Comparative study of upsampling methods for super-resolution in remote sensing

Luis Salgueiro Romero[a], Javier Marcello[b], and Verónica Vilaplana[c]

[a,c]Image Processing Group, Universitat Politècnica de Catalunya, Barcelona, Catalonia/Spain
[b]Instituto de Oceanografia y Cambio Global, Universidad de Las Palmas de Gran Canaria, Unidad Asociada ULPGC-CSIC, Las Palmas de Gran Canaria, Spain

ABSTRACT

Many remote sensing applications require high spatial resolution images, but the elevated cost of these images makes some studies unfeasible. Single-image super-resolution algorithms can improve the spatial resolution of a low-resolution image by recovering feature details learned from pairs of low-high resolution images. In this work, several configurations of ESRGAN, a state-of-the-art algorithm for image super-resolution are tested. We make a comparison between several scenarios, with different modes of upsampling and channels involved. The best results are obtained training a model with RGB-IR channels and using progressive upsampling.

Keywords: Super-resolution, Deep Learning, Remote Sensing, WorldView-2

1. INTRODUCTION

Today several satellite platforms for remote sensing are available. However, due to limitations in imaging sensors, acquired images usually have limited spatial resolution. For instance, a Landsat-8 satellite produces images with a 30m of Ground Sampling Distance (GSD), which is the minimum surface cover of the Earth projected by a pixel. This resolution is good enough for large scale studies but it can not be used for detecting objects with sizes smaller than this resolution. On the other hand, WorldView-2 (WV2) is a commercial very high-resolution satellite that provides images with a GSD around 2m. Nevertheless, these images might not always be available and in some cases may be difficult to obtain, making a challenge to tackle small scale studies. Reconstruction of a high resolution image from a low resolution image of the same scene, acquired from different views, different sensors or at different conditions is therefore a very interesting and challenging problem.

Single image super-resolution (SISR) aims to estimate a High Resolution (HR) version image from a Low Resolution (LR) one. It is an ill-posed problem because multiple HR versions can be obtained from a particular LR image [1]. Simple super-resolution methods like linear or bicubic interpolation are very fast and do not require training but tend to over-smooth image textures [2]. Recently, deep learning convolutional neural networks (CNN) have been shown to outperform traditional methods on natural images. CNNs try to establish an end-to-end mapping between low and high-resolution images, but they may fail to produce good results for high scaling factor values [2].

There have been several improvements since the work presented in [1], where a shallow network with four layers was trained. The concept of residual learning [3], that learns differences between the LR-HR resolution instead of learning a sophisticated mapping to produce the fine resolution image directly, has led to significant improvements. SRGAN [2] introduces the idea of adversarial learning, producing a more photo-realistic result on the upscaled LR image. ESRGAN [4] follows the same approach, with an improved architecture that uses a more complex and dense combination of residual layers.

Several works address the problem of enhancing the spatial resolution of satellite images due to the increasing availability of data collections. In [5], a shallow and a deep neural network are trained to improve the resolution of the Advanced Wide Field Sensor with 54m GSD, to resemble the resolution of the Linear Imaging Self Scanner satellite with 24m GSD. In [6], CNN architectures are trained to enhance Landsat images (30m GSD) using Sentinel-2 images (10m GSD).

The goal of this work is to compare several configurations of ESRGAN, a state-of-the-art algorithm for image super resolution, on remote sensing data. For testing the models, we work with original and low-resolution versions of the WorldView-2 Euopean Cities dataset [7].

Salgueiro, L.; Marcello, J.; Vilaplana, V. Comparative study of upsampling methods for super-resolution in remote sensing. A: International Conference on Machine Vision. "Twelfth International Conference on Machine Vision (ICMV 2019): 16–18 November 2019, Amsterdam, The Netherlands". Washington: International Society for Photo-Optical Instrumentation Engineers (SPIE), 2019, ISBN 9781510636446.

http://dx.doi.org/10.1117/12.2557357

2. DATA

The data used in this study is the WorldView-2 European Cities [7] dataset. This collection of WorldView-2 (WV2) images is provided by the European Space Agency. Images have been acquired between July 2010 and July 2015, and cover the most populated areas in Europe.

The dataset is available with Level-3 processing, i.e. a digital elevation model was used to accurately georeference pixels [8]. Images are available in 11-bit digital value format, and for the correction process, the ENVI software [9] was used. First, radiometric calibration was applied to convert digital values to radiance values, then the FLAASH algorithm [10] was used for the atmospheric correction. This algorithm requires the setting of several parameters: atmospheric model, aerosol model, azimuth and zenith angles. For simplicity, none aerosol model was used and the zenith and azimuth angles were determined using the configuration files that came with the images. The atmospheric model was established depending on the coordinates and the acquisition season.

A total of 32 WV2 images were used, with sizes around 10Kx10K pixels. Also, 4 of the 8 channels were considered, specifically RGB and IR channels, commonly used in remote sensing applications. The spatial resolution is 1.6m/pixel. For testing the super-resolution algorithms, low-resolution images were generated using a Gaussian filter for anti-aliasing and downsampling by a factor of 4.

We split the dataset into training (90%), validation (5%) and test (5%) subsets. Each subset contains corresponding pairs of LR and HR tiles (1Kx1K pixels in the LR image and a 4Kx4K in the HR image).

3. METHODOLOGY

3.1 Super-resolution network

SRGAN was one of the first models capable of generating photo-realistic images with a scaling factor of 4, recovering high detail texture from a downsampled image [2]. The generator network, called SRResNet, was trained with the mean squared error (MSE) loss and introduced the use of several residual blocks, as depicted in Figure 1. They used two upsampling modules, each module upscaling the image by a factor of 2, making the upsampling in a progressive way and leaving the model to learn the right weights to upscale the LR images.

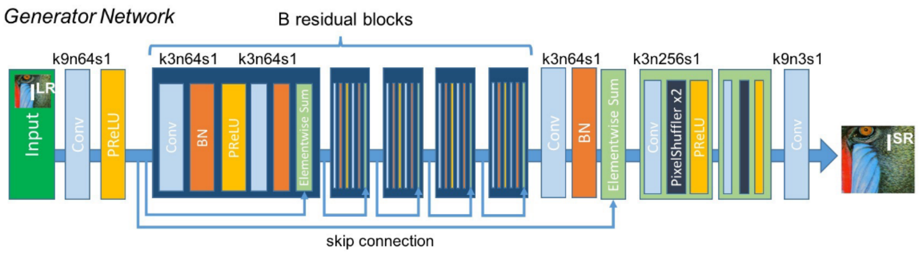


Figure 1. SRResNet architecture. Source [2]

The SRGAN residual block (see Figure 2) is composed of a concatenation of convolution, batch normalization (BN) and Leaky-ReLU activation layers. The output of each block is added to its input through a skip connection. Blocks are concatenated and the output of the residual blocks is combined with the input by an element-wise sum operation, before the upsampling modules.

ESRGAN proposed some improvements over the SRGAN. Since batch normalization layers tend to produce artifacts in the SR images, each residual block was replaced with a more efficient block, composed of just a combination of convolutional and activation layers, Also, a more complex residual configuration, called Residual in Residual Dense Block (RRDB) (see Figure 3) was used for multi-level residual learning. With these changes, more residual blocks were added and boosted the performance. The generator model, called the PSNR-Oriented model, was trained with L1-loss, and later the model was fine-tuned following an adversarial approach. ESRGAN is currently considered the state-of-the-art algorithm for super-resolution with a scale factor of 4 [11].

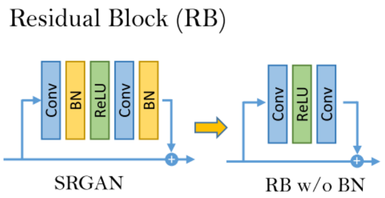


Figure 2. Residual blocks. Source: [4]

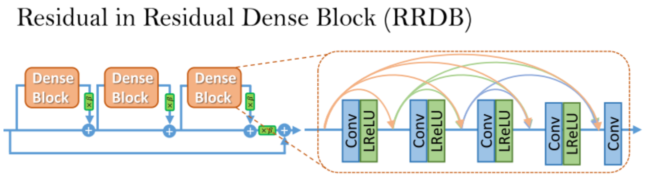


Figure 3. Residual in Residual Dense Block (RRDB). Source [4]

In this work, a comparison between the learning-based upsampling of the ERSGAN PSRN-oriented model and a pre-upsampling method for super-resolution was performed, where the upsample modules were removed for the pre-upsampled method. Also, fine-tuning was done using the weights of a pretrained PSNR-Oriented model provided by the authors of ESRGAN [12].

3.2 Experiments

First, experiment **T1** used the raw-weights of the PSNR-oriented model to test the efficiency of using cross-domain trained models. This model was trained on the DIV-2K dataset (natural RGB images).

Then, in experiment **T2**, we fine-tuned the PSNR-Oriented model with the WV2 dataset, training all the layers with three channels (RGB). In experiments **T1** and **T2** the output images are obtained in two steps. We first super-resolve RGB channels, then RG-IR channels and we finally concatenate the results to produce the RGB-IR image.

Experiment **T3** is similar to **T2**, but modifying input and output layers to work with 4 channels (RGB-IR). **T3** and **T2** make the upsampling progressively as in SRGAN and ESRGAN, two modules of an upsampling factor of 2 were used to obtain final images scaled by a factor of 4. With these two experiments, we test the contribution of the IR band in the performance of super-resolution in remote sensing images.

In experiment **T4**, the upsampling stage is done with just one block instead of the two blocks used by the original PSNR-oriented model. The original weights of the residual part of PSNR-oriented model (pre-trained on DIV-2K) were used to initialize the model. The input and output layers were modified as in experiment **T3** and the new weights were initialized with random values, following a Gaussian distribution with 0 mean and 0.001 of variance.

In experiment **T5**, the upsample modules of the PSNR-oriented model were removed, and the LR images were upsampled to the resolution of the HR images before being input to the network, using bicubic interpolation. The initial weights of the residual part of PSNR-oriented model were copied from the pre-trained model, as in **T4**, and then fine-tuned with the WV2 dataset. This experiment is useful especially when the dataset contains pair LR-HR images obtained from different platforms, where pre-processing steps like co-registration, the LR images need to be upsampled to be with the same scale of the HR image, providing a common reference to the images to be able to be compared.

Table 1 provides a brief summary of all the experiments.

Experiments	Description
T1	No Fine tuning 3D (Model PSNR-Oriented)
T2	Fine Tuning 3D
T3	Fine Tuning 4D
T4	Fine Tuning 4D with only 1 module of upsample
T5	Fine Tuning 4D and Pre-Upsample

Table 1. Summary of experiments

We used the code and pre-trained models of the authors of ESRGAN [12], making the necessary changes to work with remote sensing images. We used Adam optimization with a learning rate of 10^{-4} and L1-loss. The models were trained for 45K iterations with a mini-batch of size 2, halving the learning rate at 20k, 30k and 40k iterations. Random crops of 128x128 were extracted from image tiles, making the training process faster and lowering memory consumption. Horizontal and vertical flips were applied for data augmentation.

Metrics used for evaluation are Peak-Signal-to-Noise-Ratio (PSNR), Structural Similarity Index (SSIM) and ERGAS index [13] (*Erreur Relative Globale Adimensionnelle de Synthese*), computed using the four (RGB-IR) channels. PSNR and SSIM are the standard metrics, computed on 16-bit data due to the dynamic range of the remote sensing images. The ERGAS index measures the quality of the super-resolved image taking into account the scaling factor (ratio) between the low resolution and high-resolution images. The ERGAS index, unlike the PSNR and SSIM, is better when it is closer to zero:

$$Ergas(X_1, X_2) = 100ratio\sqrt{\frac{1}{n_{bands}}\sum_{i=0}^{n_{bands}}\left[\frac{RMSE(X_1^i, X_2^i))}{\hat{X}_2^i}\right]^2} \tag{1}$$

4. RESULTS

Figure 4 shows plots of PSNR, SSIM and ERGAS metrics for experiments **T1** to **T5** as well as the metrics for the LR image and a baseline bicubic interpolation on the test set, that is composed of 16 images. Tables 2 to 4 in the appendix present the numerical values per image.

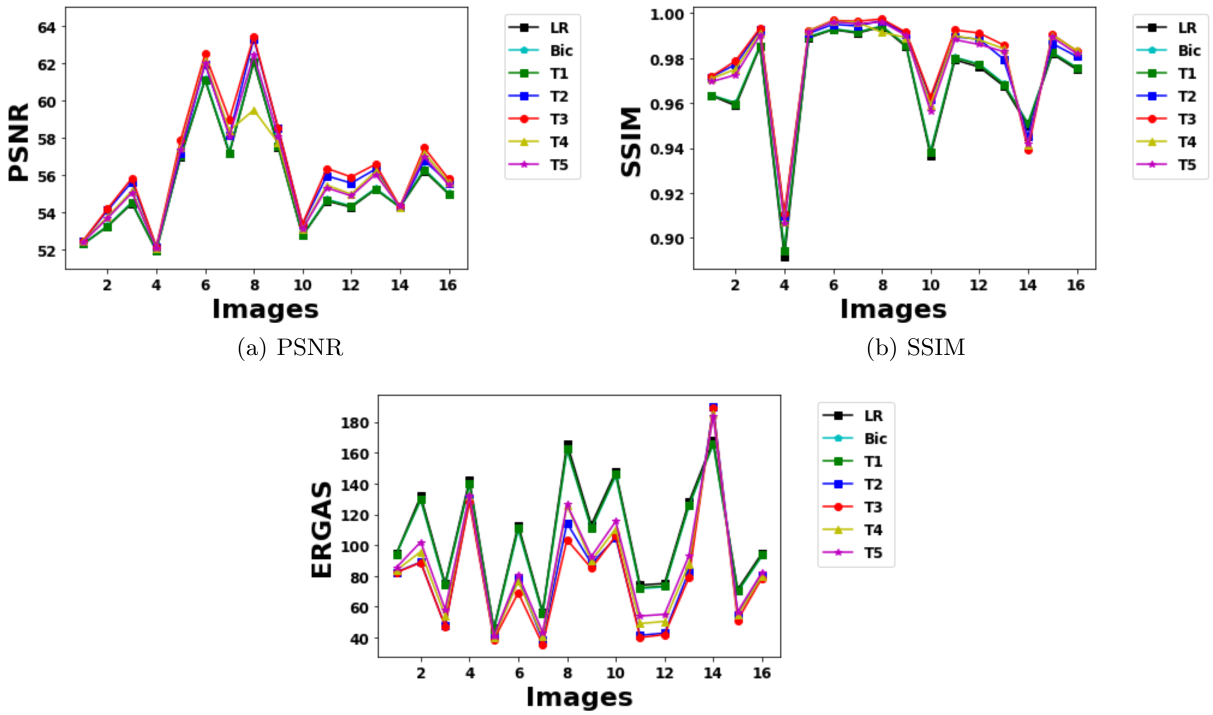
We observe that experiment **T1** presents almost the same results as the bicubic interpolation, in contrast to fine-tuning the pre-trained models (**T2** to **T5**), where in most of the images there are noticeable differences with the corresponding LR values.

In most cases experiments **T2** and **T3** show better numerical results than **T4** and **T5**, which suggests that fine-tuning either with three (RGB) or four channels (RGB-IR) using two consecutive blocks of upsampling is better than using a single upsampling block or applying a previous upsampling stage, at least for the European Cities dataset. In some textured images (e.g. a dense forest as in Figures 5), results show almost no improvement in metrics, even though visual inspection shows noticeable improvement of methods **T2** to **T5** over LR and **T1**.

Experiment **T5** shows that spatial details in the LR image can be improved after upscaling the image with a bicubic interpolation, with a PSNR difference of 0.32 dB in mean over the bicubic result.

The strategy proposed in experiment **T5** could be useful when the approach is used on pairs of low and high resolution images acquired with different sensors (e.g. Landsat and WordView-2). In this case, a co-registration step is mandatory, and therefore low resolution images must be previously upsampled to be on the same scale as the high resolution image. Moreover, comparing **T5** and **T3** results, the difference in mean of 0.65 dB may suggest that a better solution for multi-sensor applications could be to downgrade again the previously upscaled LR image and use the configuration proposed for experiment **T3**.

Another interesting experiment is **T4**, where only one block was used for upsampling (since the upsampling module can be configured for any scaling factor). Table 2 shows that the difference in mean PSNR with **T3** is 0.636 dB. Compared with **T5**, **T4** obtained a mean improvement of only 0.011 db on the test set. These results



(a) PSNR

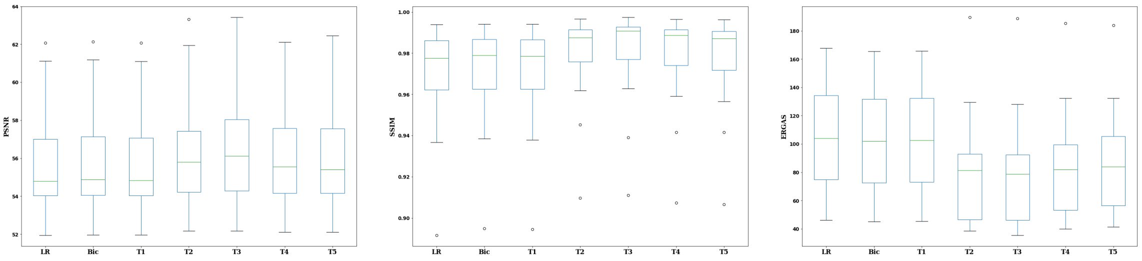
(b) SSIM

(c) ERGAS

Figure 4. PSNR, SSIM and ERGAS metrics for each image in the test set.

suggest that making an upscaling with only one block may not be a good strategy for high scaling factors, while a progressive upsampling may be a better solution.

Figure 5 presents the box-plots for the different experiments and metrics on the test set. The box-plots provide a good indication of the distribution of results, where the size of the box encloses the inter-quartile range which encompasses 50% of the data around the median. The plots also show that **T3** presents the best results according to the three metrics, with a marginal gain above the second one, which is **T2**. The methodological difference between these two experiments is that **T3** was trained with an additional channel (IR), which is important for several types of analysis in remote sensing. Typical super-resolution algorithms consider only 3 channels. In general, experiments using the pre-trained model show better results that directly using the PSNR-oriented model or the bicubic interpolation.



(a) Box-plot - PSNR

(b) Box-plot - SSIM

(c) Box-plot - ERGAS

Figure 5. Box-plot showing median, inter-quartile range and some outliers on the test set.

Figure 6 shows some RGB images and false colour images (Infrared, Red and Green) from the test set. The false-color in remote sensing is usually used for analyzing vegetation areas, where depending on the vegetation type and condition, several shades of red appear due to the high reflectance that vegetation presents in the infrared band [14].

5. CONCLUSION

This work evaluated the use of the ESRGAN architecture for the super-resolution of remote sensing images. Several configurations were tested, considering the number of channels involved and the upsampling methodology. The European Cities WorldView-2 dataset was used for training and testing the models. The best results were obtained using four channels (RGB-IR) and initializing the network with the weights of the pre-trained model, with a progressive upscaling approach. As future work we plan to add an adversarial training stage, and to test the models using LR and HR pairs from different sensors.

ACKNOWLEDGMENTS

This work has been partially supported by the ARTEMISAT-2 (CTM2016-77733-R) and MALEGRA TEC2016-75976-R projects, funded by the Spanish AEI, FEDER funds,and by the Spanish Ministerio de Economía y Competitividad, respectively. L.S.R. would like to acknowledge the BECAL (Becas Carlos Antonio López) scholarship for the financial support.

REFERENCES

[1] 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, 2015.

[2] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, *et al.*, "Photo-realistic single image super-resolution using a generative adversarial network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4681–4690, 2017.

[3] Z. Wang, J. Chen, and S. C. Hoi, "Deep learning for image super-resolution: A survey," *arXiv preprint arXiv:1902.06068*, 2019.

[4] X. Wang, K. Yu, S. Wu, J. Gu, Y. Liu, C. Dong, Y. Qiao, and C. Change Loy, "Esrgan: Enhanced super-resolution generative adversarial networks," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

[5] C. B. Collins, J. M. Beck, S. M. Bridges, J. A. Rushing, and S. J. Graves, "Deep learning for multisensor image resolution enhancement," in *Proceedings of the 1st Workshop on Artificial Intelligence and Deep Learning for Geographic Knowledge Discovery*, pp. 37–44, ACM, 2017.

[6] D. Pouliot, R. Latifovic, J. Pasher, and J. Duffe, "Landsat super-resolution enhancement using convolution neural networks and sentinel-2 for training," *Remote Sensing*, vol. 10, no. 3, p. 394, 2018.

[7] E. S. A. (ESA), "Worldview-2 european cities." `https://earth.esa.int/web/guest/-/worldview-2-european-cities-dataset`. Access: 22-07-2019.

[8] C. o. E. Penn State and mineral science, "Exploring imagery and elevation data in gis applications - preprocessing." `https://www.e-education.psu.edu/geog480/node/497`. Access: 29-07-2019.

[9] L. H. G. Solutions, "Envi software." `https://www.harris.com/solution/envi`. Access: 22-07-2019.

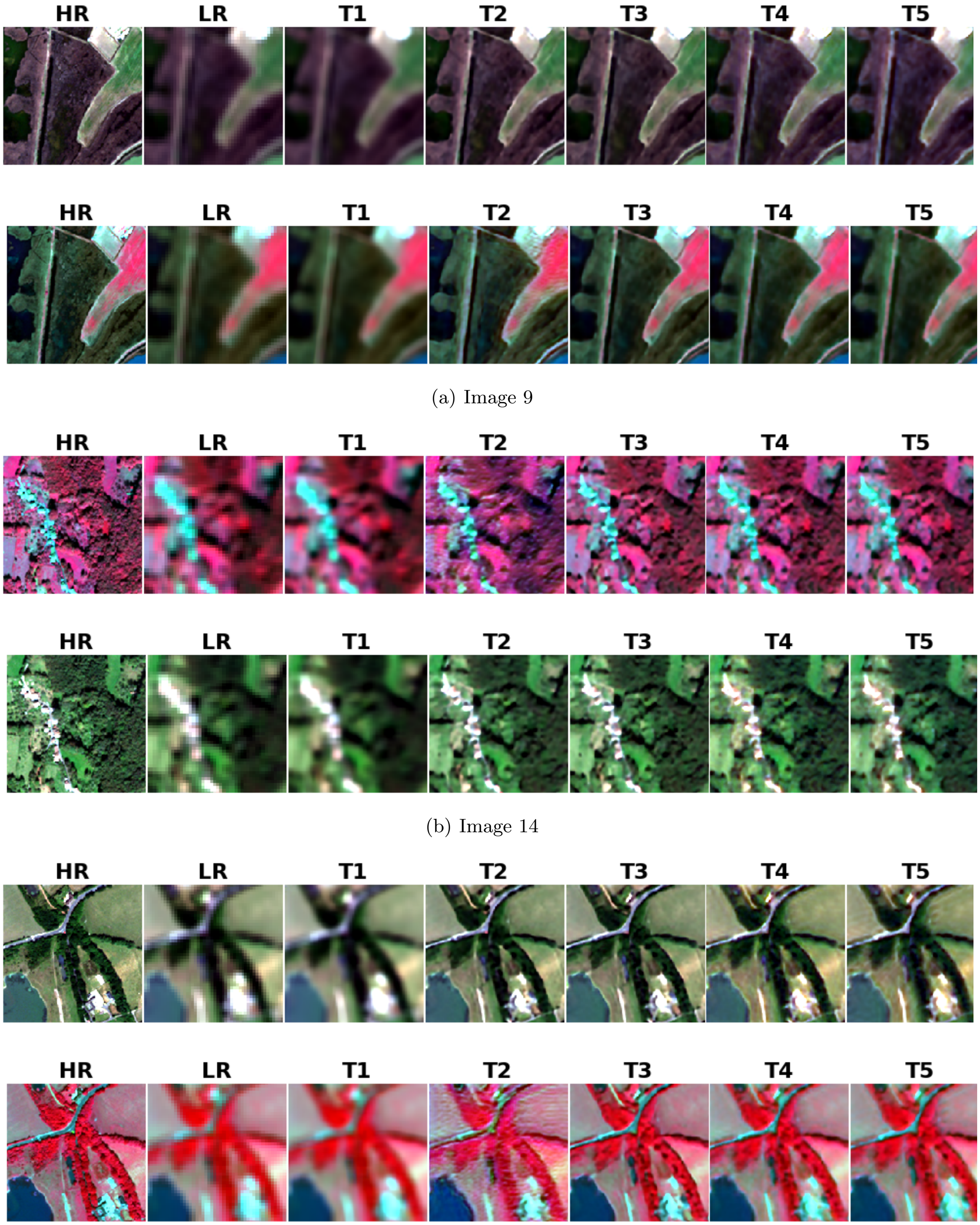
[10] T. Cooley, G. P. Anderson, G. W. Felde, M. L. Hoke, A. J. Ratkowski, J. H. Chetwynd, J. A. Gardner, S. M. Adler-Golden, M. W. Matthew, A. Berk, L. S. Bernstein, P. K. Acharya, D. Miller, and P. Lewis, "Flaash, a modtran4-based atmospheric correction algorithm, its application and validation," in *IEEE International Geoscience and Remote Sensing Symposium*, vol. 3, pp. 1414–1418 vol.3, June 2002.

[11] Y. Blau, R. Mechrez, R. Timofte, T. Michaeli, and L. Zelnik-Manor, "The 2018 pirm challenge on perceptual image super-resolution," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 0–0, 2018.

[12] X. Wang, K. Yu, S. Wu, J. Gu, Y. Liu, C. Dong, Y. Qiao, and C. Change Loy, "Esrgan: Enhanced super-resolution generative adversarial networks." `https://github.com/xinntao/BasicSR`, 2019.

[13] L. Wald, *Data fusion: definitions and architectures: fusion of images of different spatial resolutions*. Presses des MINES, 2002.

[14] S. Centre for Remote Imaging and P. (CRISP), "Interpreting optical remote sensing images." `"https://crisp.nus.edu.sg/~research/tutorial/opt_int.htm"`. Access: 27-06-2019.



(a) Image 9

(b) Image 14

(c) Image 16

Figure 6. Results for test images 9, 14 and 16. RGB and IR-RG (false-color) channels

6. APPENDIX

	LR	**Bicub**	**T1**	**T2**	**T3**	**T4**	**T5**
Image 1	52.31	52.31	52.31	52.49	**52.49**	52.47	52.43
Image 2	53.25	53.28	53.27	54.11	**54.2**	53.77	53.69
Image 3	54.49	54.56	54.52	55.64	**55.84**	55.16	55.05
Image 4	51.95	51.96	51.96	**52.17**	**52.17**	52.11	52.11
Image 5	56.97	57.08	57.02	57.19	**57.89**	57.52	57.37
Image 6	61.11	61.19	61.1	61.95	**62.48**	62.11	61.99
Image 7	57.15	57.25	57.2	58.15	**58.98**	58.38	58.08
Image 8	62.06	62.13	62.08	63.31	**63.42**	59.49	62.44
Image 9	57.5	57.59	57.54	58.51	**58.53**	57.74	58.1
Image 10	52.76	52.78	52.77	53.36	**53.39**	53.13	53.11
Image 11	54.62	54.7	54.66	55.96	**56.36**	55.46	55.32
Image 12	54.28	54.35	54.32	55.57	**55.9**	54.98	54.89
Image 13	55.24	55.3	55.26	56.32	**56.58**	56.19	56.04
Image 14	54.28	54.31	54.28	54.25	**54.31**	54.28	54.31
Image 15	56.23	56.32	56.27	56.78	**57.46**	57.2	56.99
Image 16	54.98	55.04	55	55.59	**55.79**	55.62	55.47
Mean	55.573	55.634	55.59	56.334	**56.611**	55.975	55.961
Std-dev	2.878	2.9	2.88	3.0678	3.188	2.682	2.965

Table 2. PSNR on test set

	LR	**Bicub**	**T1**	**T2**	**T3**	**T4**	**T5**
Image 1	0.963307	0.963512	0.96347	0.971557	0.971735	0.97081	0.969484
Image 2	0.958772	0.960148	0.959754	0.977177	0.978788	0.975053	0.972508
Image 3	0.985038	0.985643	0.985414	0.992671	0.993321	0.991235	0.990133
Image 4	0.891678	0.894971	0.894471	0.909703	0.911064	0.907371	0.906537
Image 5	0.989018	0.989487	0.98935	0.990938	0.992193	0.992213	0.991538
Image 6	0.992661	0.993009	0.992823	0.995043	0.996734	0.996379	0.996169
Image 7	0.991081	0.991583	0.9914	0.994257	0.996356	0.995852	0.995205
Image 8	0.993888	0.994171	0.994066	0.99666	0.99738	0.991429	0.996134
Image 9	0.985004	0.985743	0.985533	0.990717	0.991416	0.989118	0.989937
Image 10	0.936682	0.938544	0.937945	0.961764	0.96274	0.959171	0.956466
Image 11	0.979202	0.980393	0.980118	0.989476	0.992473	0.989934	0.988154
Image 12	0.976008	0.977337	0.977015	0.988345	0.991122	0.988088	0.986059
Image 13	0.967603	0.968864	0.968357	0.979492	0.985778	0.984207	0.98285
Image 14	0.949775	0.9509	0.950808	0.945239	0.939135	0.941506	0.941664
Image 15	0.981812	0.982581	0.982326	0.986429	0.99051	0.990199	0.989251
Image 16	0.97494	0.975849	0.975557	0.980708	0.983144	0.983423	0.981954
Mean	0.969	0.9707	0.9705	0.978	0.9796	0.9778	0.97715
Std-dev	0.0264	0.0257	0.0258	0.0228	0.0238	0.0238	0.0241

Table 3. SSIM on test set

	LR	**Bicub**	**T1**	**T2**	**T3**	**T4**	**T5**
Image 1	94.44	94.1	94.21	82.46	82.23	83.6	85.43
Image 2	131.74	129.08	129.79	89.04	88.69	95.76	102.07
Image 3	75.18	73.42	74.04	47.56	47.45	54.26	58.38
Image 4	142.1	139.66	140.01	129.43	128.09	132.29	132.31
Image 5	46.1	44.9	45.27	40.15	38.76	40.02	41.25
Image 6	112.94	109.97	111.23	78.32	69.1	76.49	80.83
Image 7	56.8	55.21	55.76	38.45	35.51	40.53	43.52
Image 8	165.95	161.56	162.58	114.33	103.53	125.67	126.4
Image 9	113.23	110.41	111.1	88.05	84.98	89.78	92.65
Image 10	147.65	144.85	145.76	104.68	105.99	110.61	115.39
Image 11	74.08	71.82	72.36	41.51	40.1	49.15	54
Image 12	75.12	72.87	73.45	42.95	41.96	50.51	55.16
Image 13	127.81	124.96	125.88	82.94	78.93	87.45	93.05
Image 14	167.86	165.59	165.81	189.56	188.7	185.36	183.8
Image 15	71.23	69.57	70.14	54.86	51.33	54.58	56.85
Image 16	94.77	92.98	93.54	80.21	78.67	79.98	82.37
Mean	106.06	103.81	104.43	81.53	79.00	84.75	87.72
Std-dev	38.3813	37.7794	37.8561	40.1779	40.1407	39.3340	38.2496

Table 4. ERGAS on test set

Name	**Title**	**Research Field**	**Personal Website**
Luis Salgueiro	PhD. Candidate	Deep learning models applied to remote sensing.	https://imatge.upc.edu/web/people/luis-fernando-salgueiro
Vernica Vilaplana	Associate Professor	Computer vision, image processing, machine learning, deep learning biomedical and remote sensing applications.	https://imatge.upc.edu/web/people/veronica-vilaplana
Javier Marcello	Full Professor	Remote sensing image processing techniques.	http://iocag.ulpgc.es/people/javier-marcello-ruiz