



SR4RS: A Tool for Super Resolution of Remote Sensing Images

SOFTWARE METAPAPER

RÉMI CRESSON 

]u[ubiquity press

ABSTRACT

The SR4RS software includes tools to apply super-resolution methods on remote sensing images. It employs TensorFlow on the deep learning side, and relies on GDAL and the Orfeo ToolBox to deal with geospatial data. The software is written in simple python, and provides user-oriented applications to train and apply state of the art models on images.

CORRESPONDING AUTHOR:

Rémi Cresson

French Research Institute
for Agriculture, Food and
Environment (INRAE), FR

remi.cresson@inrae.fr

KEYWORDS:

super-resolution; remote
sensing; user-oriented; orfeo
toolbox; tensorflow; geospatial

TO CITE THIS ARTICLE:

Cresson R 2022 SR4RS: A Tool
for Super Resolution of Remote
Sensing Images. *Journal of
Open Research Software*, 10: 1.
DOI: [https://doi.org/10.5334/
jors.369](https://doi.org/10.5334/jors.369)

(1) OVERVIEW
INTRODUCTION

Single image super-resolution consists in generating a high-resolution image from a low-resolution image. In recent years, deep learning approaches have emerged as state of the art for single image super-resolution [1]. However, applying them to remote sensing images requires a few adjustments of the methodology applied on traditional true color images datasets. Remote sensing are a bit different in nature, because they usually have more than the red, green and blue channels, they often have, for instance, spectral bands in the near infrared domain. Those channels are generally encoded on 16 bits, against 8 bits for traditional images. Moreover, remote sensing images usually include metadata enabling to map each pixel on the earth surface, in computing their position in physical units. For this purpose, images generated with super-resolution methods must have an updated metadata, determined from the transformation applied to the low resolution image. Our software implement the well established single image super-resolution approach of Ledig et al. [1]. It provides an application to train the network, and we have improved the training step using recent advances in image generation [2, 3] that improve the stability of the training process. A second application enables to apply the trained model over low resolution images, to generate synthetic high resolution images. Up to our knowledge, while a growing number of software are available on GitHub for traditional images, no such open-source software is available for remote sensing image processing, integrating the specificity of geospatial data.

IMPLEMENTATION AND ARCHITECTURE

Our software relies on the Geospatial Abstraction Library (GDAL) [4] and the Orfeo ToolBox (OTB) [5] to perform I/O operations on remote sensing images and geospatial vector layers. TensorFlow [6] is used to perform the computations to train and apply the deep neural networks. We rely on an extension of the OTB that eases the use of TensorFlow on remote sensing images, OTBTF (Figure 1). We use OTB applications shipped in this extension, to (1) extract images patches in remote sensing images,

and (2) apply a trained and serialized TensorFlow model, the SavedModel. Our software implements (i) the model training from the remote sensing images patches, using TensorFlow for the model build and the parameters optimization and GDAL for the patches images reading, and (ii) the high resolution image generation, from the trained model, using OTBTF. Figure 3 summarizes the involved software stack.

1. Model training This part consist in accessing patches of images using GDAL, and feeding the collected samples to a TensorFlow computational graph implementing a deep convolutional neural network. This neural network is optimized each step, where new images patches are gathered, and fed to the training process (Figure 2). The neural network consist in an implementation of Ledig et al. [1] network, modified to suit remote sensing images: the number of channels in the images, which can be greater than 3 in remote sensing images, is detected and used to build the model accordingly. This way, we can train a model that is not restricted to true color images, and that can process a multispectral remote sensing image for instance. The loss function consist in a contribution of diverse functions that express the gap between the target image, i.e. the high-resolution image, and the image generated by the model. Our application enables the user to carefully set the weight of each function used to derive the final loss function, used to optimize the model parameters. The application also writes logs for the TensorFlow’s dashboard, called Tensorboard, enabling the visualization of losses values and preview images generated from the network. The user can then

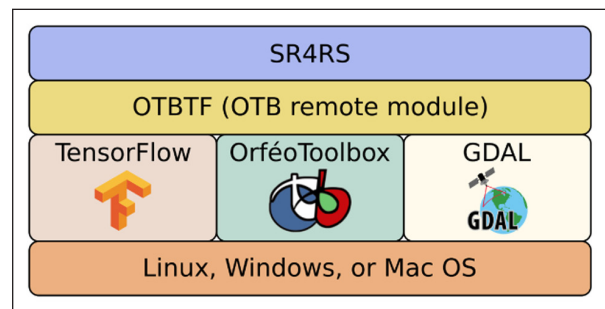


Figure 1 Cross platform software stack.

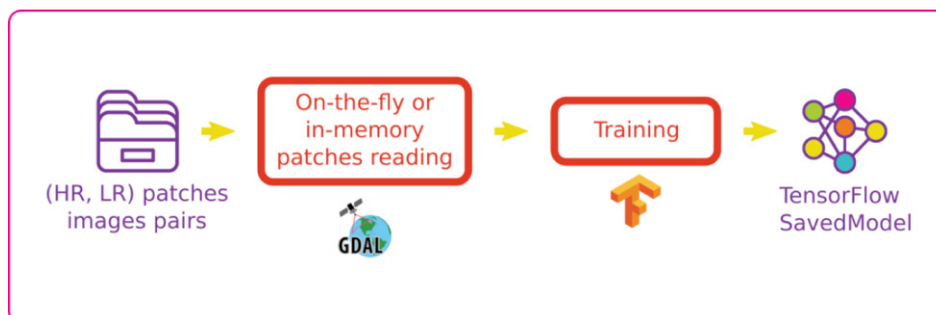


Figure 2 Step 1: model training. GDAL is employed to read the input images (on-the-fly or in-memory). TensorFlow performs the training of the network, and export the trained weights in the SavedModel format.

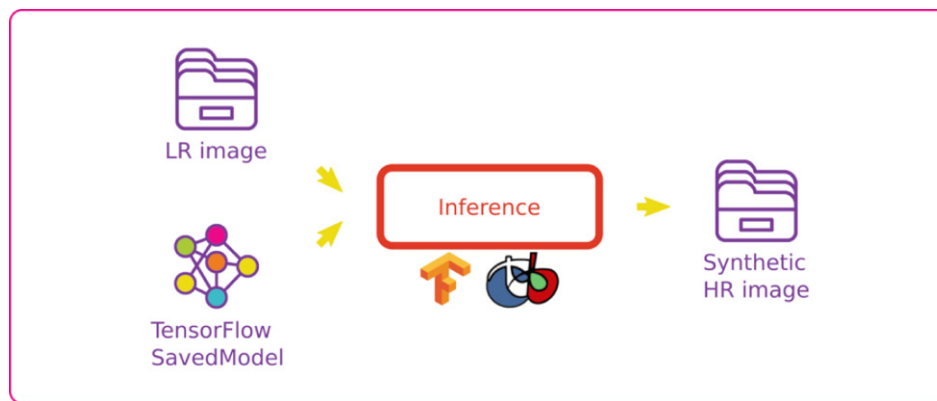


Figure 3 Step 2: inference using a trained model, to generate a synthetic high-resolution geospatial image from one entire low-resolution remote sensing image. Weights of the trained model is stored in the SavedModel format.

experiment a variety of settings, and monitor the losses dynamics and the visual results during the training process. During the training, the model parameters are regularly saved in the file-system. Hence the application can train a new model from scratch, or from a particular state saved in the file-system. At the end of the execution process, the model is exported in the SavedModel format, and ready to be used in the next step.

2. Model inference In this part, a second application is used to apply the previously trained model on low-resolution remote sensing image (Figure 3). The application employs OTBTF to read the SavedModel and apply it over the entire input image. Since the entire image cannot fit in the GPU memory or computer RAM, the output image is generated in a region-by-region fashion, using the so-called streaming mechanism implemented in the OTB. The application eases the user experience, since he or she just has to provide the input image path, the SavedModel path, and chose a file name for the generated high resolution image.

QUALITY CONTROL

Since the training process is stochastic, unit tests for the training cannot be set. The inference part however, is guaranteed through OTBTF and OTB, which include a number of unit and functional tests, that can be trigger after the build of the library. In particular, OTBTF checks that the inference of a deep convolutional network lead to the correct result. The software documentation explains how to install the software in a simple manner, using an existing docker image where all the dependencies are installed, and ready to use. The software documentation provides a detailed description of the application parameters. It also include a typical use case, detailing each steps that the user has to perform, and explaining how to properly use the software with well known remote sensing products. Finally, a pre-trained model is provided with the software (see Reuse section) that can be used directly to generate high-resolution images from Sentinel-2 images with Red, Green, Blue and Near-infrared channels.

(2) AVAILABILITY OPERATING SYSTEM

Our software can run on any platform thank to the OTBTF docker images. While it theoretically can be installed on Windows, Linux, and Mac, due to the used libraries, we only have extensively used it on linux environments (CentOS 8, and Ubuntu 18.04).

PROGRAMMING LANGUAGE

Python (minimum, 3.5)

ADDITIONAL SYSTEM REQUIREMENTS

For the training step, we advise to use a GPU suited for deep learning computations. While the inference step can be performed on CPU, we strongly recommend a GPU to speed-up the process.

DEPENDENCIES

- OTB >= 7.2.0
- GDAL >= 2.4.0
- Tensorflow >= 1.15

LIST OF CONTRIBUTORS

- Rémi Cresson
- Benjamin Commandre
- Vincent Delbar
- Nicolas Narçon

SOFTWARE LOCATION

Archive (e.g. institutional repository, general repository) (required – please see instructions on journal website for depositing archive copy of software in a suitable repository)

Name: SR4RS

Persistent identifier: [10.5281/zenodo.4537578](https://zenodo.org/record/4537578)

Licence: MIT

Publisher: Rémi Cresson

Version published: r1.3

Date published: 26/01/22

Code repository GitHub (required)

Name: SR4RS

Persistent identifier: <https://github.com/remicres/sr4rs>

Licence: MIT

Date published: 12/02/2021

Emulation environment (if appropriate)

LANGUAGE

English

(3) REUSE POTENTIAL

The software could be reused by researchers in the field of deep learning applied to remote sensing images. In particular, the software provides a practical framework to implementation new methods for super resolution of remote sensing images. Along with the software is

provided a pre-trained model corresponding to the default settings values of the training application of the tool. The pre-trained can be downloaded from the link available in the documentation (*PRETRAINED_MODELS.md*). The model was trained from 250 different Spot-6 and Spot-7 scenes covering the entire France Mainland, acquired during the year 2020, from march to october, and the Sentinel-2 images acquired close to the same day (Sentinel-2 products from the THEIA Land data center). Spot-6 and Spot-7 images were interpolated at 2.5 meters and radiometrically calibrated to match the Sentinel-2 radiometry. Around 150k patches were used to train the model. The users can either start the training from the weights of the pre-trained model, or simply use it directly on Sentinel-2 images (channels Red, Green, Blue and Near-infrared at 10 meters spacing). We provide representative illustrative images obtained with the provided model in *Figure 4*.

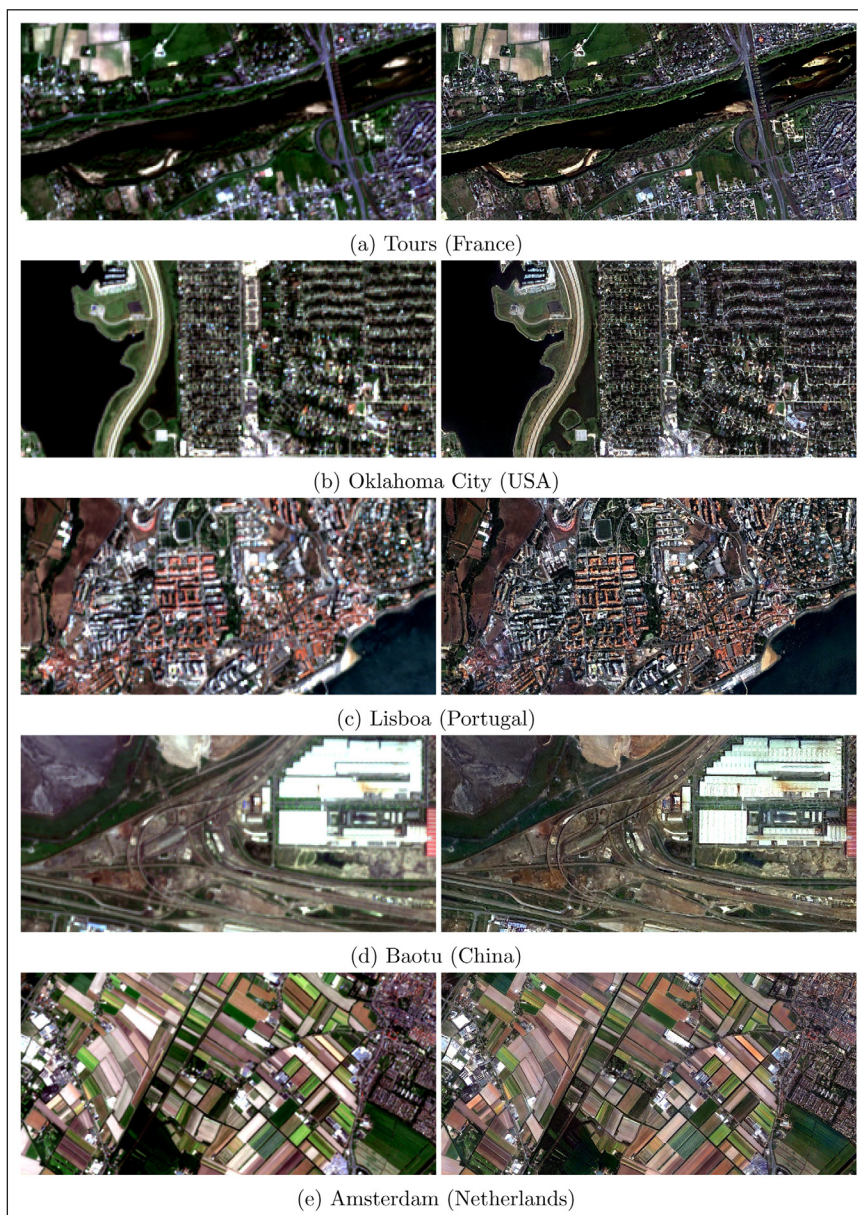


Figure 4 Results obtained all around the earth using the available pre-trained model. Left: original Sentinel-2 images (in RGB colors), right: synthetic high-resolution images (2.5 meters spacing) generated with the software.

It should also be noted that the model can be applied inside any pipeline composed of OTB applications, thank to the streaming support.¹ Also, thank to the integration of OTB inside the QGIS software, any model obtained with our software can be used directly from QGIS with the *TensorflowModelServe* application, which is part of the OTBTF. QGIS provides a number of features to interact in graphical user interface with the OTB, i.e. input image choice, and display the result of in the cartographic canvas. QGIS also enable the chaining of various software for geospatial data processing with its so-called model builder (e.g. GDAL/OGR, GRASS, SAGA, etc.). We note that to enable the OTBTF applications in QGIS, the library must be compiled in the same environment, and with the QGIS descriptors.

NOTE

- <https://www.orfeo-toolbox.org/CookBook/PythonAPI.html#in-memory-connection>.

ACKNOWLEDGEMENTS

This work was granted access to the HPC resources of IDRIS under the allocation AD011011608 made by GENCI. The author wants to thank Manuel Grizonnet, Julien Michel, Yannick Tanguy, Victor Poughon, David Youssefi, Gérald Fenoy, Yvain Tisserand, Geoffroy Daguet, The GDAL, OTB, TensorFlow and stackoverflow community, the OSGeo foundation.


FUNDING STATEMENT

This work has been supported by the Programme National de Télédétection Spatiale (PNTS, <http://programmes.insu.cnrs.fr/pnts/>), grant n° PNTS-2020-07.

COMPETING INTERESTS

The authors have no competing interests to declare.

AUTHOR AFFILIATIONS

Rémi Cresson  orcid.org/0000-0003-4017-2635
French Research Institute for Agriculture, Food and Environment (INRAE), FR

REFERENCES

- Ledig C, Theis L, Huszár F, Caballero J, Cunningham A, Acosta A, Shi W.** Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017; 4681–4690. DOI: <https://doi.org/10.1109/CVPR.2017.19>
- Karras T**, et al. “Progressive growing of gans for improved quality, stability, and variation.” *arXiv preprint arXiv:1710.10196*; 2017.
- Karnewar A, Wang, O.** Msg-gan: Multi-scale gradients for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020; 7799–7808. DOI: <https://doi.org/10.1109/CVPR42600.2020.00782>
- Warmerdam F.** The geospatial data abstraction library. In *Open source approaches in spatial data handling*. 2008; 87–104. Berlin, Heidelberg: Springer. DOI: https://doi.org/10.1007/978-3-540-74831-1_5
- Grizonnet M, Michel J, Poughon V, Inglada J, Savinaud M, Cresson R.** Orfeo ToolBox: open source processing of remote sensing images. *Open Geospatial Data, Software and Standards*. 2017; 2(1): 1–8. DOI: <https://doi.org/10.1186/s40965-017-0031-6>
- Abadi M, Barham P, Chen J, Chen Z, Davis A, Dean J, ..., Zheng X.** Tensorflow: A system for large-scale machine learning. In *12th USENIX symposium on operating systems design and implementation (OSDI 16)*. 2016; 265–283.

TO CITE THIS ARTICLE:

Cresson R 2022 SR4RS: A Tool for Super Resolution of Remote Sensing Images. *Journal of Open Research Software*, 10: 1. DOI: <https://doi.org/10.5334/jors.369>

Submitted: 12 February 2021 Accepted: 17 February 2022 Published: 02 March 2022

COPYRIGHT:

© 2022 The Author(s). This is an open-access article distributed under the terms of the Creative Commons Attribution 4.0 International License (CC-BY 4.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited. See <http://creativecommons.org/licenses/by/4.0/>.

Journal of Open Research Software is a peer-reviewed open access journal published by Ubiquity Press.