SUPER-RESOLUTION FOR SENTINEL-2 IMAGES

M. Galar[1,*], R. Sesma[2], C. Ayala[2], C. Aranda[2]

[1] Institute of Smart Cities (ISC), Public University of Navarre,
Campus de Arrosadía s/n, 31006, Pamplona, Spain - mikel.galar@unavarra.es
[2] Tracasa Instrumental, Calle Cabárceno, 6, 31621 Sarriguren, Navarra
- (rsesma, cayala, caranda)@itracasa.es

KEY WORDS: Super-resolution, deep learning, sentinel-2, image enhancement, convolutional neural network, optical images

ABSTRACT:

Obtaining Sentinel-2 imagery of higher spatial resolution than the native bands while ensuring that output imagery preserves the original radiometry has become a key issue since the deployment of Sentinel-2 satellites. Several studies have been carried out on the upsampling of 20m and 60m Sentinel-2 bands to 10 meters resolution taking advantage of 10m bands. However, how to super-resolve 10m bands to higher resolutions is still an open problem. Recently, deep learning-based techniques has become a de facto standard for single-image super-resolution. The problem is that neural network learning for super-resolution requires image pairs at both the original resolution (10m in Sentinel-2) and the target resolution (e.g., 5m or 2.5m). Since there is no way to obtain higher resolution images for Sentinel-2, we propose to consider images from others sensors having the greatest similarity in terms of spectral bands, which will be appropriately pre-processed. These images, together with Sentinel-2 images, will form our training set. We carry out several experiments using state-of-the-art Convolutional Neural Networks for single-image super-resolution showing that this methodology is a first step toward greater spatial resolution of Sentinel-2 images.

1. INTRODUCTION

The European Space Agency and its Copernicus mission are promoting research on earth observation via Sentinel missions. Sentinel-2 satellites capture multi-spectral images with 13 spectral bands every five days at the equator, allowing for monitoring the evolution of the earth surface. Their main usage is providing information for agriculture, forestry, food security and risk management among others (Drusch et al., 2012). The 13 spectral bands of Sentinel-2 capture images in the visible/near infrared (VNIR) and short wave infrared spectral range (SWIR) at different resolutions. However, only RGB and NIR bands are provided at the highest resolution of 10m, whereas the rest are given at either 20 or 60m.

Therefore, single image super-resolution (SISR) emerges as a possible way for improving these resolutions (Yang et al., 2018). Greater spatial resolution allows for a finer analysis and hence, more knowledge about the true condition of the earth. Previous works have been mainly focused on obtaining all 13 bands in 10m resolution using both the information of lower resolution bands and the existing 10m resolution bands (Lanaras et al., 2018, Gargiulo et al., 2018). However, these methods cannot be used for further increasing the resolution of RGB and NIR bands (e.g., 5m or 2.5m) as they require having bands at the target resolution.

Since 2012, deep learning has become the best tool for dealing with almost every problem related to computer vision and image processing (Goodfellow et al., 2016, Krizhevsky et al., 2012). Convolutional Neural Networks (CNNs) (Lecun et al., 1998) are usually considered to deal with images. Their most well-known applications are image classification (He et al., 2016), semantic segmentation (Ronneberger et al., 2015) or face recognition (Deng et al., 2018). However, their application has gone beyond standard problems and they are being actively used for remote sensing applications (Ball et al., 2017). Another scenario where CNNs have stood out is single image super-resolution (Yang et al., 2018). Several methods have been proposed in the literature for standard images using different architectures and learning methods, from standard CNNs (Kim et al., 2016) to Generative Adversarial Networks (GANs) (Ledig et al., 2017). These methods have clearly outperformed previous classical models ranging from bicubic interpolation to reconstruction methods (Yan et al., 2015). Hence, they seem to be excellent candidates for super-resolving Sentinel-2 RGB images to greater resolutions.

Nevertheless using these networks trained in standard images for super-resolution of satellite images has shown to provide poor results (Liebel, Körner, 2016) as they are not specifically trained for the characteristics of these kinds of images. Therefore, there is a need for creating specific CNNs for the problem at hand, the super-resolution of Sentinel-2 images. In this work, we will focus on the RGB bands, although the same methodology can be extended to NIR band. CNNs for super-resolution fall into the category of supervised machine learning. This means that the neural network is trained by giving the desired output for each input image. Hence, in super-resolution, low and high resolution image pairs are required. This is a challenging scenario as there are no higher than 10m resolution images available from Sentinel-2. Consequently, the main question is how to create these image pairs.

In this work we propose to consider satellite images from other sensors as a source for training neural networks for SISR of Sentinel-2 images. A similar approach has been considered in (Beaulieu et al., 2018). Nonetheless, few experiments are carried out and the usage of the specific sensors is not properly justified. We have reviewed existing sensors aiming at finding the most similar one to Sentinel-2 in terms of spectral bands (RGB and NIR). As we explain in Section 3, we found that im-

*Corresponding author

ages from RapidEye [1] are captured in almost the same spectral band, expecting that the obtained images will be the most similar to Sentinel-2, but having twice the resolution of these (5m). To increase the similarity, we tried to find images of the same date. However, even with this constraint and after the proper pre-processing, we found that there were some effects resulting in very dissimilar zones in the image pairs. For this reason, both a manual and automatic validation processes were required.

For the experimental study we have considered a state-of-the-art model called EDSR (Enhanced Deep Residual Networks) (Lim et al., 2017) with some modifications and several images from California area, as they are freely available from [2]. Several learning strategies have been tested and evaluated using the commonly considered metrics for super-resolution evaluation: the peak signal to noise ratio (PSNR) and the structural similarity (SSIM) (Zhou Wang et al., 2004). We will show that the proposed learning scheme leads to promising results but several challenges remain to be addressed.

The remainder of this paper is organized as follows. In Section 2, we briefly introduce deep learning and CNNs, mainly focusing on image super-resolution. Then, Section 3 presents our proposal for super-resolving Sentinel-2 images. The experiments are carried out in Section 4. Finally, our conclusions and future work are presented in Section 5.

2. PRELIMINARIES

Deep learning has supposed a major advance in artificial intelligence due to the excellent results obtained in various tasks such as computer vision, natural language processing, speech recognition or machine translation. More specifically, CNNs have super-passed previous computer vision and image processing methods for tasks such as classification, semantic segmentation, face recognition or image super-resolution. This is why we focus our attention on CNN-based SISR.

The first CNN was proposed by LeCun et al. (Lecun et al., 1998) for the classification of handwritten digit recognition with the well-known MNIST dataset. However, until AlexNet was proposed in 2012 (Krizhevsky et al., 2012), the power of CNNs was ignored. This model led to a 10% of increase in accuracy with respect to previous non-CNN-based models. Since then, this number has decreased to numbers even below human capability thanks to CNNs. Anyway, their capability goes beyond image classification problems and SISR is another field where they have stood out. We briefly recall several approaches for this purpose in Section 2.1 and focus on EDSR model in Section 2.2, which is the model considered for the experiments.

2.1 CNNs for Single Image Super-Resolution

The objective of SISR is to increase the spatial resolution of an image, considering only the information in the image itself and some acquired knowledge in the form of an algorithm or model (Yang et al., 2018). In the literature, three kinds of methods can be found for this purpose: interpolation-based, reconstruction-based and learning-based methods. Among interpolation based methods, bicubic interpolation (Keys, 1981) is the most well-known approach. Reconstruction-based methods (Yan et al., 2015) make use of some sophisticated prior knowledge to generate flexible and sharp details. Learning-based methods have been recently dominated by deep learning approaches with excellent results in standard images (Yang et al., 2018). This is why we focus on these types of methods.

For learning how to super-resolve an image using CNNs, one needs to give the CNN pairs of a low resolution image and a high resolution image. This way, the network is able to extract high-level abstractions from the low resolution image bridging the gap between the low resolution and high resolution spaces. Commonly, the pairs of images are obtained out of the same high resolution image by downsampling. This is the main problem we aim to face in this work, as there are no Sentinel-2 RGB images available at 5m resolution, our objective.

Once the training set is available, several different CNN architectures and optimization objectives can be considered (Yang et al., 2018). SRCNN (Dong et al., 2014) was the first architecture presented for SISR. It was based on firstly carrying out a bicubic interpolation and then going through a three-layer CNN. Both parts have been further improved with most recent approaches as bicubic interpolation resulted in a high computational cost and could produce wrong estimations in the final image. Likewise, deeper and more complex architectures can lead to better results as in other computer vision tasks. Regarding upsampling, Pixel Shuffle with sub-pixel convolution (Shi et al., 2016) was proposed as a part of ESPCN, improving both computational complexity and final performance. Moreover, ICNR initialization (Aitken et al., 2017) allowed to remove the checkerboard pattern present in several CNN-based approaches. With respect to the depth of the networks, VDSR (Kim et al., 2016) was the first deep network for super-resolution. It was composed of 20 layers based on the well-known VGG network (Simonyan, Zisserman, 2014) and started also from bicubic interpolation, although a residual connection was added aiming at improving performance and accelerate convergence. SRResNet (Ledig et al., 2017) is based on concatenating several ResBlocks commonly used for image classification. However, no major adaptations were made for the super-resolution problem, which can be suboptimal as argued by the authors of EDSR (Lim et al., 2017). This CNN is based on removing unnecessary modules from SRResNet and using the proper loss function to achieve the best performance in the problem at hand. We briefly detail the properties of this network in the next section as it is the base for our proposal.

Apart from the architecture, the loss function considered for learning the parameters of the CNN is another key factor. The L2 norm, i.e., Mean Square Error (MSE), has been the most widely used loss function. Nevertheless, in EDSR the authors used the L1 norm, i.e, Mean Absolute Error (MAE), claiming that it resulted in better convergence. The usage of GANs can also be seen as a different form of training. This is the case of SRGAN (Ledig et al., 2017), which trains a SRResNet using GAN learning. That is, a discriminator network is used for learning whether the produced image is the real high resolution image or the super-resolved one, and its loss function is combined with the L2 norm. This kind of learning tend to lead to good visual results, but this is not usually reflected in the performance measures due to their capability to picture missing pixels.

Notice that most of the proposed networks focus on 4x scale, where difference between bicubic interpolation and CNN-based approaches becomes higher. In this work, we focus on a first

[1] https://directory.eoportal.org/web/eoportal/satellite-missions/r/rapideye

[2] https://www.planet.com/trial/

step for Sentinel-2 super-resolution aiming at doubling the resolution of the original images (2x).

2.2 EDRS: Enhanced Deep Residual Networks

EDSR has several properties that makes it different from previous approaches. It is based on SRResNet, but with the proper modifications according to the properties of SISR. Bearing this in mind, the authors proposed to remove batch normalization from ResBlocks. Using these blocks is interesting as they allow the information to flow through the network without modifications, since the output should be similar to the input. However, the key point here was that removing batch normalization the information suffered less changes, which was desired in this case due to the aforementioned reason (different from image classification problem). Moreover, a residual scaling factor (Lim et al., 2017) was introduced into the network to stabilize learning (default value of 0.1).

Two main parameters are required to define the architecture of EDSR: the number of ResBlocks and the number of filters. In this work we consider the simplest version of EDSR with 8 ResBlocks and 64 filters. After the ResBlocks, a Pixel Shuffle upsampling is used to finally increase the resolution of the image. This is done at the end, which makes EDSR faster than other alternatives where upsampling is performed just before going through the network, which results in all operations being performed over the higher resolution image. Interestingly, this operation procedure allows EDSR to take advantage of lower scales super-resolutions for higher scales ones. This is known as progressive resizing. This means that the EDRS learned for 2x super-resolution can be used as a pre-trained model for 4x super-resolution. The only required modification is to add a new Pixel Shuffle upsampling at the end of the network. This allows one to make convergence for higher resolutions faster. Likewise, we will use a similar strategy although we will always work with 2x scale. A scheme of the EDSR used in this work is presented in Figure 1.

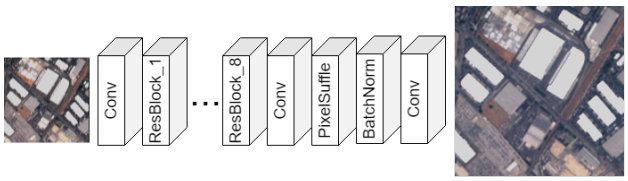

Figure 1. Architecture of EDSR.

With respect to the loss function, the authors proposed to use L1 norm because they found that its convergence was faster than using L2 norm. In testing phase, they included a self-ensemble model (named as EDSR+ in the original paper), where the same image was augmented for testing with flips and 90 rotations up to 7 augmented inputs plus the original image. Then the 8 images are passed to the network and the predictions are averaged (obviously, after undoing transformations).

3. SENTINEL-2 TO RAPIDEYE: SUPER-RESOLUTION OF SENTINEL-2 IMAGES

In this section we present our proposal for the super-resolution of Sentinel-2 images, which consists in learning a EDSR model using images from a different sensor as target images. The proposal including the explanation of why we consider RapidEye satellite is presented in 3.1. Then, Section 3.2 details the data we have used for training and Section 3.3 explains the different settings we have considered for the network training.

3.1 Proposal

The main problem we need to address in order to apply EDSR to Sentinel-2 images is that we do not have any high resolution image at 5m. Therefore, we tried to find the sensor with the most similar spectral bands to those of Sentinel-2 but providing us with higher resolution images at 5m. We found that the satellite satisfying these properties was RapidEye [3] operating since 2009. In Figure 2, we can observe that the spectral bands for both satellites are similar and hence, we consider RapidEye images as excellent candidates for our purpose.

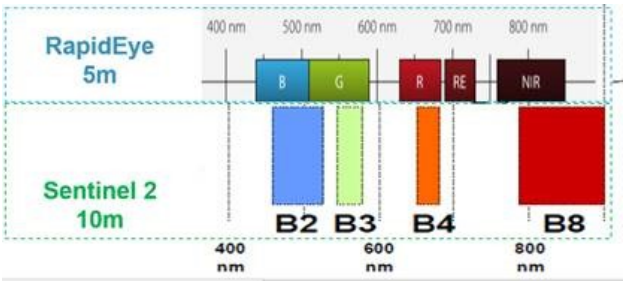

Figure 2. Comparison between RapidEye and Sentinel-2 spectral bands for RGB and NIR.

Obviously, the raw products of RapidEye and Sentinel-2 have different magnitudes. In Sentinel-2, one can select the the processing level (e.g., L1C is top of atmosphere reflectance and L2A is bottom of atmosphere reflectance). Otherwise, in RapidEye digital numbers are provided, which need to be converted into the appropriate magnitude (top of atmosphere reflectance in our case). These aspects are detailed in the next section. Notice that we should restrict ourselves to work with top of atmosphere reflectance as it is the level of processing in which RapidEye images are provided.

3.2 Datasets

Once the most suitable sensor for super-resolving Sentinel-2 images has been decided, we need to download image pairs. Since images are captured by different satellites, it may be difficult to find image pairs that are temporarily close to each other so that we may find the minimum number of changes between them. For this reason, we try to make them match the date as much as possible and we also only consider images with cloud cover less than 10%.

RapidEye images were downloaded using 14 days free trial from Planet [4], which gives access to open California. Notice that we aim to extend this study in the future with strategically selected images from different areas of interest. Hence, all the images considered in this study are from California state (United States of America, USA). To make our network more robust against the super-resolution of urban areas, we focused on the main urban cities of California: Los Angeles, Beberlyhills, Calabasas, San Jose, Hayward and Yuba.

In first place, we downloaded images from RapidEye satisfying the cloud cover restriction (Analytic Ortho Tile products are

[3] https://directory.eoportal.org/web/eoportal/satellite-missions/r/rapideye

[4] https://www.planet.com/

used). Then, we found the images from Sentinel-2 satisfying the same restriction and we took the ones minimizing the difference with respect to the acquisition date of RapidEye ones (Sentinel-2 L1C products are used). For the future, we aim to optimize this process by first matching image pairs with minimum difference between dates (with less than 10% cloud cover in both).

After downloading the images, we need to normalize them so that they are almost the same but with different resolutions. To do so, we carried out the following process:

1. Convert Sentinel-2 products to GeoTIFF raster (RGB).
2. Convert RapidEye RGB data to Top of Atmosphere Reflectance.
3. Match RapidEye tiles in Sentinel-2 image and crop accordingly.
4. Outlier values in RGB images are detected with the percentiles 1 and 99 of each band in both products. These values are changed to take the minimum or the maximum possible value, respectively.
5. With the maximum and minimum values after outlier removal, MaxMin normalization is performed to create the RGB images in uint8.

At this point, we have both images at the same scale and range, almost ready to be used for training the network. However, by visual inspection we found that there were some places where both images were highly different. Since such a large difference could hinder the learning of our network, we carried out both a manual and automatic validation processes by patches. Hence, we split the images in patches of 96x96 pixels in Sentinel-2 resolution (a typical value used in SISR with CNNs) and revised all patches one by one looking at major differences, which were marked for removal. Afterwards, a statistical validation based on both the mean and standard deviation of the pixel intensity values was carried out (we acknowledge that doing it the other way would have been more efficient). We computed the mean and standard deviation of each band and satellite for all the patches of 96x96 pixels and plotted the ratio between Sentinel-2 and RapidEye patches in histograms. By visual inspections different threshold values were selected for the mean ($0.94 < \mu < 1.15$) and standard deviation ($0.77 < \sigma < 1.25$). A patch pair was removed if any of those inequalities was not satisfied for at least one band. An example of the differences between images from both satellites and the corresponding mask with validated/removed patches is shown in Figure 3. After validation, 3931 patches were considered for the experiments out of 6048, i.e., 35% of the patches were removed either by manual (1711) or automatic (442) validation.

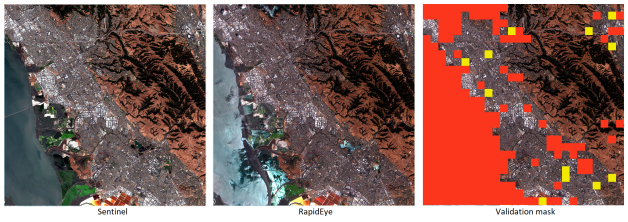

Figure 3. Example of validated/removed patches in the city Hayward (in red patches removed by manual validation, in yellow those removed by the statistical validation).

The whole set of images considered for our study is summarized in Figure 4 and Table 1, where for each image pairs, the main city covering the images, the dates of the Sentinel-2 and RapidEye images, the delay between both images (in days), the set to which we have assigned the images and the final number of generated patches from the image pairs are presented. Moreover, Table 2 shows the final number of patches considered for each set (training, validation and test). Recall that this data partitioning is the one usually considered for training and evaluating machine learning models. Most of the data is used for training (approximately 75% of the patches). Few data is considered for validation (7.5%), which in our case serves for deciding when the network is saved during training and the rest is used for testing, that is, to obtain the final result of each configuration over a set of patches that were not used for fitting the model.

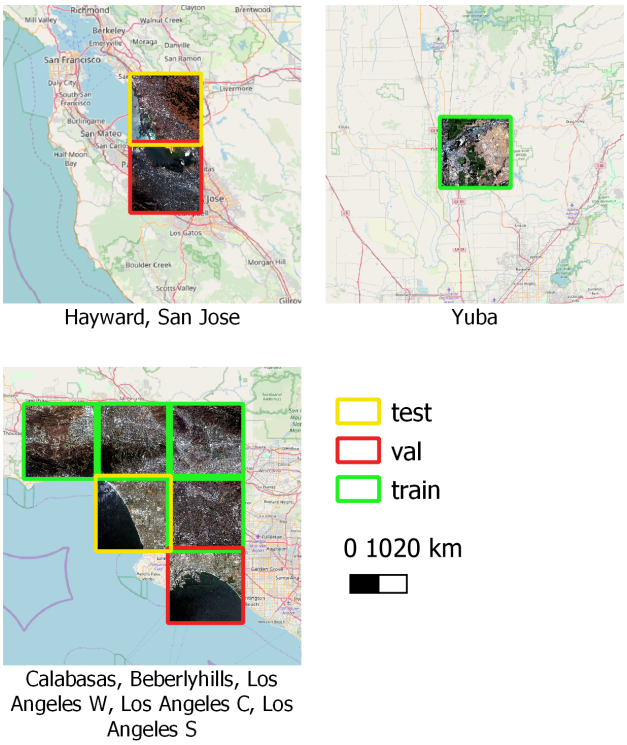

Figure 4. Location of images considered for the study.

City	Sentinel date	RapidEye date	Delay (d)	Set	#Patches
Yuba	2018-08-28	2018-08-29	1	Train	616
Calabasas	2018-07-23	2018-07-02	21	Train	581
Beberlyhills	2018-07-23	2018-06-13	40	Train	584
Los Angeles N	2018-07-23	2018-08-05	13	Train	611
Los Angeles C	2018-07-23	2018-08-05	13	Train	526
San Jose	2018-07-09	2018-08-20	42	Val	72
Los Angeles S	2018-07-23	2018-08-05	13	Val	223
Hayward	2018-07-09	2018-07-04	5	Test	414
Los Angeles W	2018-07-23	2018-08-05	10	Test	304

Table 1. Summary of the images used from Sentinel-2 and RapidEye to form our dataset.

3.3 Network training

Regarding our implementation of EDRS, we have used fast.ai library (Howard et al., 2018) with PyTorch (Paszke et al., 2017) and followed several guidelines from [5]. In order to avoid checkerboard pattern produced by Pixel Shuffle we initialize this layer using ICNR (Aitken et al., 2017). The other major change is

[5] https://course.fast.ai/

Set	Images	#Patches	Ratio %
Train	5	2918	74.2
Val	2	295	7.5
Test	2	718	18.3
Total	9	3931	100

Table 2. Summary of the images used from Sentinel-2 and RapidEye to form our dataset.

that we have considered a more advanced loss function that suits better our specific scenario where both images comes from different sources. In this case, although we have established a proper validation so that images from different sensors are almost the same, only relying on L1 or L2 norms led to blurry results. Hence, we modified this behavior by adding both a feature loss based on VGG16 (Simonyan, Zisserman, 2014) and a style loss (Johnson et al., 2016) based on the same network. The former computes the L1 loss between the activations of different layers of the VGG16 when both the target image and the super-resolved one are forwarded through the network. The latter is commonly used in style transfer and tries to force the super-resolved image to have similar correlations to those of the target one among the activations of the different channels in several layers of VGG16. Using these losses together with the L1 norm (pixel loss) allows us to make the network focus not only on individual pixel differences but also on the overall look of the resulting image.

Following (Lim et al., 2017), we used a batch size of 16. We followed the guidelines of (Smith, 2018) for training the network using one-cycle policy. That is, when training a network from scratch we first looked for the most appropriate learning rate for each run using the learning rate finder. Then, we run 50 epoch with this maximum learning rate. Again, learning rate finder is used for finding the best learning rate for another 100 epoch run. As we will explain afterwards we take advantage of transfer learning by progressive resizing. In this cases, we start from an already trained model and hence, we run learning rate finder and use this learning rate for further training 50 epochs. Afterwards, the learning rate is divided by 10 and 30 more epoch are run. This process is repeated twice. The learning rates used for each configuration and the learning scheme used are detailed in Table 3. The different configurations are explained hereafter.

With the idea of progressive resizing of EDSR, where first the model for 2x is trained and then, the model for 4x is trained from the previous one simply adding another upsampling layer, we have thought of different strategies for learning using Sentinel-2 and RapidEye images. However, in this case we always consider the same scale factor (2x), but change the resolution of the images from which we train the network. This is known to accelerate convergence and improve generalization. Bearing this idea in mind, we tested different learning strategies. To do so, we consider image patches at different resolutions: low resolution (20m), medium resolution (10m) and high resolution (5m). Notice that with this nomenclature, the original Sentinel-2 (SE) images are in medium resolution and RapidEye (RE) ones in high resolution. Our final objective is to perform that translation from 10m to 5m (SE2RE). However, we can first pre-train the network to move from 20m to 10m and use that network to faster and better train the subsequent model super-resolving from 10m to 5m. Moreover, we can perform these pre-trainings either with Sentinel-2 images or RapidEye ones (as we can downsample both of them to 10m and 20m, respectively). Table 3 summarizes all the configurations tested. Mainly, we can differentiate four main ideas:

1. Pre-train with only RapidEye images (RE2RE) and finally fine-tune with Sentinel-2 to RapidEye (SE2RE). Here, we have two main possibilities, to pre-train first from 20m to 10m (1.1 model) and then from 10m to 5m using RapidEye (1.2 model) and then move to Sentinel-2 to RapidEye (10m to 5m again but with different images, 1.4 model); or to pre-train with RapidEye only for 20m to 10m and then move to Sentinel-2 to RapidEye (10m to 5m, 1.3 model).
2. Pre-train with Sentinel-2 as long as possible (20m to 10m, 2.1 model), and then continue with Sentinel-2 to RapidEye (10m to 5m, 2.2 model).
3. Always maintain the idea of training from Sentinel-2 to RapidEye for pre-training (20m to 10m, 3.1 model) and fine-tuning (10m to 5m, 3.2 model).
4. Do not carry out pre-traing and directly train the network from scratch for super-resolving 10m to 5m (Sentinel-2 to RapidEye, 4.1 model).

Configuration	Learning scheme
1.1. $RE^{20} \rightarrow RE^{10}$	50 ep, lr 5e-3 → 100 ep, lr 1e-03
1.2. $RE^{10} \rightarrow RE^{5}$ (from 1.1)	50 ep, lr 1e-4 → 100 ep, lr 1e-3
1.3. $SE^{10} \rightarrow RE^{5}$ (from 1.1)	50 ep, lr 1e-4 → 30 ep 1e-5 → 30 ep, lr 1e-6
1.4. $SE^{10} \rightarrow RE^{5}$ (from 1.2)	50 ep 1e-3 → 30 ep, lr 1-4 → 30 ep, lr 1e-5
2.1. $SE^{20} \rightarrow SE^{10}$	50 ep, lr 5.25e-3 → 100 ep, lr 5.25e-3
2.2. $SE^{10} \rightarrow RE^{5}$ (from 2.1)	50 ep, lr 2.75e-4 → 30 ep, lr 2.75e-5 → 30 ep 2.75e-6
3.1. $SE^{20} \rightarrow RE^{10}$	50 ep, lr 4.37e-3 → 100 ep, lr 1e-3
3.2. $SE^{10} \rightarrow RE^{5}$ (from 3.1)	50 ep, lr 1e-4 → 30 ep 1e-5 → 30 ep, lr 1e-6
4.1. $SE^{10} \rightarrow RE^{5}$	50 ep, lr 6.31e-3 → 100 ep, lr 1e-3 → → 30 ep, lr 1e-4 → 30 ep, lr 1e-5

*ep: epoch; lr: learning rate; RE: RapidEye; SE: Sentinel-2

Table 3. Configurations considered in the experiments.

The rest of the parameters for training the network are presented in Table 4. We consider bicubic interpolation for comparison with the proposed super-resolution. In the future, we aim to extend this comparison with more complex methods.

Param. name	Value
Batch size	16
VGG16 layers (for feature/style losses)	First 3 Max-pooling inputs
VGG16 layer weights (feature loss)	(0.2, 0.7, 0.1)
VGG16 layer weights (style loss)	(200, 2450, 50)
Losses weighting (pixel, feature, style losses)	(1.0, 1.0, 1.0)
Optimizer	Adam
Learning strategy	Once Cycle Policy (pct_start=0.7)
Weight decay	1e-7

Table 4. Common parameters for all configurations.

3.4 Evaluation measures

For the evaluation of the results obtained, we have considered the two most widely used metrics for super-resolution evaluation: the peak signal-to-noise ratio (PSNR) and the structural similarity (SSIM) index (Zhou Wang et al., 2004).

The PSNR measure the image restoration quality comparing the obtained super-resolved image (from SE) with the ground

truth (from RE). Notice that it is tightly related to the mean squared error and hence, measures the differences between images pixel-wise:

$$\text{PSNR}(y, \hat{y}) = 10 \cdot \log_{10}(\frac{v_{max}^2}{\text{MSE}}) \quad (1)$$

where v_{max}^2 is the greatest possible difference between two pixel values and

$$\text{MSE}(y, \hat{y}) = \frac{1}{N \cdot M \cdot C} \sum_{i=1}^{N} \sum_{j=1}^{M} \sum_{k=1}^{C} (y_{ijk} - \hat{y}_{ijk})^2 \quad (2)$$

where N, M, C are the number of rows, columns and channels of the image, respectively.

Different from the PSNR, the SSIM is designed to be consistent with human perception. Hence, it may even be more important than PSNR for certain scenarios such as ours. Notice that we do not have real ground truth of Sentinel-2 at 5m, but approximate ones from RapidEye. Hence, perception may capture better the quality of the prediction rather than the pixel-wise difference.

4. EXPERIMENTAL STUDY

In this section we present and discuss the results of our proposal.

4.1 Results

Table 5 presents the results in terms of PSNR and SSMI performance measures. Observe that we show the results for both the super-resolution of RapidEye (10m) to RapidEye (5m) (RE2RE) and Sentinel-2 (10m) to RapidEye (5m) (SE2RE). The former allows us to check whether pre-training in RapidEye is working properly, whereas the latter is the main focus of this work, the results of Sentinel-2 super-resolution. Notice that we do not expect to achieve the same results with Sentinel-2 super-resolution as those we can achieve with RapidEye 10m to 5m, as we are dealing with images coming from different sensors and hence, the objective is not to transfer one image to the other but simply to super-resolve the first one. Anyway, this numbers gives us an intuition of how well super-resolution is performing.

Additionally, in Figure 5 we also provide several examples of super-resolved patches so that visual comparison between bicubic interpolation and the proposed method can be performed. The configuration selected is the one with the best performance metrics (model 1.4).

4.2 Discussion

We will first analyze the results in Table 5, discussing the effects of the different configurations. Then, we will comment on the visual results in Figure 5.

First, we can observe that bicubic interpolation can be outperformed by our EDSR-based solution in both scenarios (RE2RE and SE2RE). The results of model 1.2 (pre-trained in 1.1) in RE2RE are impressive, achieving 35 dB in PSNR and a SSIM of almost 0.96. Obviously, these results do not transfer well when we evaluate the model in SE2RE task. Performance is decreased due to two main reasons: 1) the model is not trained with Sentinel-2 images; 2) The task is much harder as super-resolved images are evaluated with images coming from a different source than the input image. Hence, it is clear that we need to train the model with Sentinel-2 images as input and RapidEye ones as outputs if we want to perform well in SE2RE.

	$RE^{10} \rightarrow RE^5$		$SE^{10} \rightarrow RE^5$	
Configuration	PSNR	SSIM	PSNR	SSIM
0.0. Bicubic	31.68	0.9094	26.80	0.8055
1.1. $RE^{20} \rightarrow RE^{10}$	33.88	0.9389	26.38	0.7989
1.2. $RE^{10} \rightarrow RE^5$ (from 1.1)	**35.49**	**0.9572**	26.95	0.8137
1.3. $SE^{10} \rightarrow RE^5$ (from 1.1)	32.61	0.9383	27.63	0.8220
1.4. $SE^{10} \rightarrow RE^5$ (from 1.2)	32.14	0.9395	**27.81**	**0.8285**
2.1. $SE^{20} \rightarrow SE^{10}$	33.76	0.9392	26.48	0.7979
2.2. $SE^{10} \rightarrow RE^5$ (from 2.1)	32.64	0.9404	27.62	0.8220
3.1. $SE^{20} \rightarrow RE^{10}$	31.23	0.9189	26.87	0.7983
3.2. $SE^{10} \rightarrow RE^5$ (from 3.1)	32.07	0.9288	27.43	0.8178
4.1. $SE^{10} \rightarrow RE^5$	32.18	0.9355	27.75	0.8253

Table 5. Results obtained by the different configurations in test set for both PSNR and SSIM.

The most straightforward way to do so is to follow model 4.1, where no pre-training is carried out. However, attending to the results, we can observe that this solution is suboptimal. The model with the best performance is 1.4, which is trained after model 1.2 is obtained, which at the same time starts from the model obtained in 1.1. This allows us to achieve a more accurate model than directly addressing the super-resolution of Sentinel-2 to RapidEye. Notice however that not all pre-trainings are performing equally, since the rest of the models are not able to improve the results of 4.1 (no pre-training).

Focusing on the visual results in Figure 5, we should highlight the difference between bicubic and the proposed approach. Images are less blurry and more sharpen. Edges are better defined. In general, it could be difficult to differentiate between the RapidEye and the super-resolved one. However, looking at the last image, one can observe that with 10m resolutions there are details that can be hardly recovered, such as one of the roads in the lower left part of the image. This can be observed in RapidEye, but there is no way to see that horizontal road in Sentinel-2 and hence, our proposal cannot imagine it. We should finally acknowledge the fact that the differences in numbers in Table 5, are more clearly observed when looking at the output images of the network, which is because the evaluation is being performed with respect to RapidEye instead of a Sentinel-2 image at 5m resolution, which does not exist.

5. CONCLUSIONS AND FUTURE WORK

In this work we have proposed a novel way for super-resolving Sentinel-2 RGB bands to 5m resolution. To do so, we consider using images from another satellite with similar spectral bands but capturing images at a higher resolution to learn a deep learning model. The selected satellite is RapidEye. With pairs of images of both satellites we have been able to train a network based on EDSR with some changes such as a loss function considering feature and style losses, a proper initialization of Pixel

Figure 5. Visual comparison between the bicubic interpolation and the proposed method.

Shuffle layer and using one-cycle learning policy. Moreover, we considered different strategies for learning based on progressive resizing idea. We should highlight the results, evaluated in terms of PSNR and SSIM, obtained by our proposed model which is based on previous training phases on RapidEye images (20m to 10m and 10m to 5m). Visual results showed that the images obtained avoid the blurry effect of bicubic interpolation. Nonetheless, there are still several future works that should be considered. Regarding the dataset used, we want to include more images for training, validation and testing. In fact, we do not only want more images but also images which are better co-registered, that is, whose capture days differ as less as possible. We will also extend the work to other zones different from California making use of subscription-based images of RapidEye.

With respect to the CNNs, we would like to carry out a proper comparison among the state-of-the-art models, including GAN-based approaches. In this way, we want to extend the experimental comparison including other methods for super-resolution not based on neural networks. Finally, it will be interesting to think of increasing the resolution of Sentinel-2 images further, e.g., considering 4x scale.

REFERENCES

Aitken, A., Ledig, C., Theis, L., Caballero, J., Wang, Z., Shi, W., 2017. Checkerboard artifact free sub-pixel convolution: A note on sub-pixel convolution, resize convolution and convolution resize. *arXiv*.

Ball, J., Anderson, D., Chan, C. S., 2017. A Comprehensive Survey of Deep Learning in Remote Sensing: Theories, Tools and Challenges for the Community. *Journal of Applied Remote Sensing*.

Beaulieu, M., Foucher, S., Haberman, D., Stewart, C., 2018. Deep image-to-image transfer applied to resolution enhancement of sentinel-2 images. *International Geoscience and Remote Sensing Symposium (IGARSS)*, 2018-July, 2611–2614.

Deng, J., Guo, J., Zafeiriou, S., 2018. ArcFace: Additive Angular Margin Loss for Deep Face Recognition. *arXiv*.

Dong, C., Loy, C. C., He, K., Tang, X., 2014. Learning a deep convolutional network for image super-resolution. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*.

Drusch, M., Del Bello, U., Carlier, S., Colin, O., Fernandez, V., Gascon, F., Hoersch, B., Isola, C., Laberinti, P., Martimort, P., Meygret, A., Spoto, F., Sy, O., Marchese, F., Bargellini, P., 2012. Sentinel-2: ESA's Optical High-Resolution Mission for GMES Operational Services. *Remote Sensing of Environment*.

Gargiulo, M., Mazza, A., Gaetano, R., Ruello, G., Scarpa, G., 2018. A CNN-Based Fusion Method for Super-Resolution of Sentinel-2 Data. *IGARSS 2018 - 2018 IEEE International Geoscience and Remote Sensing Symposium*, 4713–4716.

Goodfellow, I., Bengio, Y., Courville, A., 2016. *Deep Learning*. MIT Press. `http://www.deeplearningbook.org`.

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*.

Howard, J. et al., 2018. fastai. `https://github.com/fastai/fastai`.

Johnson, J., Alahi, A., Fei-Fei, L., 2016. Perceptual losses for real-time style transfer and super-resolution. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*.

Keys, R. G., 1981. Cubic Convolution Interpolation for Digital Image Processing. *IEEE Transactions on Acoustics, Speech, and Signal Processing*.

Kim, J., Lee, J. K., Lee, K. M., 2016. Accurate image super-resolution using very deep convolutional networks. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*.

Krizhevsky, A., Sutskever, I., Hinton, G. E., 2012. Imagenet classification with deep convolutional neural networks. *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, 1097–1105.

Lanaras, C., Bioucas-Dias, J., Galliani, S., Baltsavias, E., Schindler, K., 2018. Super-resolution of Sentinel-2 images: Learning a globally applicable deep neural network. *ISPRS Journal of Photogrammetry and Remote Sensing*.

Lecun, Y., Bottou, L., Bengio, Y., Ha, P., 1998. LeNet. *Proceedings of the IEEE*.

Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z., Shi, W., 2017. Photo-realistic single image super-resolution using a generative adversarial network. *Procs. 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*.

Liebel, L., Körner, M., 2016. Single-image super resolution for multispectral remote sensing data using convolutional neural networks. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*, 41(July), 883–890.

Lim, B., Son, S., Kim, H., Nah, S., Lee, K. M., 2017. Enhanced Deep Residual Networks for Single Image Super-Resolution. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2017-July, 1132–1140.

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., Lerer, A., 2017. Automatic differentiation in PyTorch. *NIPS Autodiff Workshop*.

Ronneberger, O., Fischer, P., Brox, T., 2015. U-net: Convolutional networks for biomedical image segmentation. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*.

Shi, W., Caballero, J., Huszar, F., Totz, J., Aitken, A. P., Bishop, R., Rueckert, D., Wang, Z., 2016. Real-Time Single Image and Video Super-Resolution Using an Efficient Sub-Pixel Convolutional Neural Network. *Procs. of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*.

Simonyan, K., Zisserman, A., 2014. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv 1409.1556*.

Smith, L., 2018. A disciplined approach to neural network hyper-parameters: Part 1 – learning rate, batch size, momentum, and weight decay. *arXiv*.

Yan, Q., Xu, Y., Yang, X., Nguyen, T. Q., 2015. Single image superresolution based on gradient profile sharpness. *IEEE Transactions on Image Processing*.

Yang, W., Zhang, X., Tian, Y., Wang, W., Xue, J.-H., 2018. Deep Learning for Single Image Super-Resolution: A Brief Review. *arXiv*, 1–17.

Zhou W., Bovik, A. C., Sheikh, H. R., Simoncelli, E. P., 2004. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4), 600-612.