. abs/1603.06937, 2016. [Online]. Available: http://arxiv.org/abs/1603.06937

[46] I. Melekhov, J. Ylioinas, J. Kannala, and E. Rahtu, "Image-based localization using hourglass networks," *CoRR*, vol. abs/1703.07971, 2017. [Online]. Available: http://arxiv.org/abs/1703.07971

[47] M. Paoletti, J. Haut, J. Plaza, and A. Plaza, "Yinyang K-means clustering for hyperspectral image analysis," in *Proceedings of the 17th International Conference on Computational and Mathematical Methods in Science and Engineering*, J. Vigo-Aguiar, Ed., Rota, 2017, pp. 1625–1636.

[48] C. González, S. Sánchez, A. Paz, J. Resano, D. Mozos, and A. Plaza, "Use of FPGA or GPU-based architectures for remotely sensed hyperspectral image processing," *Integration, the VLSI Journal*, vol. 46, no. 2, pp. 89 – 103, 2013.

[49] A. Plaza, J. Plaza, A. Paz, and S. Sanchez, "Parallel Hyperspectral Image and Signal Processing," *IEEE Signal Processing Magazine*, vol. 28, no. 3, pp. 119–126, 2011.

[50] J. Haut, M. Paoletti, J. Plaza, and A. Plaza, "Cloud implementation of the K-means algorithm for hyperspectral image analysis," *Journal of Supercomputing*, vol. 73, no. 1, 2017.

[51] J. M. Haut, M. E. Paoletti, A. Paz-Gallardo, J. Plaza, and A. Plaza, "Cloud implementation of logistic regression for hyperspectral image classification," in *Proceedings of the 17th International Conference on Computational and Mathematical Methods in Science and Engineering, CMMSE*, 2017, pp. 1063–2321.

[52] A. J. Plaza, C.-I. Chang, L. Di, and Y. Bai, *High Performance Computing in Remote Sensing Book Review Book Review*, Taylor & Francis, Ed. Boca Raton, Florida: Chapman & Hall/CRC Press, Computer & Information Science Series, 2008.

[53] K. Nasrollahi and T. B. Moeslund, "Super-resolution: a comprehensive survey," *Machine vision and applications*, vol. 25, no. 6, pp. 1423–1468, 2014.

[54] R. Fernandez-Beltran, P. Latorre-Carmona, and F. Pla, "Single-frame super-resolution in remote sensing: a practical overview," *International Journal of Remote Sensing*, vol. 38, no. 1, pp. 314–354, 2017.

[55] M. Irani and S. Peleg, "Improving resolution by image registration," *CVGIP: Graph. Models Image Process.*, vol. 53, no. 3, pp. 231–239, 1991.

[56] J. Sun, Z. Xu, and H.-Y. Shum, "Image super-resolution using gradient profile prior," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[57] L. Lucy, "An iterative technique for the rectification of observed distributions," *The Astronomical Journal*, vol. 79, p. 745, 1974.

[58] R. C. Gonzalez and R. E. Woods, *Digital Image Processing (3rd Edition)*. Prentice-Hall, Inc., 2006.

[59] G. Deng, "A generalized unsharp masking algorithm," *IEEE transactions on Image Processing*, vol. 20, no. 5, pp. 1249–1261, 2011.

[60] J. Yang, J. Wright, T. S. Huang, and Y. Ma, "Image super-resolution via sparse representation," *IEEE Trans. Img. Proc.*, vol. 19, no. 11, pp. 2861–2873, 2010.

[61] R. Timofte, V. De Smet, and L. Van Gool, "Anchored neighborhood regression for fast example-based super-resolution," in *IEEE International Conference on Computer Vision*, 2013, pp. 1920–1927.

[62] C. Dong, C. Loy, K. He, and X. Tang, "Learning a deep convolutional network for image super-resolution," in *European Conference on Computer Vision*, 2014, vol. 8692, pp. 184–199.

[63] G. Polatkan, M. Zhou, L. Carin, D. Blei, and I. Daubechies, "A bayesian nonparametric approach to image super-resolution," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 2, pp. 346–358, 2015.

[64] D. Glasner, S. Bagon, and M. Irani, "Super-resolution from a single image," in *IEEE International Conference on Computer Vision*, 2009.

[65] J. Huang, S. A., and A. N., "Single image super-resolution from

transformed self-exemplars," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 5197–5206.

[66] T. Michaeli and M. Irani, "Blind deblurring using internal patch recurrence," in *European Conference on Computer Vision*, 2014, pp. 783–798.

[67] S. Baker and T. Kanade, "Limits on super-resolution and how to break them," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 9, pp. 1167–1183, 2002.

[68] V. Syrris, S. Ferri, D. Ehrlich, and M. Pesaresi, "Image enhancement and feature extraction based on low-resolution satellite data," *Ieee Journal Of Selected Topics In Applied Earth Observations And Remote Sensing*, vol. 8, no. 5, pp. 1986–1995, 2015.

[69] C. Dong, C. C. Loy, K. He, and X. Tang, "Image super-resolution using deep convolutional networks," *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 2, pp. 295–307, 2016.

[70] J. Kim, J. Kwon Lee, and K. Mu Lee, "Accurate image super-resolution using very deep convolutional networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1646–1654.

[71] B. Lim, S. Son, H. Kim, S. Nah, and K. M. Lee, "Enhanced deep residual networks for single image super-resolution," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, vol. 1, no. 2, 2017, p. 3.

[72] F. A. Mianji, Y. Gu, Y. Zhang, and J. Zhang, "Enhanced self-training superresolution mapping technique for hyperspectral imagery," *IEEE Geoscience and Remote Sensing Letters*, vol. 8, no. 4, pp. 671–675, 2011.

[73] J. M. Bioucas-Dias, A. Plaza, N. Dobigeon, M. Parente, Q. Du, P. Gader, and J. Chanussot, "Hyperspectral unmixing overview: Geometrical, statistical, and sparse regression-based approaches," *IEEE journal of selected topics in applied earth observations and remote sensing*, vol. 5, no. 2, pp. 354–379, 2012.

[74] W. Hu, Y. Huang, L. Wei, F. Zhang, and H. Li, "Deep convolutional neural networks for hyperspectral image classification," *Journal of Sensors*, 2015.

[75] M. E. Paoletti, J. M. Haut, J. Plaza, A. Plaza, Q. Liu, and R. Hang, "Multicore Implementation of the Multi-Scale Adaptive Deep Pyramid Matching Model for Remotely Sensed Image Classification," in *2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, Fort Worth, Texas, 2017, pp. 2247–2250.

[76] S. Yu, S. Jia, and C. Xu, "Convolutional neural networks for hyperspectral image classification," *Neurocomputing*, vol. 219, pp. 88–98, 1 2017. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S0925231216310104

[77] V. Nair and G. E. Hinton, "Rectified Linear Units Improve Restricted Boltzmann Machines," in *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, Johannes Fürnkranz and Thorsten Joachims, Ed. Omnipress, 2010, pp. 807–814.

[78] B. Xu, N. Wang, T. Chen, and M. Li, "Empirical evaluation of rectified activations in convolutional network," *arXiv preprint arXiv:1505.00853*, 2015.

[79] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier Nonlinearities Improve Neural Network Acoustic Models," in *International Conference on Machine Learning (ICML)*, 2013, p. 30.

[80] D. Ulyanov, A. Vedaldi, and V. S. Lempitsky, "Deep image prior," *CoRR*, vol. abs/1711.10925, 2017. [Online]. Available: http://arxiv.org/abs/1711.10925

[81] K. Turkowski, "Graphics gems," A. S. Glassner, Ed. Academic Press Professional, Inc., 1990, ch. Filters for Common Resampling Tasks, pp. 147–165.

[82] D. P. Kingma and J. L. Ba, "ADAM: {A} method for stochastic optimization," *CoRR*, 2014. [Online]. Available: http://arxiv.org/abs/1412.6980

[83] Q. Huynh-Thu and M. Ghanbari, "Scope of validity of psnr in image/video quality assessment," *Electronics Letters*, vol. 44, no. 13, pp. 800–801, June 2008.

[84] R. H. Yuhas, A. F. Goetz, and J. W. Boardman, "Discrimination among semi-arid landscape endmembers using the spectral angle mapper (sam) algorithm," 1992.

[85] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, April 2004.

[86] M. A. Veganzones, M. Simes, G. Licciardi, N. Yokoya, J. M. Bioucas-Dias, and J. Chanussot, "Hyperspectral super-resolution of locally low rank images from complementary multisource data," *IEEE Transactions on Image Processing*, vol. 25, no. 1, pp. 274–288, Jan 2016.

[87] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," 2017.

[88] C. Ledig, L. Theis, F. Huszr, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi, "Photo-realistic single image super-resolution using a generative adversarial network," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017, pp. 105–114.

[89] Y. Yang and S. Newsam, "Bag-of-visual-words and spatial extensions for land-use classification," in *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems*, ser. GIS '10. New York, NY, USA: ACM, 2010, pp. 270–279. [Online]. Available: http://doi.acm.org/10.1145/1869790.1869829

[90] Q. Zou, L. Ni, T. Zhang, and Q. Wang, "Deep learning based feature selection for remote sensing scene classification," *IEEE*

Geoscience and Remote Sensing Letters, vol. 12, no. 11, pp. 2321–2325, Nov 2015.

[91] G. Cheng, J. Han, and X. Lu, "Remote sensing image scene classification: Benchmark and state of the art," *Proceedings of the IEEE*, vol. 105, no. 10, pp. 1865–1883, Oct 2017.

[92] K. Turkowski, "Filters for common resampling tasks," in *Graphics gems*. Academic Press Professional, Inc., 1990, pp. 147–165.

[93] G. Freedman and R. Fattal, "Image and video upscaling from local self-examples," *ACM Trans. Graph.*, vol. 30, no. 2, pp. 1–11, 2011.

[94] H. He and W.-C. Siu, "Single image super-resolution using gaussian process regression." in *IEEE Conference on Computer Vision and Pattern Recognition*, 2011, pp. 449–456.

[95] N. Zhao, Q. Wei, A. Basarab, N. Dobigeon, D. Kouamé, and J.-Y. Tourneret, "Fast single image super-resolution using a new analytical solution for l2-l2 problems," *IEEE Transactions on Image Processing*, vol. 25, no. 8, pp. 3683–3697, 2016.

TABLE V

SR RESULTS FOR TEST IMAGES FROM 1 TO 4. THE BEST RESULT FOR EACH IMAGE, SCALING RATIO AND METRIC IS HIGHLIGHTED IN BOLD FONT.

Image	Method	Ratio 2x							Ratio 4x						
		TIME	NRMSE	PSNR	ERGAS	Qindex	SSIM	SAM	TIME	NRMSE	PSNR	ERGAS	Qindex	SSIM	SAM
agricultural	**BCI**	**0.04**	0.0792	24.96	6.174	0.5814	0.5683	0.0171	**0.00**	0.0957	23.31	3.719	0.2660	0.2681	**0.0208**
	IBP	0.13	0.0775	25.14	6.050	0.6255	0.6151	0.0176	0.35	0.0947	23.41	3.683	0.3473	0.3440	0.0214
	GPP	15.99	0.0800	24.87	6.245	0.5836	0.5706	0.0180	11.66	0.0953	23.35	3.705	0.2790	0.2785	0.0212
	SRI	211.50	0.0792	24.96	6.183	0.6242	0.6216	0.0180	212.77	0.0948	23.39	3.695	**0.3906**	**0.3782**	0.0223
	LSE	763.52	0.0819	24.66	6.348	0.6086	0.6067	0.0187	259.15	0.1000	22.93	3.811	0.3014	0.3252	0.0235
	GPR	167.33	0.0903	23.81	7.064	0.4016	0.4032	0.0197	88.07	0.0972	23.18	3.778	0.2458	0.2561	0.0217
	BDB	147.94	0.0983	23.08	7.695	0.4428	0.4498	0.0215	250.55	0.1138	21.80	4.447	0.1351	0.1698	0.0255
	DLU	0.10	0.0784	25.04	6.124	0.6148	0.6037	0.0177	0.07	0.0953	23.35	3.706	0.2827	0.2806	0.0213
	DRE	0.07	0.0784	25.04	6.124	0.6148	0.6037	0.0177	0.03	0.0953	23.35	3.706	0.2827	0.2805	0.0213
	FSR	0.55	0.0786	25.02	6.136	0.5879	0.5867	0.0180	0.62	0.1001	22.92	3.899	0.1695	0.2216	0.0225
	TSE	13.84	0.0776	25.13	6.057	0.6495	**0.6475**	**0.0168**	13.18	0.0944	23.43	**3.670**	0.3357	0.3304	0.0209
	UMK	0.05	0.0785	25.03	6.132	0.6154	0.6043	0.0177	0.01	0.0946	23.41	3.681	0.3238	0.3171	0.0213
	Ours	232.87	**0.0755**	**25.37**	**5.922**	**0.6513**	0.6403	0.0173	127.75	**0.0942**	**23.45**	3.679	0.3366	0.3277	0.0211
agricultural2	**BCI**	**0.01**	0.0386	36.39	1.782	0.8263	0.8113	**0.0053**	**0.01**	0.0749	30.65	1.733	0.4539	0.4390	0.0080
	IBP	0.10	0.0347	37.32	1.602	0.8616	0.8491	0.0071	0.35	0.0693	31.31	1.603	0.5905	0.5927	0.0092
	GPP	19.90	0.0375	36.66	1.729	0.8299	0.8139	0.0071	14.52	0.0742	30.73	1.717	0.4642	0.4480	0.0092
	SRI	229.18	0.0333	37.67	1.537	0.8644	0.8517	0.0071	210.29	0.0772	30.37	1.785	0.5790	0.5603	0.0094
	LSE	762.90	0.0409	35.89	1.874	0.8378	0.8258	0.0071	259.40	0.0741	30.74	1.679	0.5799	0.6033	0.0111
	GPR	163.57	0.0588	32.75	2.720	0.6357	0.6189	0.0073	85.87	0.0805	30.02	1.862	0.3927	0.3810	0.0093
	BDB	145.97	0.0768	30.42	3.561	0.6350	0.6444	0.0079	247.64	0.1001	28.13	2.316	0.4000	0.4324	0.0099
	DLU	0.07	0.0349	37.28	1.608	0.8599	0.8468	0.0071	0.06	0.0718	31.01	1.662	0.4981	0.4782	0.0092
	DRE	0.03	0.0349	37.28	1.608	0.8598	0.8468	0.0071	0.03	0.0718	31.01	1.662	0.4981	0.4782	0.0092
	FSR	0.47	0.0419	35.68	1.938	0.7970	0.7897	0.0071	0.58	0.0884	29.20	2.054	0.2691	0.3158	0.0096
	TSE	11.99	**0.0312**	**38.24**	**1.439**	**0.8820**	**0.8709**	0.0055	11.92	0.0583	32.81	1.346	0.6393	0.6313	**0.0078**
	UMK	0.01	0.0350	37.26	1.613	0.8600	0.8483	0.0071	0.01	0.0683	31.45	1.578	0.5646	0.5480	0.0091
	Ours	232.62	0.0324	37.92	1.501	0.8732	0.8612	0.0057	127.47	**0.0554**	**33.26**	**1.284**	**0.6571**	**0.6436**	0.0080
airplane	**BCI**	**0.00**	0.0323	30.02	3.058	0.8125	0.8958	0.0092	**0.00**	0.0536	25.63	2.534	0.5385	0.7508	0.0145
	IBP	0.12	0.0297	30.76	2.811	0.8345	0.9079	0.0104	0.36	0.0511	26.04	2.420	0.5969	0.7637	0.0158
	GPP	15.76	0.0320	30.11	3.030	0.8066	0.8935	0.0104	12.16	0.0525	25.81	2.484	0.5426	0.7550	0.0151
	SRI	208.46	0.0256	32.04	2.429	0.8444	0.9173	0.0095	208.54	0.0535	25.65	2.533	0.5849	0.7647	0.0174
	LSE	770.02	0.0314	30.26	2.950	0.8227	0.9036	0.0112	259.70	0.0592	24.76	2.751	0.5480	0.7546	0.0176
	GPR	170.60	0.0437	27.40	4.132	0.6756	0.8194	0.0113	77.84	0.0570	25.10	2.695	0.4891	0.7239	0.0156
	BDB	144.17	0.0581	24.92	5.490	0.6850	0.8133	0.0139	245.76	0.0929	20.84	4.391	0.3362	0.6170	0.0188
	DLU	0.07	0.0298	30.71	2.825	0.8321	0.9063	0.0105	0.06	0.0517	25.93	2.449	0.5532	0.7556	0.0151
	DRE	0.03	0.0298	30.72	2.825	0.8321	0.9063	0.0105	0.03	0.0517	25.93	2.449	0.5532	0.7556	0.0151
	FSR	0.45	0.0373	28.77	3.533	0.7657	0.8721	0.0107	1.25	0.0686	23.48	3.246	0.3353	0.6706	0.0168
	TSE	12.86	0.0256	32.06	2.421	**0.8639**	**0.9249**	0.0089	12.30	0.0455	27.04	2.156	**0.6139**	**0.7920**	0.0145
	UMK	0.01	0.0298	30.72	2.822	0.8327	0.9066	0.0106	0.01	0.0505	26.14	2.392	0.5851	0.7634	0.0153
	Ours	232.25	**0.0252**	**32.17**	**2.395**	0.8487	0.9178	**0.0088**	127.17	**0.0426**	**27.61**	**2.026**	0.6046	0.7889	**0.0129**
baseball	**BCI**	**0.00**	0.0229	33.43	2.531	0.8554	0.8828	0.0261	**0.00**	0.0440	27.77	2.393	0.5387	0.6451	0.0405
	IBP	0.11	0.0207	34.33	2.314	0.8780	0.9024	0.0270	0.35	0.0422	28.13	2.312	**0.6121**	0.6946	0.0414
	GPP	16.72	0.0229	33.43	2.542	0.8545	0.8829	0.0268	11.97	0.0434	27.89	2.361	0.5476	0.6472	0.0407
	SRI	210.27	0.0211	34.17	2.383	0.8714	0.8963	0.0275	207.94	0.0474	27.12	2.633	0.5808	0.6655	0.0432
	LSE	769.38	0.0254	32.54	2.819	0.8481	0.8786	0.0276	260.18	0.0498	26.69	2.695	0.5390	0.6541	0.0464
	GPR	158.75	0.0334	30.16	3.692	0.7213	0.7856	0.0275	76.68	0.0466	27.26	2.543	0.4936	0.6181	0.0409
	BDB	144.95	0.0442	27.73	4.973	0.7232	0.7731	0.0306	245.32	0.0751	23.12	4.189	0.3146	0.4817	0.0459
	DLU	0.06	0.0207	34.32	2.318	0.8774	0.9018	0.0270	0.07	0.0429	27.99	2.335	0.5624	0.6584	0.0408
	DRE	0.03	0.0207	34.32	2.318	0.8774	0.9018	0.0270	0.03	0.0429	27.99	2.335	0.5624	0.6584	0.0408
	FSR	0.52	0.0287	31.47	3.186	0.8081	0.8490	0.0275	0.83	0.0554	25.76	3.038	0.3548	0.5581	0.0422
	TSE	12.69	**0.0187**	**35.18**	**2.083**	**0.9038**	**0.9197**	**0.0237**	11.92	0.0409	28.41	2.224	0.6085	**0.6967**	**0.0395**
	UMK	0.01	0.0206	34.34	2.314	0.8779	0.9022	0.0270	0.01	0.0419	28.20	2.286	0.5990	0.6854	0.0410
	Ours	231.37	0.0197	34.75	2.206	0.8935	0.9107	0.0248	127.15	**0.0400**	**28.59**	**2.195**	0.5971	0.6775	0.0400

TABLE VI

SR RESULTS FOR TEST IMAGES FROM 5 TO 8. THE BEST RESULT FOR EACH IMAGE, SCALING RATIO AND METRIC IS HIGHLIGHTED IN BOLD FONT.

Image	Method	Ratio 2x							Ratio 4x						
		TIME	NRMSE	PSNR	ERGAS	Qindex	SSIM	SAM	TIME	NRMSE	PSNR	ERGAS	Qindex	SSIM	SAM
bridge	**BCI**	**0.01**	0.0420	27.54	6.547	0.7783	0.8709	**0.0074**	**0.01**	0.0603	24.40	4.701	0.4862	0.7153	**0.0100**
	IBP	0.12	0.0390	28.18	6.079	0.7855	0.8913	0.0091	0.35	0.0588	24.62	4.581	0.5182	0.7577	0.0121
	GPP	18.40	0.0422	27.50	6.576	0.7549	0.8708	0.0093	13.10	0.0597	24.48	4.657	0.4752	0.7249	0.0115
	SRI	210.07	**0.0359**	**28.89**	**5.601**	0.7975	0.9044	0.0088	209.12	0.0606	24.35	4.722	0.5165	0.7667	0.0153
	LSE	770.12	0.0412	27.70	6.359	0.7569	0.8837	0.0096	260.15	0.0600	24.44	4.566	0.4527	0.7528	0.0151
	GPR	185.47	0.0525	25.60	8.184	0.6234	0.7910	0.0099	89.35	0.0631	24.00	4.909	0.4293	0.6821	0.0118
	BDB	144.85	0.0725	22.80	11.308	0.5311	0.7520	0.0128	246.05	0.0869	21.22	6.785	0.2427	0.6368	0.0141
	DLU	0.06	0.0391	28.16	6.092	0.7831	0.8902	0.0091	0.06	0.0593	24.54	4.622	0.4821	0.7263	0.0115
	DRE	0.03	0.0391	28.16	6.092	0.7832	0.8902	0.0091	0.03	0.0593	24.54	4.622	0.4821	0.7263	0.0115
	FSR	0.66	0.0470	26.56	7.328	0.6947	0.8472	0.0098	1.17	0.0698	23.13	5.439	0.2694	0.6789	0.0127
	TSE	12.33	0.0361	28.86	5.623	**0.8280**	**0.9068**	0.0076	12.30	0.0559	25.05	**4.361**	**0.5475**	**0.7720**	0.0109
	UMK	0.01	0.0391	28.16	6.093	0.7834	0.8903	0.0091	0.01	0.0584	24.67	4.556	0.5073	0.7508	0.0117
	Ours	231.72	0.0362	28.82	5.681	0.7812	0.9010	0.0094	126.79	**0.0556**	**25.09**	4.365	0.4815	0.7554	0.0119
circular-farmland	**BCI**	**0.00**	0.0290	30.75	5.429	0.8571	0.8901	0.0180	**0.00**	0.0481	26.35	4.506	0.5906	0.7094	0.0286
	IBP	0.12	0.0263	31.60	4.923	0.8823	0.9103	0.0204	0.35	0.0462	26.71	4.320	0.6379	0.7372	0.0350
	GPP	16.67	0.0288	30.81	5.392	0.8545	0.8900	0.0211	11.32	0.0469	26.58	4.390	0.5997	0.7173	0.0302
	SRI	206.04	0.0239	32.44	4.472	0.8902	0.9193	0.0193	207.14	0.0484	26.30	4.529	0.6410	0.7476	0.0366
	LSE	772.64	0.0297	30.55	5.490	0.8645	0.8964	0.0209	259.57	0.0480	26.37	4.370	0.6109	0.7433	0.0378
	GPR	160.94	0.0401	27.95	7.499	0.7136	0.7911	0.0243	76.44	0.0513	25.79	4.812	0.5441	0.6786	0.0313
	BDB	144.76	0.0578	24.77	10.822	0.6824	0.7656	0.0302	246.43	0.0951	20.44	8.889	0.3422	0.5214	0.0462
	DLU	0.07	0.0264	31.57	4.942	0.8805	0.9089	0.0205	0.06	0.0467	26.62	4.371	0.6047	0.7202	0.0307
	DRE	0.03	0.0264	31.57	4.942	0.8805	0.9089	0.0205	0.03	0.0467	26.62	4.371	0.6047	0.7202	0.0307
	FSR	0.47	0.0345	29.23	6.467	0.8120	0.8655	0.0224	1.20	0.0618	24.19	5.850	0.4218	0.6255	0.0348
	TSE	12.45	**0.0227**	**32.87**	**4.252**	**0.9039**	**0.9270**	**0.0168**	12.33	0.0400	27.96	3.741	**0.6776**	**0.7725**	**0.0269**
	UMK	0.01	0.0264	31.57	4.939	0.8808	0.9092	0.0206	0.01	0.0457	26.80	4.277	0.6285	0.7349	0.0331
	Ours	231.78	0.0231	32.72	4.357	0.8964	0.9191	0.0186	126.49	**0.0393**	**28.11**	**3.707**	0.6542	0.7577	0.0288
harbor	**BCI**	**0.00**	0.0909	21.73	11.416	0.8244	0.8936	0.0336	**0.01**	0.1795	15.81	11.307	0.4593	0.5842	**0.0465**
	IBP	0.13	0.0747	23.43	9.363	0.8446	0.9239	0.0340	0.34	0.1677	16.41	10.523	0.5244	0.6419	0.0513
	GPP	17.28	0.0866	22.15	10.872	0.8161	0.9007	0.0358	12.28	0.1761	15.98	11.091	0.4668	0.5973	0.0482
	SRI	203.16	0.0567	25.82	7.128	0.8715	0.9523	0.0321	210.48	0.1707	16.25	10.727	0.5554	0.6826	0.0549
	LSE	767.89	0.0908	21.73	11.119	0.8335	0.9117	0.0349	260.34	0.1800	15.79	10.988	0.5009	0.6283	0.0564
	GPR	185.44	0.1396	18.00	17.509	0.6608	0.7532	0.0400	80.39	0.1905	15.30	11.981	0.4069	0.5463	0.0489
	BDB	144.61	0.1797	15.81	22.522	0.6390	0.7491	0.0508	252.04	0.2811	11.92	17.763	0.2495	0.4221	0.0626
	DLU	0.07	0.0750	23.40	9.398	0.8424	0.9222	0.0343	0.07	0.1733	16.12	10.913	0.4750	0.6043	0.0495
	DRE	0.03	0.0750	23.40	9.398	0.8424	0.9221	0.0345	0.03	0.1733	16.12	10.913	0.4749	0.6043	0.0495
	FSR	0.33	0.1062	20.38	13.322	0.7707	0.8672	0.0369	1.81	0.2187	14.10	13.869	0.2914	0.4727	0.0510
	TSE	13.27	0.0580	25.63	7.294	**0.8949**	0.9539	**0.0289**	12.79	0.1648	16.56	10.375	**0.5594**	0.6712	0.0499
	UMK	0.00	0.0746	23.44	9.341	0.8428	0.9227	0.0343	0.01	0.1680	16.39	10.562	0.5104	0.6321	0.0511
	Ours	231.16	**0.0505**	**26.84**	**6.352**	0.8731	**0.9592**	0.0320	127.59	**0.1491**	**17.43**	**9.422**	0.5517	**0.7096**	0.0485
industry	**BCI**	**0.01**	0.0226	33.38	1.660	0.8162	0.9220	**0.0028**	**0.01**	0.0445	27.49	1.636	0.4595	0.7720	**0.0046**
	IBP	0.25	0.0196	34.63	1.438	0.8536	0.9365	0.0046	0.87	0.0418	28.03	1.538	0.5380	0.8066	0.0067
	GPP	53.63	0.0218	33.67	1.607	0.8110	0.9211	0.0047	34.96	0.0432	27.74	1.590	0.4666	0.7775	0.0062
	SRI	729.33	0.0182	35.28	1.335	0.8542	0.9375	0.0045	710.10	0.0471	26.99	1.734	0.5287	0.8027	0.0072
	LSE	1756.30	0.0286	31.34	2.082	0.8031	0.9191	0.0049	601.97	0.0556	25.56	2.009	0.4517	0.7828	0.0086
	GPR	438.63	0.0347	29.65	2.553	0.6489	0.8479	0.0051	177.01	0.0500	26.47	1.839	0.4229	0.7488	0.0064
	BDB	320.45	0.0475	26.92	3.497	0.6824	0.8513	0.0060	465.03	0.1038	20.13	3.820	0.2000	0.6283	0.0093
	DLU	0.20	0.0197	34.58	1.446	0.8516	0.9355	0.0046	0.19	0.0428	27.83	1.574	0.4767	0.7798	0.0062
	DRE	0.09	0.0197	34.58	1.446	0.8516	0.9355	0.0046	0.11	0.0428	27.83	1.574	0.4767	0.7798	0.0062
	FSR	1.19	0.0287	31.30	2.109	0.7494	0.8975	0.0048	3.53	0.0586	25.09	2.151	0.2682	0.7219	0.0067
	TSE	33.38	**0.0171**	**35.79**	**1.259**	**0.8752**	**0.9495**	0.0028	31.35	0.0395	28.52	1.453	**0.5431**	**0.8239**	0.0056
	UMK	0.01	0.0196	34.60	1.444	0.8524	0.9357	0.0046	0.01	0.0414	28.11	1.523	0.5176	0.7986	0.0064
	Ours	483.64	0.0194	34.69	1.432	0.8328	0.9301	0.0043	244.28	**0.0361**	**29.29**	**1.334**	0.4635	0.7961	0.0060

TABLE VII

SR RESULTS FOR TEST IMAGES FROM 9 TO 12. THE BEST RESULT FOR EACH IMAGE, SCALING RATIO AND METRIC IS HIGHLIGHTED IN BOLD FONT.

Image	Method	Ratio 2x							Ratio 4x						
		TIME	NRMSE	PSNR	ERGAS	Qindex	SSIM	SAM	TIME	NRMSE	PSNR	ERGAS	Qindex	SSIM	SAM
intersection	**BCI**	**0.00**	0.1024	19.79	16.184	0.7327	0.7444	0.0323	**0.00**	0.1539	16.26	12.163	0.3438	0.3845	**0.0405**
	IBP	0.13	0.0896	20.95	14.152	0.7961	0.8011	0.0339	0.34	0.1501	16.48	11.847	0.4446	0.4498	0.0437
	GPP	19.19	0.1038	19.68	16.404	0.7325	0.7422	0.0340	12.55	0.1526	16.33	12.061	0.3599	0.3997	0.0413
	SRI	209.92	0.0773	22.24	12.214	0.8347	0.8384	0.0331	213.57	0.1518	16.38	11.976	**0.4589**	0.4635	0.0449
	LSE	768.00	0.0902	20.90	14.128	0.7965	0.8009	0.0337	259.93	0.1525	16.34	11.731	0.4126	0.4382	0.0456
	GPR	153.27	0.1371	17.26	21.683	0.4889	0.5276	0.0366	74.79	0.1580	16.03	12.527	0.2949	0.3368	0.0419
	BDB	146.49	0.1486	16.56	23.544	0.5540	0.5816	0.0396	247.34	0.1907	14.39	15.112	0.1617	0.2384	0.0482
	DLU	0.06	0.0894	20.97	14.118	0.7938	0.7983	0.0339	0.07	0.1520	16.36	12.014	0.3728	0.4054	0.0414
	DRE	0.03	0.0894	20.98	14.113	0.7938	0.7983	0.0340	0.03	0.1520	16.36	12.014	0.3728	0.4054	0.0414
	FSR	0.60	0.1136	18.89	17.960	0.6849	0.7027	0.0351	1.36	0.1685	15.47	13.369	0.1733	0.2807	0.0433
	TSE	12.73	0.0747	22.54	11.804	0.8449	0.8467	**0.0313**	12.40	0.1449	16.78	11.458	0.4516	**0.4728**	0.0408
	UMK	0.01	0.0893	20.98	14.104	0.7943	0.7989	0.0341	0.01	0.1502	16.47	11.864	0.4199	0.4333	0.0426
	Ours	230.74	**0.0684**	**23.30**	**10.863**	**0.8496**	**0.8516**	0.0332	126.27	**0.1440**	**16.84**	**11.439**	0.4467	0.4655	0.0405
parking	**BCI**	**0.01**	0.0566	24.94	6.919	0.7762	0.8054	0.0181	**0.01**	0.0824	21.68	5.040	0.4701	0.5754	**0.0267**
	IBP	0.26	0.0530	25.52	6.473	0.8111	0.8338	0.0192	0.86	0.0801	21.93	4.896	**0.5427**	0.6099	0.0284
	GPP	51.93	0.0564	24.97	6.897	0.7781	0.8069	0.0199	32.94	0.0816	21.77	4.990	0.4795	0.5812	0.0273
	SRI	728.21	**0.0489**	**26.21**	**5.981**	0.8350	0.8558	0.0174	718.42	0.0851	21.41	5.199	0.5286	0.5946	0.0293
	LSE	1762.70	0.0588	24.62	7.126	0.7812	0.8133	0.0214	601.40	0.0878	21.13	5.270	0.4753	0.5728	0.0331
	GPR	364.76	0.0717	22.88	8.772	0.6078	0.6772	0.0230	143.54	0.0856	21.35	5.238	0.4167	0.5432	0.0285
	BDB	326.70	0.0906	20.86	11.086	0.5741	0.6432	0.0269	476.06	0.1241	18.12	7.593	0.2304	0.3971	0.0376
	DLU	0.19	0.0533	25.47	6.514	0.8077	0.8307	0.0194	0.19	0.0809	21.84	4.948	0.4909	0.5851	0.0275
	DRE	0.09	0.0533	25.47	6.514	0.8077	0.8306	0.0194	0.10	0.0809	21.84	4.948	0.4909	0.5851	0.0275
	FSR	1.43	0.0614	24.23	7.511	0.7404	0.7807	0.0211	3.63	0.0976	20.21	5.960	0.2700	0.4673	0.0322
	TSE	31.00	0.0494	26.12	6.042	**0.8422**	**0.8598**	**0.0174**	32.04	0.0796	21.98	4.869	0.5384	**0.6132**	0.0276
	UMK	0.01	0.0533	25.47	6.512	0.8086	0.8314	0.0195	0.01	0.0798	21.96	4.877	0.5273	0.6028	0.0279
	Ours	482.46	0.0491	26.18	6.032	0.8243	0.8480	0.0187	244.20	**0.0778**	**22.18**	**4.780**	0.5238	0.6063	0.0272
residential	**BCI**	**0.01**	0.0371	28.62	3.957	0.8883	0.9094	0.0096	**0.01**	0.0696	23.14	3.715	0.6571	0.7094	**0.0184**
	IBP	0.25	0.0331	29.59	3.540	0.9002	0.9244	0.0113	0.86	0.0659	23.63	3.517	0.6951	0.7359	0.0219
	GPP	42.02	0.0367	28.71	3.920	0.8830	0.9078	0.0115	29.62	0.0679	23.36	3.624	0.6682	0.7201	0.0192
	SRI	706.87	**0.0287**	**30.85**	**3.062**	0.9157	0.9394	0.0104	708.70	0.0722	22.83	3.856	0.6807	0.7375	0.0245
	LSE	1751.80	0.0428	27.37	4.558	0.8706	0.9021	0.0131	601.85	0.0848	21.43	4.429	0.6397	0.7035	0.0309
	GPR	390.19	0.0547	25.25	5.833	0.7838	0.8174	0.0135	148.06	0.0760	22.39	4.049	0.6011	0.6698	0.0203
	BDB	312.90	0.0835	21.57	8.916	0.6965	0.7466	0.0185	463.51	0.1453	16.76	7.749	0.3084	0.4072	0.0342
	DLU	0.19	0.0334	29.53	3.568	0.8987	0.9231	0.0114	0.19	0.0666	23.53	3.555	0.6733	0.7224	0.0193
	DRE	0.09	0.0334	29.53	3.568	0.8987	0.9230	0.0114	0.09	0.0666	23.53	3.555	0.6733	0.7224	0.0193
	FSR	1.26	0.0463	26.69	4.944	0.8448	0.8768	0.0126	3.88	0.0948	20.47	5.069	0.4598	0.5715	0.0234
	TSE	32.21	0.0292	30.68	3.124	**0.9242**	**0.9420**	**0.0096**	31.44	0.0602	24.40	3.216	**0.7402**	**0.7789**	0.0205
	UMK	0.01	0.0334	29.53	3.564	0.8989	0.9233	0.0115	0.01	0.0648	23.76	3.461	0.6919	0.7351	0.0204
	Ours	478.78	0.0290	30.74	3.114	0.9029	0.9340	0.0124	243.60	**0.0562**	**25.01**	**3.013**	0.7278	0.7787	0.0194
road	**BCI**	**0.01**	0.0535	25.78	6.043	0.7491	0.8939	0.0122	**0.01**	0.0975	20.57	5.505	0.4591	0.7277	0.0211
	IBP	0.12	0.0485	26.63	5.488	0.7676	0.9048	0.0145	0.35	0.0840	21.86	4.775	0.5206	0.7560	0.0250
	GPP	16.08	0.0530	25.86	5.993	0.7395	0.8910	0.0145	12.41	0.0938	20.90	5.295	0.4674	0.7390	0.0225
	SRI	200.28	0.0254	32.26	2.866	0.8018	0.9320	0.0124	206.48	0.0793	22.36	4.576	0.5428	0.7932	0.0217
	LSE	767.82	0.0501	26.36	5.638	0.7701	0.9140	0.0143	261.27	0.0860	21.66	4.800	0.4977	0.7865	0.0261
	GPR	194.94	0.0754	22.80	8.508	0.6341	0.8262	0.0152	85.12	0.1100	19.52	6.186	0.4090	0.6959	0.0230
	BDB	145.12	0.1276	18.23	14.506	0.5256	0.7416	0.0207	246.28	0.2005	14.31	11.425	0.2109	0.5307	0.0266
	DLU	0.06	0.0490	26.54	5.541	0.7637	0.9027	0.0146	0.07	0.0895	21.32	5.050	0.4774	0.7475	0.0228
	DRE	0.03	0.0490	26.54	5.541	0.7637	0.9027	0.0146	0.03	0.0895	21.32	5.050	0.4774	0.7475	0.0228
	FSR	0.39	0.0662	23.94	7.468	0.6991	0.8693	0.0145	1.88	0.1355	17.71	7.750	0.2758	0.6433	0.0232
	TSE	12.89	0.0365	29.11	4.117	**0.8204**	**0.9344**	0.0110	13.25	0.0668	23.86	3.769	**0.5787**	**0.8290**	0.0191
	UMK	0.01	0.0489	26.56	5.529	0.7641	0.9031	0.0147	0.01	0.0833	21.93	4.718	0.5064	0.7560	0.0236
	Ours	230.88	**0.0224**	**33.34**	**2.543**	0.7939	0.9298	**0.0108**	131.82	**0.0541**	**25.69**	**3.068**	0.5351	0.8247	**0.0186**