Deep Neural Network for Super-Resolution of Multitemporal Remote Sensing Images

A Degree Thesis Submitted to
the Faculty of the Escola Tècnica d'Enginyeria de Telecomunicació de Barcelona
Universitat Politècnica de Catalunya

In partial fulfilment of the requirements for the
Bachelor's degree in Telecommunications Technologies and Services Engineering
Major in Telecommunications Systems

Author
Pol Masó Ayats
Advisors
Verónica Vilaplana Besler
Luis Fernando Salgueiro Romero

Barcelona, June 2020

Deep Neural Network for Super-Resolution of Multitemporal Remote Sensing Images

Abstract

Since few years ago, artificial intelligence (AI) has become a spotlight technology in which a lot of people are interested in. Most of them want to do research and use it to solve a huge variety of modern and difficult computing problems which could be associated with a wide variety of interesting fields. As soon as AI has improved, convolutional neural networks (CNN) have taken an excellent role in the world of image processing, in particular, for Remote Sensing applications. Nevertheless, artificial intelligence for multi-image superresolution from multi-temporal imagery has received little attention so far.

In this work, it is proposed a CNN, which exploits both spatial and temporal correlations in the low-resolution images by using two different convolutional layers (2D and 3D convolutions) to combine multiple satellite images from the same scene which are taken in different temporal moments.

The experiments have been carried out using a dataset generated by Sentinel-2 (European Space Agency satellite) images captured over 2 different places over the world, New York and El Cairo. This model aims to obtain super-resolution images from five low-resolution images, or less, being aware of the number of input images that the CNN has.

Key words--*Multi-temporal images, convolutional neural networks, multi-image superresolution, artificial intelligence*

Xarxa Neuronal Profunda per Super-Resolució d'Imatges de Teledetecció Multitemporal

Resum

Des de fa uns anys, la intel·ligència artificial (IA) s'ha convertit en una tecnologia destacada en la qual moltes persones estan interessades. La majoria volen fer recerca i utilitzar-la amb l'objectiu de resoldre una gran varietat de problemes informàtics moderns i difícils que es poden associar en una àmplia varietat de camps interessants. Tan aviat com l'IA va millorar, les xarxes neuronals convolucionals (CNN) van desenvolupar un paper fonamental en el món del processament d'imatges, en particular, per aplicacions de Teledetecció. Malgrat això, la intel·ligència artificial per a la superresolució d'imatges capturades des de satèl·lits a partir d'imatges multitemporals ha rebut poca atenció fins ara.

En aquest treball, es proposa una CNN, que explota correlacions tant espacials com temporals en imatges de baixa resolució mitjançant dues capes convolucionals diferents (convolucions 2D i 3D) amb l'objectiu de combinar diverses imatges d'una mateixa escena capturades en diferents instants temporals.

Els experiments s'han realitzat mitjançant un conjunt de dades generades amb imatges del Sentinel-2 (satèl·lit de l'Agència Espacial Europea), les quals han estat obtingudes en 2 llocs diferents del món, Nova York i El Caire. L'objectiu d'aquest experiment és obtenir imatges de gran resolució a partir de cinc imatges de baixa resolució, o menys, de la mateixa escena, tenint en compte el nombre d'imatges que es tenen com a input de la CNN.

Paraules clau--*Imatges multi-temporals, xarxes neuronals convolucionals, superresolució de múltiples imatges, intel·ligència artificial*

Red Neuronal Profunda para la Súper Resolución de Imágenes de Teledetección Multitemporal

Resumen

Desde hace unos años, la inteligencia artificial (IA) se ha convertido en una tecnología destacada en la que mucha gente está interesada. La mayoría de ellos quiere investigar y usarla con el objetivo de resolver una gran variedad de problemas informáticos modernos y difíciles que podrían asociarse en una amplia variedad de campos interesantes. Tan pronto como la IA ha mejorado, las redes neuronales convolucionales (CNN) han desempeñado un papel fundamental en el mundo del procesamiento de imágenes, en particular, para aplicaciones de Teledetección. Sin embargo, la inteligencia artificial para la superresolución de imágenes satelitales a partir de imágenes multitemporales ha recibido poca atención hasta ahora.

En este trabajo, se propone una CNN, que explota las correlaciones espaciales y temporales en imágenes satelitales de baja resolución mediante el uso de dos tipos de capas convolucionales diferentes (convolución 2D y 3D) para combinar múltiples imágenes de la misma escena que se toman en diferentes instantes temporales.

Los experimentos se han llevado a cabo utilizando un conjunto de datos generados con imágenes del Sentinel-2 (satélite de la Agencia Espacial Europea), las cuales han sido capturadas en 2 lugares diferentes del mundo, Nueva York y El Cairo. El objetivo de este experimento es obtener una imagen de superresolución a partir cinco imágenes de baja resolución, o menos, de la misma escena, teniendo en cuenta el número de imágenes que se tienen como input en la CNN.

Palabras clave--*Imágenes multitemporales, redes neuronales convolucionales, superresolución de múltiples imágenes, inteligencia artificial*

Acknowledgements

First, I would like to thank the advisors of my thesis Verónica Vilaplana Besler and Luis Fernando Salgueiro Romero for their constant orientation and guidance during the months that the project has been done, giving advice and reorienting the project when problems appeared. Moreover, I would like to appreciate their daily support and knowledge on the development problems and use of software.

To conclude, I would also acknowledge the support given by my family to me during all the difficult and decisive moments which I have passed during the realization of the project.

Revision history and approval record

Revision	Date	Purpose
0	01/06/2020	Document creation
1	14/06/2020	Document revision
2	25/06/2020	Document revision
3	28/06/2020	Document revision

DOCUMENT DISTRIBUTION LIST

Name	e-mail
Pol Masó Ayats	polmaso98@gmail.com
Verónica Vilaplana Besler	veronica.vilaplana@upc.edu
Luis Fernando Salgueiro Romero	luis.fernando.salgueiro@upc.edu

Written by:		Reviewed and approved by:	
Date	01/06/2020	Date	28/06/2020
Name	Pol Masó Ayats	Name	Verónica Vilaplana Besler Luis Fernando Salgueiro Romero
Position	Project Author	Position	Project Supervisor

Table of contents

List of Figures:

List of Tables:

1. Introduction

The project was carried out in the Image and Video Processing Group (GPI) from the Signal Theory and Communications Department (TSC) at the Technical University of Catalonia (UPC).

This project consists of the design and the implementation of a Deep Neural Network with the aim of generating a super-resolution remote sensing image from a set of low-resolution images captured by European Space Agency (ESA) Sentinel-2 satellite.

1.1. Statement of purpose

Deep learning is a recent technology that is evolving very quickly. Within the world of Deep Learning we find the Convolutional Neural Networks (CNN), that have been successfully applied to image processing and in particular to enhance the resolution of remote sensing imagery. However, deep learning techniques for multi-image super-resolution from multitemporal imagery (MISR) have received little attention so far due to the lack of datasets.

For this reason, in this thesis a CNN model that address the MISR problem is presented. The model was trained using Sentinel-2 imagery from ESA. The objective of the experiment is to obtain a super-resolved image from a set of low-resolution (LR) versions of the same scene. The original high-resolution (HR) images came from Sentinel-2 and are used as target, that is, images which are used to compare the results obtained by the proposed deep learning architecture. The LR images were artificially down-scaled when the dataset was created. We used the temporal availability feature of sentinel imagery, choosing LR images which were taken in different temporal instants and use them as input for the CNN. Some pre-processing steps like the selection of images with low percentage of cloud coverage and the elimination of dark zones in images were necessary to form the dataset, the SNAP software [24] was used for that purpose.
The main contribution of this project is a CNN-based architecture to combine multiple registered images from the same scene exploiting both temporal and spatial correlations. I also discuss how a variable number of LR inputs images can modify the super-resolved obtained image using the same hyperparameters in each training process.

The starting point of this work is a system that won a challenge in 2019 [2] called DeepSUM. One of the main goals of this project was to replicate this model with the addition of the necessary modifications to deal with Sentinel-2 images. Moreover, the programming language used to implement the Deep Neural Network is also a different from the one used in the

DeepSUM project, specifically, I have used Python as the programming language and Pytorch as framework.

Other software used was ENVI, for visualizing and processing the images.

Based on the above-mentioned motivation, the main objectives of this project are:

- Learning about Phyton programming language and Pytorch framework for the implementation of Deep Neural Networks.
- Learning the basics of Deep Learning and models for super-resolution of satellite images.
- Getting familiar with one of the frameworks used for Convolutional Neural Networks (CNN) and using it to create an entire deep network.
- Learning how to manage a big dataset of images to use it to train and test the Deep Neural Network developed.
- Implementing a Deep Neural Network which processes low-resolution images to obtain a super-resolution image.
- Analyzing the results obtained while different number of LR multitemporal images are used as input to train the Deep Neural Network.

1.2. Work rganization

Work Packages, Milestones and Gantt Diagram tables followed during the project can be found at Appendix A.

1.3. Project structure

In this first chapter, a brief introduction of Multi-image Super-Resolution (MISR) was done. Chapter 2 presents the state of the art of SISR and MISR models.

Chapter 3 gives an introduction about the methodology and the experimental development of the project. Moreover, chapter 5 and 6 present the budget, project conclusions and future development respectively.

Finally, the Appendix contains additional content about the Final Thesis.

2. State of the art:

Monitoring and mapping the Earth play a very important role while using remote sensing techniques. Augmenting the accessibility of high spatial resolution, remote sensing data is useful for a lot of applications, among others, urban mapping, vegetation growth monitoring, intelligence gathering and military surveillance. Nevertheless, the augmenting quality of spatial and spectral resolution of the instruments which are onboarded on the satellites generates a large amount of data that complicates compression algorithms [25]-[26] to meet the disponible downlink bandwidth. This usually causes the reduction availability of HR products. This problem combined with the high cost of small missions is a potential incentive of needing to develop a new generation of post-processing instruments to enhance the spatial resolution of remote sensing imagery.

Techniques based on super-resolution (SR) are useful to achieve the reconstruction of high resolution (HR) pictures from some low resolution (LR) images.

There are two main methods used to obtain super-resolution images: single-image SR (SISR) and multiple-image SR (MISR). SISR exploits spatial correlations in a single image with the aim to obtain the HR version. Nevertheless, the information available in a single image is limited as some information is lost during the image formation process. Moreover, due to the use of only one specific LR input image, it requires a longer training process to achieve the objective of making the correct correspondence of the LR image with the high-resolution one.

Other applications use multiple images of the same scene in different time frameworks to be fused by means of MISR techniques, in which the obtention of the super-resolution image takes benefit of high spatial-frequency details given by the different observations of the same scene.

In terms of remote sensing, multiple images could be obtained by multiple satellites capturing the same scene at different times, by a spacecraft during multiple orbits, or may be acquired at the same time using different sensors on the same satellite. However, using a technique based in MISR is not as easy as it looks. There are relevant problems which have to be taken into account and have to be solved when these methods are used. The most usual are: invariance to absolute brightness variability, image registration, time-varying scene content and unreliable data (e.g., due to cloud coverage).

Deep learning methods have been applied successfully in SISR, but not a lot of work has been done for the MISR problem with remote sensing data.

SISR techniques can be grouped into three main classes: optimization-based methods, learning based methods and interpolation-based methods. Optimization-based methods are mainly used to model prior knowledge about natural images and include small variation priors [6], gradient-profile prior [7]-[8] and non-local similarity [9]-[10]. The addition of prior knowledge restricts the potential solution space performing higher quality solutions. Nevertheless, when the upscaling factor increases, the results degrade promptly. Moreover, these methods are computationally expensive.

Learning-based methods work modelling the concurrence between LR and HR images to obtain the SR predicted image. Some methods related to learning-based are: sparse-coding [27]-[28], anchored neighbourhood regression [29], random forest [30], k-nearest neighbours [31], among others. These techniques have a high capability of extracting high-level features from images.

More recently, Convolutional Neural Networks (CNN) have been exploited for remote sensing imagery. CNN takes benefit of the fact that the input is composed of images and they constrain the model in a more sensible way. Particularly, the layers of a CNN have neurons arranged in 3 dimensions: width, height and depth (the word depth refers to the third dimension of a volume, not the depth of the full Neural Network, this is the number of layers in a network). Each layer of a CNN transforms the 3D input volume to a 3D output volume of neuron activations.

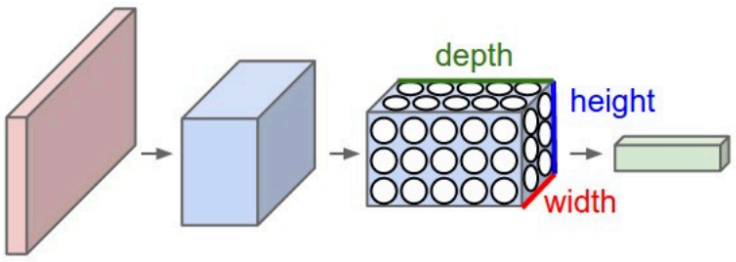


Figure 2.1. Simple example of a Convolutional Neural Network. (2020). [Photo]. CS231n Convolutional Neural Networks for Visual Recognition. https://cs231n.github.io/convolutional-networks/#overview

On the other hand, the first work about MISR was proposed by Tsai and Huang [12], in which a frequency-domain technique to fuse multiple under-sampled images with sub-pixels displacements to improve Landsat TM spatial resolution images was used. Despite its difficulty, other MISR methods related to spatial-domain techniques were proposed over the years. Some of them are: iterative back projection (IBP) [13], projection onto convex sets (POCS) [14]-[15], regularized methods [16]-[17], etc.

The IBP [13] attempts to improve an initial conjecture of the super-resolved image by back projecting the difference between an artificial LR picture and real LR pictures to the SR image. The cons are coming from the difficulty of the model image degradation processes and the awkwardness in including image priors.

POCS [14]-[15] tries to estimate the SR images below meticulous motion compensations while restoration and interpolation problems are solved. This method is affected by high computational cost and slow convergence.

Regularized methods [16]-[17] are the most efficient multi-frame SR reconstruction approximation. They preserve edge information while eliminating image noise.

A few years ago, convolutional neural networks which use MISR methods started to be employed to solve problems in the context of video super-resolution [32]-[33]. Two steps are used to complete this work: motion estimation and compensation followed by an up-sampling process. A CNN is trained to decipher both motion estimation and HR image reconstruction tasks by setting up a number of pixels-dependent filters and residual correction.

Even though, not a lot of work has been done on deep learning MISR techniques in the context of remote sensing. For this reason, an end-to-end trainable CNN and using MISR method in satellite images is aimed to be tackled in this thesis by converting the LR input images in a single SR image.

3. Methodology:

3.1. The Sentinel-2 Dataset

European Sentinel-2 satellite [3]-[4], is an Earth observation satellite which is part of the group of missions from European Space Agency (ESA) inside the Copernicus special program. It was launched in two stages. Sentinel-2A was launched in 2015 into a Sun-synchronous orbit at an altitude of 786km. Moreover, its cufflink, Sentinel-2B, was launched one year after, in 2016 in the same orbit, but with a lag of 180º, giving a high revisit frequency. Specifically, the revisit time between the two satellites is 5 days and each satellite has a revisit itself time of 10 days.

The table 3.1 contains a summary of the orbital information of Sentinel-2A and 2B:

ALTITUDE	INCLINATION	PERIOD	CYCLE	GROUND-TRACK DEVIATION	LOCAL TIME AT DESCENDING NODE
786 km	98.62 deg	100.6 min	10 days	+- 2 km	10:30 hours

Table 3.1. Orbital information of Sentinel-2 satellites.

Sentinel-2 brings a multispectral high-resolution camera with 12 spectral bands which is designed to map land cover and vegetation growth across the entire globe. It uses a sweeping system along the path (push-broom) to generate an image 290km wide with the benefit of being able to offer a very high geometric and spectral performance in its data. The camera has two large focal planes, one in the visible (VIS) and near-infrared (NIR) bands and the other in the mid-infrared (SWIR).

Nowadays, it is difficult to find a dataset which contains both LR images and the corresponding HR observation of the same scene, as captured from the same platform. This is the main reason of having to generate a new dataset using HR images captured by Sentinel-2 satellite of ESA where LR observations for each specific scene are obtained through a down-sampling process of the HR image by assuming a sensor imaging model.

To create the dataset, I have acquired HR images with a resolution of 10m (bands 2, 3, 4 and 8). Images correspond to two different regions, New York and El Cairo, from Sentinel-2 mission, with a revisit time of 5 days. At the beginning, images from Spain were also planned to be used but, finally and with the purpose of using a dataset as different as possible in terms of image characteristics (buildings, vegetation, fields, desert, etc.), only New York and El Cairo images were used.

Images are provided as level 2A products composed of radiometrically and geometrically corrected Top-of-Atmosphere reflectance in Plate Carre projection for the visible (VIS) and near-infrared (NIR) spectral bands.

Once these images are collected, a pre-processing image treatment has been done using SNAP and ENVI software. The size of the HR collected images is 10980 x 10980 pixels. Each image has been cropped in smaller parts (128 x 128) and the dark zones have been removed because they do not give relevant information and also introduce errors while training the Deep Neural Network. The images have four channels with a bit-depth of 16 bits.

Each sample consists of five LR images of the same scene, but captured in a different time and a HR image used as the target. To obtain the LR images, a down sampling pre-processing transformation has been applied when images acquired from Seninel-2 have been down sampled to images of size 64x64.

In total, the dataset contains 13700 samples, that corresponds to 68500 LR images and 13700 HR images. The images of a specific sample are captured in different times over a maximum period of two months, during July and August 2019. Changes and weather (clouds, water, shadows, missing regions, agricultural activities, human activity, etc.) generate a problem in the similarity of the images and give a major challenge in the fusion of the images and reconstruction of the super-resolution one.

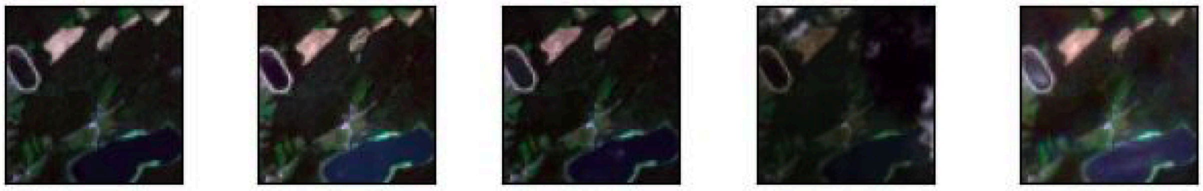


Figure 3.1. Example of one scene of five down-scaled input images.

An interesting application of MISR is the development of this SR product from multiple and more frequent LR images which could provide augmented resolution and higher temporal availability among LR images.

3.2. Proposed Method

The proposed method attempts to reconstruct a super-resolution image I^{SR} given a set of N LR images $I^{LR}_{[0,\,N-1]}$ which are from the same, scene but captured in different temporal instants. The output of the Deep Neural Network is formulated as:

$$I^{SR} = f\,(I^{LR}_{[0,\,N-1]},\, \varphi)$$

where φ corresponds to the model parameters and f refers to the mapping function which reconstructs the SR image using LR images as input. In order to use PyCharm as the programming framework, the input down-scaled images $I^{LR}_{[0,\,N-1]}$ and the target I^{HR} image used to compare the result obtained from the Deep Neural Network are depicted as tensors with shape B x N x C x H x W and B x C x rH x rW. B, represents the batch size; N, the number of images; C, the number of channels, in particular, as bands 2 (blue), 3 (green), 4 (red) and 8 (NIR) are used, the number of channels in each picture is 4. H and W represent the height and width of the input pictures respectively and r is the upscaling factor (equals to 2). Assuming that the HR image corresponds to the same region as the LR input images, there are some points to take into account.

- The LR images are all registered with each other.
- The LR images and the HR image are registered among them.
- The brightness of the HR image could be slightly different compared with the LR images.
- The presence of clouds, shadows and corrupted pixels is possible in both HR and LR images.
- The HR image is the result of not down-sampling randomly one of the LR images.

To address these issues, I took as starting point the DeepSUM model [1], which proposes a supervised deep learning technique where a CNN learns the residual between ground truth and the bicubic interpolation of the input.

Before the introduction of the images inside the CNN, a bicubic interpolation preprocessing step with scaling factor equal to 2 has been applied in all the input pictures with the aim to obtain a 128 x 128 size in all of them.

The CNN is composed of two main blocks. A schematic overview of the network is being able to be visualized in the Figure 3.2.

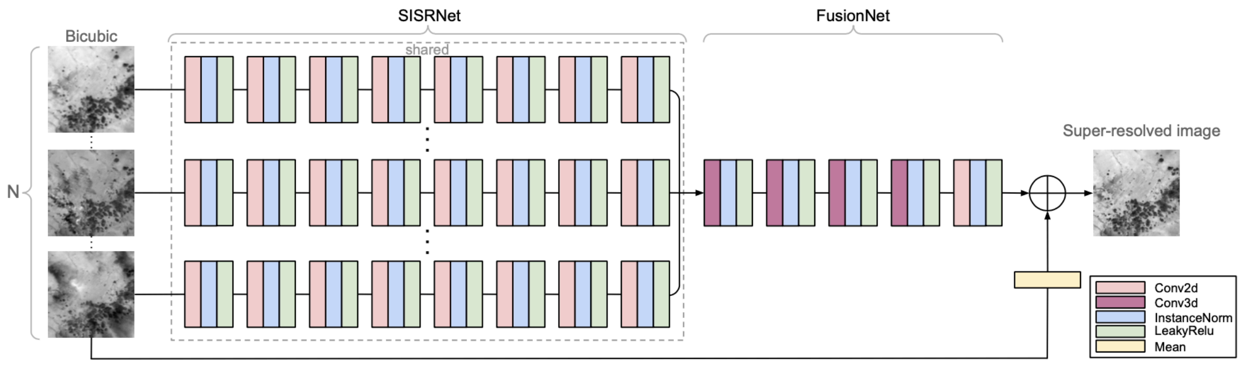


Figure 3.2. CNN model taken as reference from DeepSUM project [1] and used as scheme to develop this project CNN.

The N input bicubic-upsampled and registered images are independently treated by a SISRNet subnetwork. Then, FusionNet subnetwork is used to fuse all the features of the images obtained by the SISRNet to produce a residual image. The obtained residual image is added element-wise to the average of the bicubic-upsampled and registered images to obtain the SR image.

3.2.1. SISRNet Architecture

The main block, called SISRNet, is a feature extractor which works as a SISR network, but with the difference that, in this case, an output projection in multiple channels is acquired. Every single N input bicubic-upsampled and registered image is processed independently in each branch of the subnetwork.

The purpose of using a SISRNet as the first block is because of the high capability of this type of networks to exploit spatial correlations aimed to improve the initial bicubic-upsampled image and the learning skill which it has to extract visual features that can be used by subsequent network blocks.

Each of the N branches of the SISRNet is divided in a sequence of 8 subblocks which follow the same structure: 2D convolution layer, Instance Normalization layer and LeakyRelu layer.

The 64 filters in each of the 8 convolutional layers of the SISRNet are shared along the temporal dimension. It means that all the N input images go through the same set of filters. They use a kernel size of 3x3, with a padding and stride equal to 1.

The Instance Normalization layer is used to execute the network training as independent as possible of the brightness differences and contrast among the input images.

Finally, the LeakyRelu as activation function [20] is implemented in each layer of the SISRNet subnetwork to speeds up training. Having the mean activation closest to 0 makes training faster. In the Figure 3.3. the LeakyRelu activation function is shown.

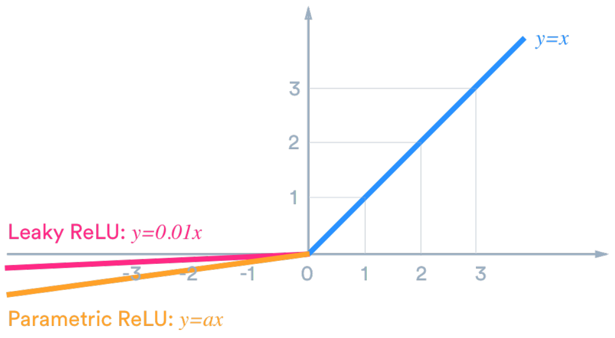


Figure 3.3. Graphic of the Leaky ReLU activation function. (2019). [Photo]. Activation Functions: Sigmoid, ReLU, Leaky ReLU and Softmax basics for Neural Networks and Deep Learning. https://medium.com/@himanshuxd/activation-functions-sigmoid-relu-leaky-relu-and-softmax-basics-for-neural-networks-and-deep-8d9c70eed91e

3.2.2. FusionNet Architecture

Replicating the DeepSUM [1] model, the featured images obtained from the SISRNet are successively fused in the FusionNet subnetwork which is composed by 4 3D convolution layers with kernel size of 2x3x3 and no padding in the temporal dimension. It aims to reduce temporal depth to 1. Moreover, a last 2D convolution layer is used to obtain the residual subnetwork output image. This architecture is useful because it carries out a slow fusion method in the feature space. It allows the network to learn how to decouple image features that are relevant to the fusion from the irrelevant changes and to create the residual image.

The high frequency details indispensable to correct the bicubically-upsampled inputs are the only features which are estimated by the network. This technique is used in remote sensing for solving image restoration problems using deep learning.

Finally, the residual image obtained on the FusionNet is element-wise added to a basic merge of the N input bicubically-upsampled and registered images in the form of their average.

The output of the Deep Neural Network is:

$$I^{SR} = (\frac{1}{N}\sum_{i\in[0,N-1]} I_i^{IRLR}) + R$$

being I^{IRLR} the input bicubically interpolated and registered images and R the residual estimated by the CNN.

3.3. Loss Function

The Euclidean distance between the HR target and the SR image obtained from the CNN is computed to calculate the loss function used to optimize the model parameters.

The Sentinel-2 satellite does not capture HR images from the same scene simultaneously, so there might be differences among these pictures because of weather conditions, modifications in the landscape and variable absolute brightness due to the revisit time which takes the satellite to capture one image from another. For this fact, the images for each sample of the dataset have been chosen as invariant as possible with the aim to solve these differences between the compared images, that is, choosing the temporal images for each sample as similar as possible and with the minimum percentage of cloud coverage.

To calculate the loss, all Euclidean distances between the I^{HR} (target) and the I^{SR} (predicted image) in each batch are computed independently. Then, the average of all Euclidean distances of each batch is obtained to minimize the loss function. To sum up, the loss computed is as follows:

$$L = mean_{I\epsilon batch} \, || \, I^{HR} - I^{SR} \, ||^2$$

The HR image used as target and also to compute the loss was chosen in relation of the cloud coverage which had in it, trying to choose the image in each sample that had a less percentage of cloud coverage. The aim of doing that is because it is tried to minimize the number of targets with areas covered by clouds or shadows with the aim to get better results and train the network more efficiently. The most interested areas to train the model are land areas; not weather conditions neither temporal distortions which could appear in some pictures if a bad selection was done.

3.4. Metrics

In this project, Peak Signal Noise Ratio (PSNR) and Structural Similarity Image (SSIM) are the metrics utilized to quantitatively evaluate the results.

3.4.1. Peak Signal Noise Ratio (PSNR)

The aim of using the PSNR [22] evaluation metric in this thesis is to obtain a quantitative value to know if the SR image has been improved or not during the training process. To tackle this, PSNR metric between the HR target and the averaged N input bicubic-upsampled and registered images is computed. Then, this result is compared to the PSNR metric between the HR target and the SR output image. A higher result in this last calculus means that the SR image has a better quality and its improvements are due to the training process.

The theoretical formula of the PSNR computed between the target and the SR output image is:

$$PSNR = 20 \log \frac{2^{32} - 1}{||I^{HR} - I^{SR}||^2}$$

The PSNR computation is meant only for pixels that are not concealed both in the target HR image and in the reconstructed image. To deal with that, dark areas were removed during the preprocessing process and the cloud coverage in pictures was as minimum as it was able.

3.4.2. Structural Similarity Image (SSIM)

SSIM [23] is a metric which quantifies image quality degradation due to some losses during data processing and it also measures, as its name shows, the difference between two similar images. It cannot decide or know which of the two images is better. The potential relevance of this metric is that using it, we are available to quantify in a 0 to 1 scale how similar the two images are.

The theoretical formula of the SSIM computed between the target and the SR output image is:

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}$$

where:

- μ_x *and* μ_y *are the average of* x *and* y *respectively.*
- σ_x^2 *and* σ_y^2 *are the variance of* x *and* y *respectively.*
- σ_{xy} *is the covariance of* x *and* y.
- $c_1 = (k_1L)^2, c_2 = (k_2L)^2$ *are two variables to stabilize the division with weak denominator.*

- *L the dynamic range of the pixel-values (in this project, the dynamic range to calculate the SSIM between the predicted image and the target is the difference between the maximum value and the minimum value of the target pixels)..*
- $k_1 = 0.01\ and\ k_2 = 0.03\ by\ default.$

In this project, we use it to estimate how much has improved the final output SR image during the training process, computing the SSIM metric between the averaged N input bicubic-upsampled and registered images and the HR target image. Then, this result is compared to the SSIM metric between the HR target and the SR output image. A higher result in this last calculus means that the SR image has a better quality.

3.5. <u>Network Training</u>

The training process is used to define and calculate the weights, bias and activation function parameters of the CNN with the aim to achieve the desired network behavior and to well-fit the network.

During training, the model training loss is computed in each epoch. From this prediction, the parameters and weights are adjusted with the aim to minimize the error. The desired network training behavior is accomplished when the acquired parameters minimize the loss function.

The algorithm used in the training process is called Backpropagation. It is the method in which the parameters are updated before starting a new epoch to minimize the loss.

Adaptative Momentum Estimation Algorithm (Adam) [19] has been the optimizer chosen for the CNN training process. It is a combination of the gradient descent with momentum algorithm and the Root Mean Square (RMS) Prop algorithm. It calculates an exponential weighted moving average of the gradient and then squares the calculated gradient. Two decay parameters which control the decay rates of the moving averages are used by this algorithm. The main arguments used to calculate the Adam optimizer are the loss function, the model parameters and the learning rate which will be reduced during the training process.

The Adam Optimization algorithm is employed for training, with momentum parameters β_1 = 0.9, β_2 = 0.999, and $\varepsilon = 10^{-8}$.

While training, hyperparameters are used. These are the parameters which are defined before the network training process and its main function is to help the algorithm to converge and find the minimum quicker.

The most important hyperparameter while the network is training is called Learning Rate. Learning Rate is the step taken in the gradient method after each backpropagations step. It is crucial to define well the Learning Rate to achieve the convergence of the problem. If a low Learning Rate is used, it will take a lot of iterations to achieve the minimum loss (convergence), if it is finally achieved. A Learning Rate very high will finalize with a non-convergence of the problem. These differences are shown in Figure 3.4.

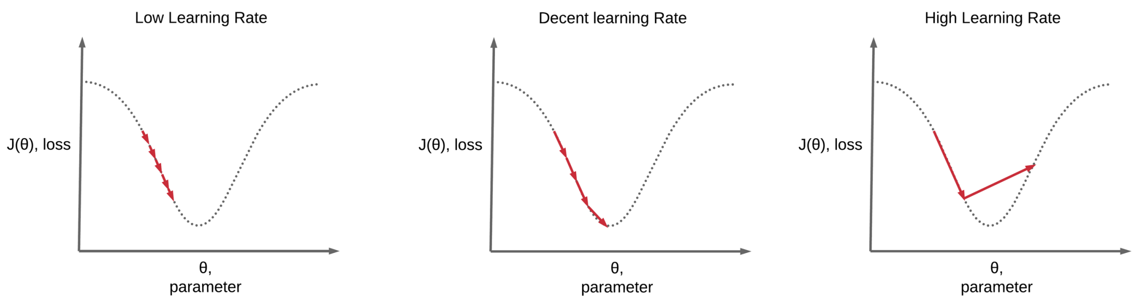


Figure 3.4. Representation of the chosen LR during training process. (2019). [Photo]. Learning Rate Scheduling. https://www.deeplearningwizard.com/deep_learning/boosting_models_pytorch/lr_scheduling/

Another hyperparameter to take into account while setting up the training process is the number of Epochs. This parameter reflects the number of times the whole training data is shown to the network while training.

The Batch Size is the hyperparameter which reflects the number of data sub samples given to the network after which parameter update happens. The bigger the batch size is, the higher will be the computational cost, but the easier will be the acquisition of the loss convergence to obtain the optimal solution.

While the training process is running, a validation process is simultaneously done with the aim to prove and identify if the training is being successful or not. The validation dataset is made from a 10% of the images used in the training process.

It is useful to see the evolution of the loss and the metrics used to define if the training process is underfitted, overfitted or good fitted.

Underfitting may happen when the CNN is being trained with less data than the required to achieve good results. When a new type of input data is used, the network is not capable to recognize it. It is not able to generalize the knowledge and also performs poorly on the training set.

Overfitting may happen when the CNN is only learning specific cases and it will be incapable to recognize new input dates. To solve it, it is useful to introduce while training, corrupted, noisy or non-common data.

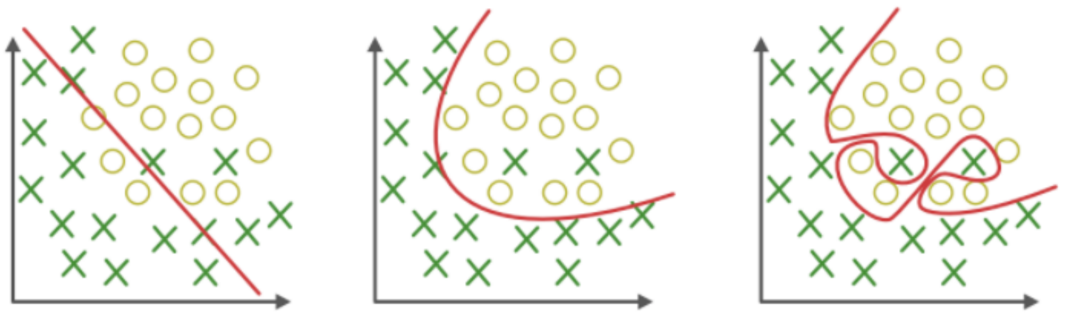


Figure 3.5. Left to right, underfitting, well-fitted and overfitting models. (2019). [Photo]. Underfitting and Overfitting in Machine Learning. https://geeksforgeeks.org/underfitting-and-overfitting-in-machine-learning/

When the training process is finalized, the network is tested to prove and validate its functionality. During this process, a new dataset which the network has not seen before is used as input. Moreover, to check the correct operation of the network, PSNR and SSIM metrics are computed during this last step, called network testing. They are compared with the results of these metrics obtained in the validation process during the training to see if the Deep Neural Network has been well trained or not.

3.6. Data Normalization and Standardization

Techniques such as Data Normalization [34] and Standardization [35] are often used while imagery datasets are being prepared with the aim to improve the network learning process.

Data Normalization [34] means to change the values of each pixel of the images with the aim to use a known scale, without distorting the difference between the interval of values or losing information. In this project, Data Normalization in a scale between 0 and 1 was available to be applied. In particular, the value of each image pixel was divided by the 2^{15} -1, which was the maximum possible value acquired by a pixel, since each pixel of the images has a depth of 16 bits.

Data Standardization [35] is the process of rescaling the imagery pixels values, so, the images have a mean value of 0 and a standard deviation of 1. Standardization assumes that data has a Gaussian distribution, which will make the technique more effective. Standardization is useful when data has varying scales and the algorithm used does make assumptions about the data is having a Gaussian distribution.

3.7. Software and hardware specifications

The computer used to perform all the experiments was the server provided by GPI (gpu and cpu), a regular laptop (just cpu) and Google Colab (gpu). Tensorboard software was used in Google Colab.

4. Experimental Results

In this section we will present the results of our MISR model with 5 LR input images. Then we will compare these results with other configurations of the model using less LR input images (3 and 4).

4.1. Calibration Process

To obtain the final results, a network calibration process was done with the aim of choosing the hyperparameters which perform better, using a small dataset generated with part of the images that were included into the final dataset. The number of input images was five as it was the amount of initial LR input images which were used in the base model during the final training process.

The first experiment was a comparison between using a standardized, normalized or both, standardized and normalized dataset with the adjusted hyperparameters. For all of these methods the same steps were followed: bicubically-upsampled interpolation, SISRNet subnetwork, FusionNet subnetwork, and finally, addition of the averaged registered and bicubically-upsampled interpolation image with the residual image which the FusionNet gives as output.

To perform this study, a large variety of hyperparameters combinations were proved, and finally the best option was a dynamic Learning Rate = 5e-4 which was reduced to the half each 10 epochs. The batch size used was set to 16 and the number of epochs computed was equal to 100.

When the results were obtained, PSNR and SISR metrics were computed to obtain quantitative results.

In the Table 4.1., the main results of the experiment are shown:

DATASET/RESULTS	*TEST INPUT PSNR*	*TEST OUTPUT PSNR*	*TEST SSIM*
NORMALIZED DATASET	*25.96 dB*	*23.68 dB*	*0.73*
STANDARDIZED DATASET	*31.24 dB*	*31.89 dB*	*0.67*
NORMALIZED AND STANDARDIZED DATASET	*31.24 dB*	*31.89 dB*	*0.68*

Table 4.1. Calibration Test Results for Normalized, Standardized and Normalized+Standardized Dataset. (Test Input PSNR: averaged input images, target; Test Output PSNR: predicted SR image, target; Test SSIM: predicted SR image, target).

To make a better normalization method selection, it is worthen to study the training and validation process of each experiment. With the aid of the Tensorboard software, it was possible to plot the train and validation losses and the metrics studied during training. In Figure 4.1., Figure 4.2., Figure 4.3., results are shown.

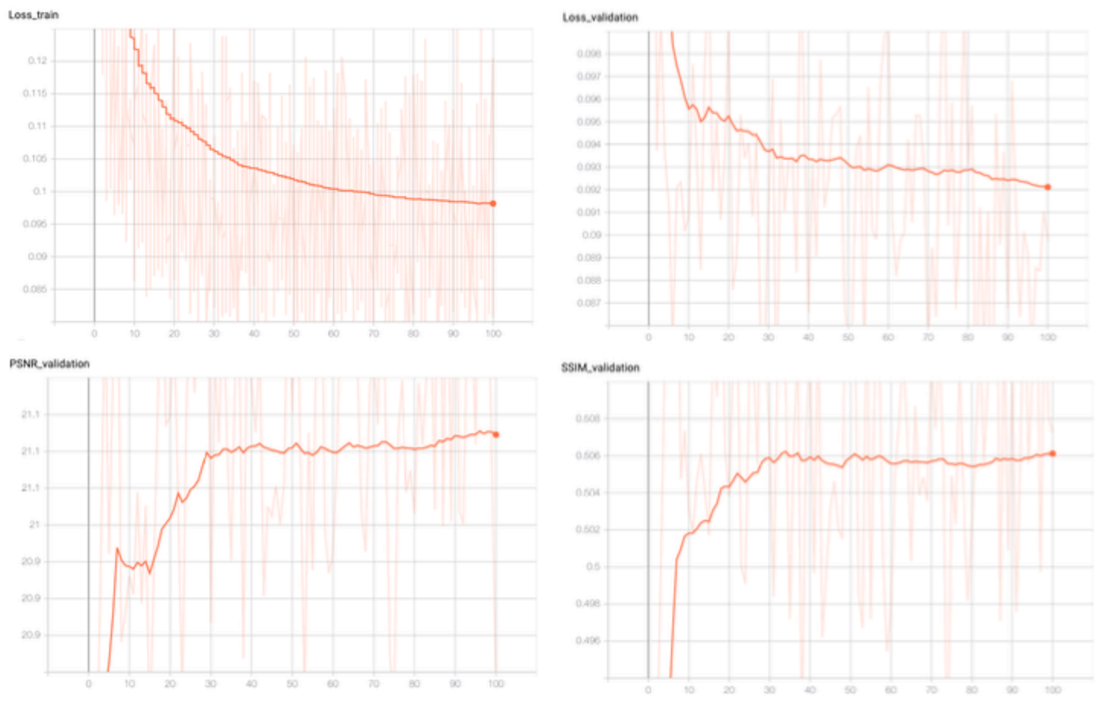


Figure 4.1. Left to right, upon to bottom; Normalized Dataset Training Loss, Validation Loss, PSNR and SSIM graphs. (horizontal axis represents epochs / vertical axis represents dB for the loss graphs).

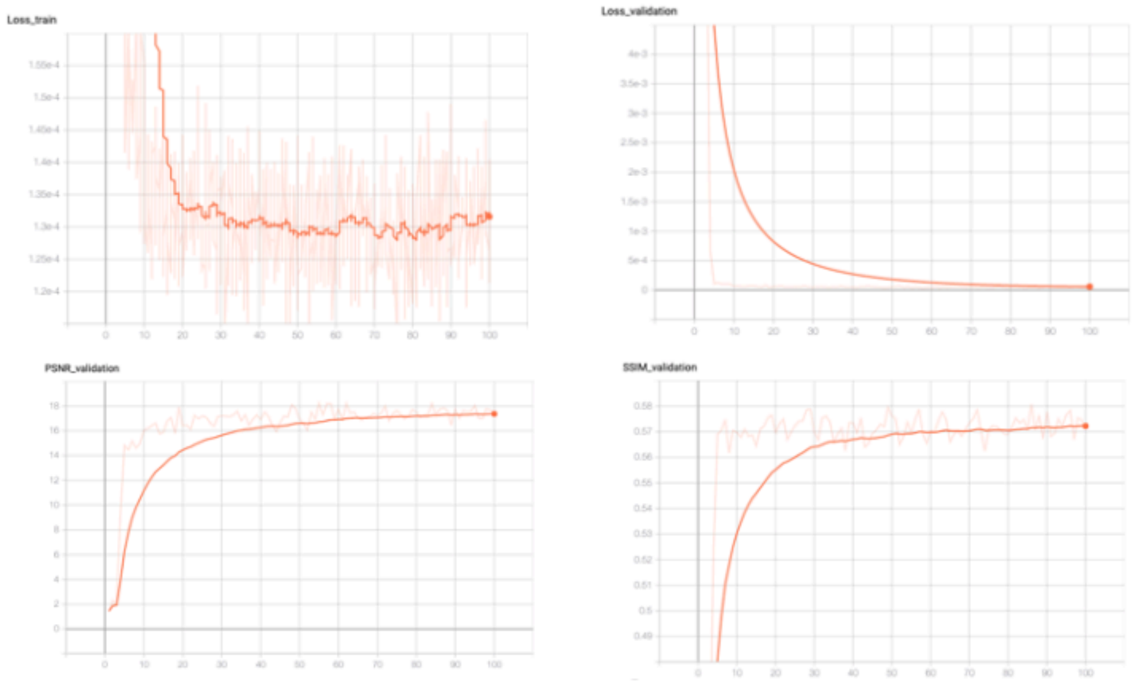


Figure 4.2. Left to right, upon to bottom; Standardized Dataset Training Loss, Validation Loss, PSNR and SSIM graphs. (horizontal axis represents epochs / vertical axis represents dB for the loss graphs).

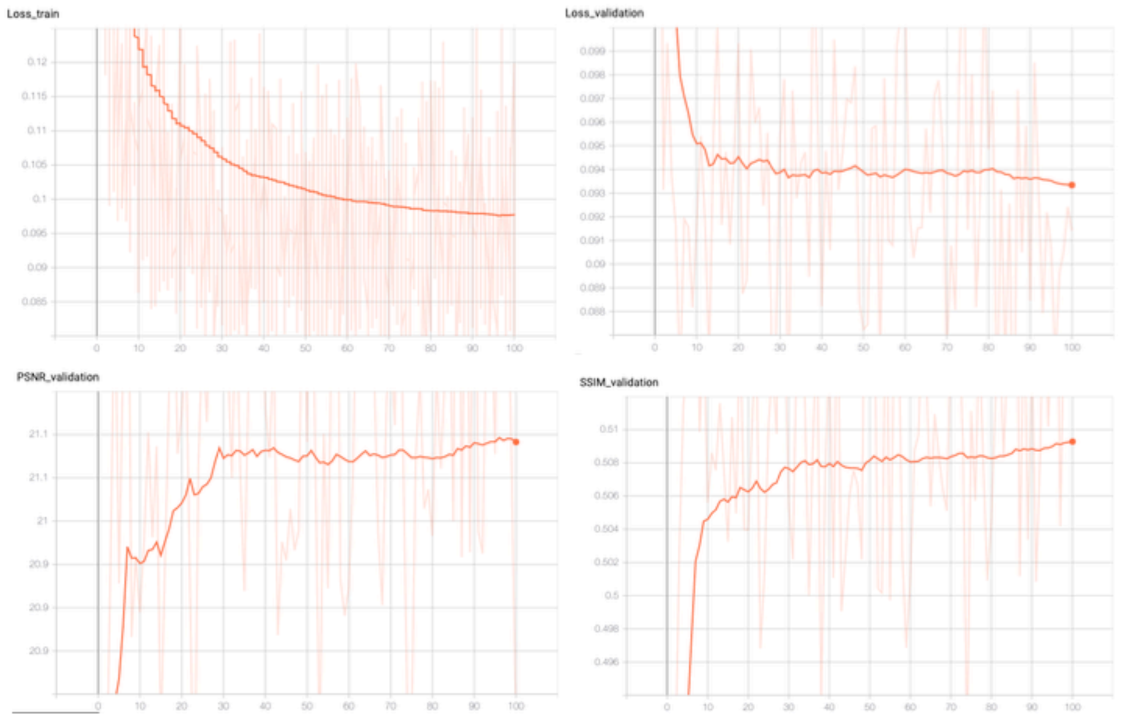


Figure 4.3. Left to right, upon to bottom; Normalized+Standardized Dataset Training Loss, Validation Loss, PSNR and SSIM graphs. (horizontal axis represents epochs / vertical axis represents dB for the loss graphs).

The first thing which was reflected in this first study was that the worst normalization method was when the network used a Normalized Dataset due to a lower PSNR test metric compared to the ones obtained by the other two methods.

Comparing the last two datasets initializations, it was seen that while testing the network, both results were similar. However, taking a close look onto the validation process, we could see a slight improvement in the validation loss when a standardized dataset was used. Validation metrics were also similar for both cases. Moreover, taking a look at the visual results, it was seen that when the dataset was only standardized, images presented a good appearance.

So, finally, we decided to use the standardized dataset to tackle the main project goals and train the final Deep Neural Networks.

Figure 4.4. shows one scene of the results obtained with the last network calibration with a standardized dataset and its relative PSNR and SSIM metrics.

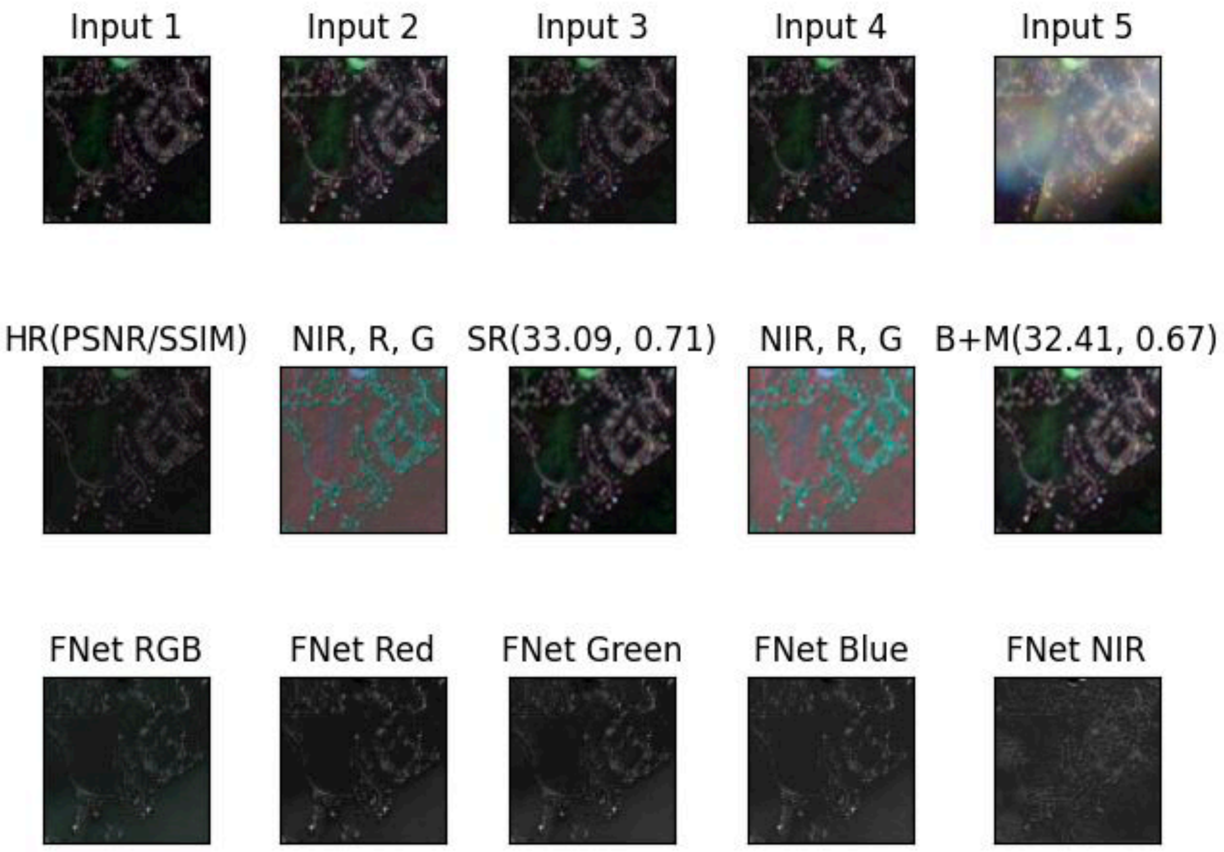


Figure 4.4. Left to right per rows, First Row: Five multitemporal low-resolution input images, Second Row: HR target, False Color target, SR output image, False Color SR image, Bicubically interpolated+Mean of the five LR inputs, Third Row: Fusion Net output images in different spectral bands.

Once achieved this first objective, it was time to set up the final dataset and make the final experiments. The final training was run with the hyperparameters used in the calibration process.

HYPERPAREMETER	*VALUE*
DYNAMIC LEARNING RATE	*5e-4 with a decrease to the half each 10 epochs*
EPOCHS	*100*
BATCH SIZE	*16*

Table 4.2. Setting up of the calibration training hyperparameters.

Despite using the hyperparameters which performed better in the calibration training process, a large and iterative process to achieve better results and improve the functionality of the CNN had to be done when the final dataset was used. These initial hyperparameters had to be resized, in particular, the learning rate was changed several times until the network achieved the desired results.

To tackle one of the objectives of this project, which was to compare the results obtained when different number of LR input images were used to train Deep Neural Networks using Sentinel-2 imagery, the training experiments for 5, 4 and 3 multitemporal LR input images were done.

In the chapters 4.2, 4.3 and 4.4, the results of these experiments are shown.

4.2. Deep Neural Network for Super-Resolution of 5 Multitemporal Remote Sensing Images

As it was mentioned in the previous chapter, to achieve the final results, some hyperparameters were modified to achieve better quantitative and qualitative results.

The hyperparameters used to train the final 5 LR input images network are shown in the Table. 4.3.

HYPERPAREMETER	*VALUE*
DYNAMIC LEARNING RATE	*5e-1 with a decrease to the half each 20 epochs*
EPOCHS	*100*
BATCH SIZE	*16*

Table 4.3. Setting up of the hyperparameters when 5 input LR images are used to train the network.

Visually, obtained results when 5 multitemporal remote sensing LR images were used as input of the CNN are shown in Figure 4.5.

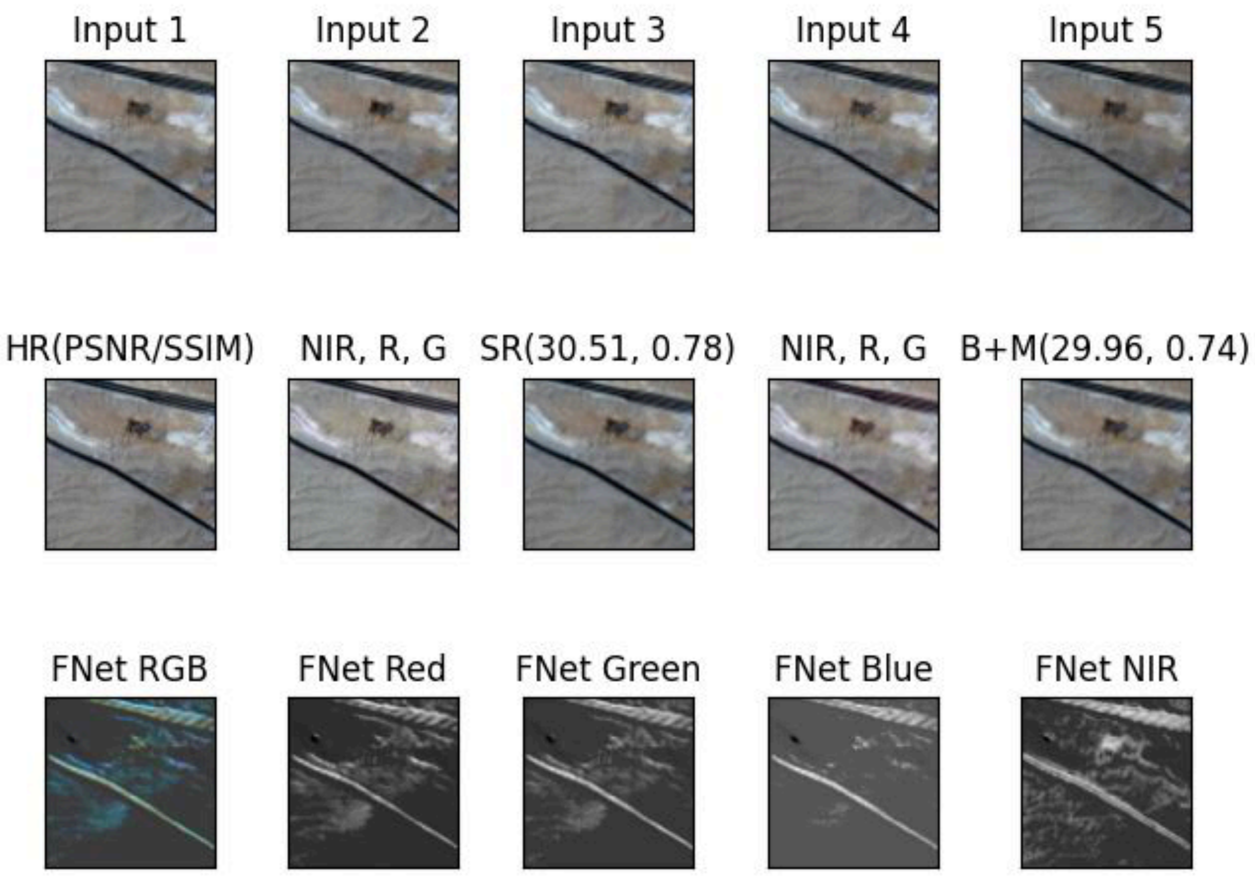


Figure 4.5. Left to right per rows, First Row: Five multitemporal low-resolution input images, Second Row: HR target, False Color target, SR output image, False Color SR image, Bicubically interpolated+Mean of the five LR inputs, Third Row: Fusion Net output images in different spectral bands.

Taking a look at the SR predicted image and comparing it with the average input bicubically-upsampled and registered image, we can notice an increase of resolution and also a higher value of the PSNR and SSIM metrics. That means that a higher qualitative predicted SR image was obtained because of a well-fitted training process. Moreover, the predicted SR image is sharper and with more quality than the LR input images. Furthermore, visualizing the FusionNet output image, we are able to know what the CNN was exactly learning from the input LR images.

In addition, with the Tensorboard software, it is possible to plot the validation training process. In Figure 4.6. the correspondent graphs of the training and validation process are shown.

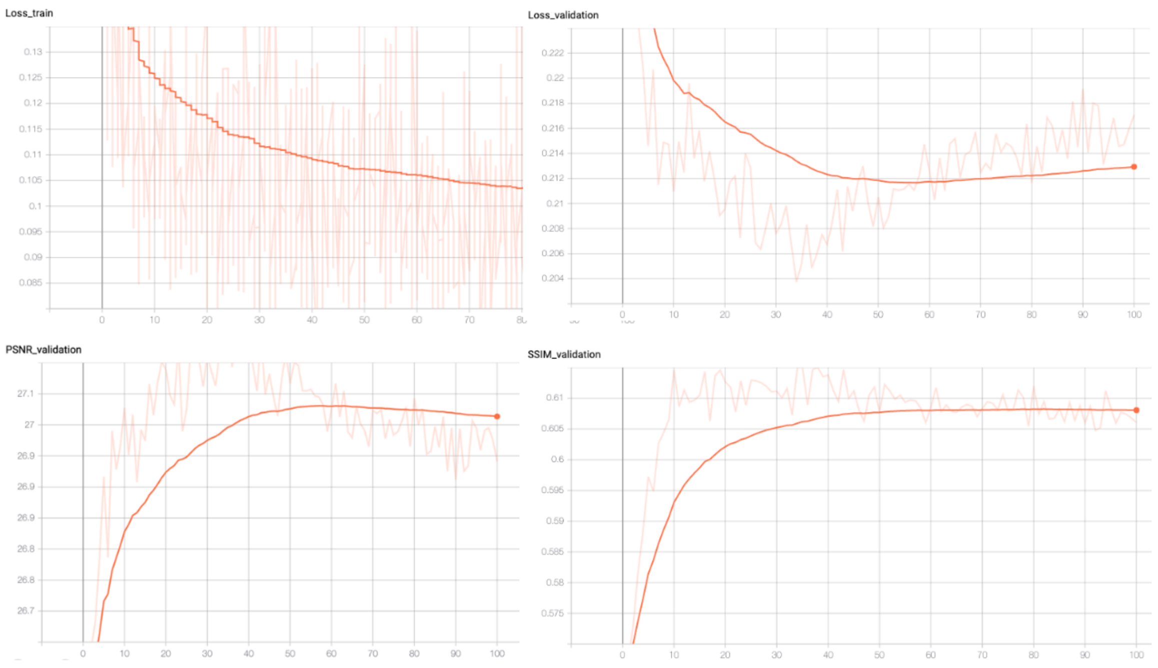


Figure 4.6. Left to right, upon to bottom; Training Loss, Validation Loss, PSNR and SSIM graphs. (horizontal axis represents epochs / vertical axis represents dB for the loss graphs).

In these pictures, it is possible to observe that the Deep Neural Network was training adequately but from epoch 60, the network starts to overfit. That means that maybe better hyperparameters could have been set to better fit the network during the training process and a higher number of epochs should have been. Moreover, a higher number of images should have been used to train the CNN. Nevertheless, at this point of the project, this is the best result achieved.

4.3. Deep Neural Network for Super-Resolution of 4 Multitemporal Remote Sensing Images

In this chapter, the results obtained when 4 LR input images were used during the network training process are presented.

We have to keep in mind that hyperparameters used were the ones which had given better results while training the Deep Neural Network with 5 LR input images. It meant that in this second experiment, the hyperparameters had not been modified. The purpose of doing that was to be able to find dependences between networks training process and knowing how they will react when a different amount of LR input images were used.

The visual results obtained when 4 multitemporal remote sensing LR input images were used are shown in Figure 4.7.

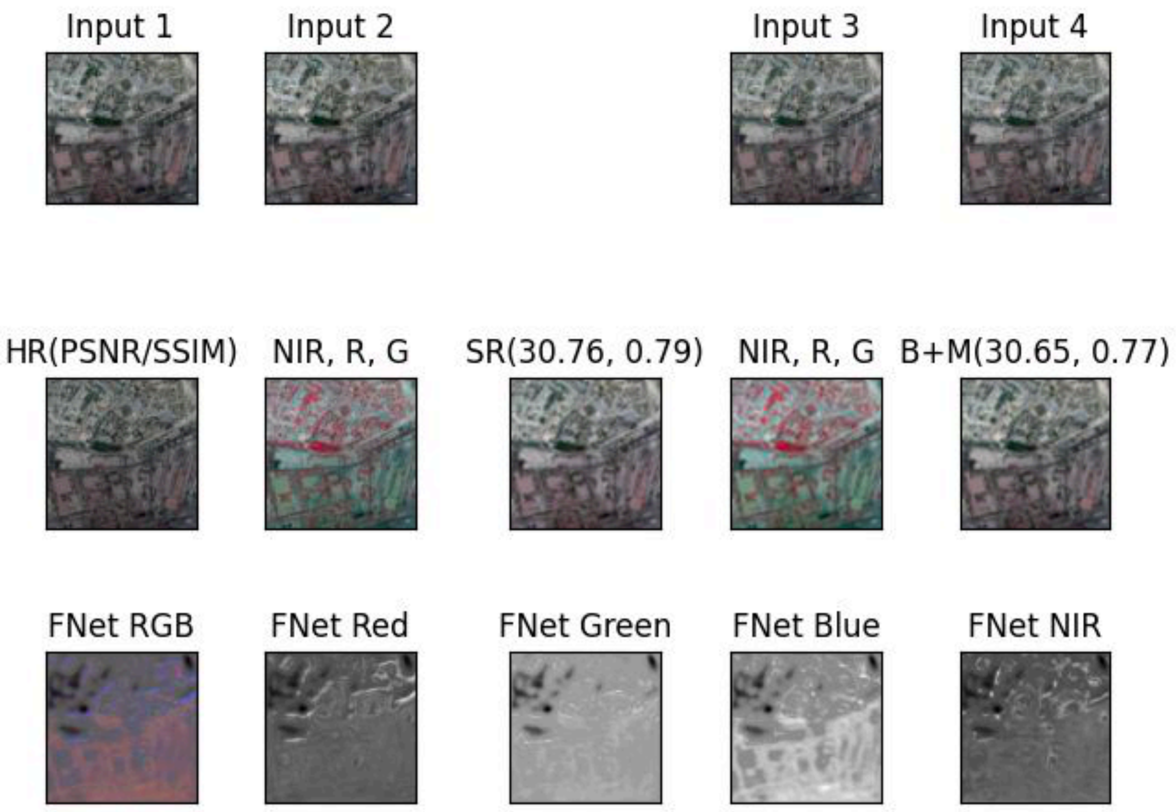


Figure 4.7. Left to right per rows, First Row: Four multitemporal low-resolution input images, Second Row: HR target, False Color target, SR output image, False Color SR image, Bicubically interpolated+Mean of the four LR inputs, Third Row: Fusion Net output images in different spectral bands.

Apparently, the predicted SR image seems to have a better quality compared with the averaged bicubically-upsampled and registered image. Moreover, when the PSNR and the SSIM metrics between these two pictures are compared, a qualitative improvement was observed. These metrics were better in the predicted SR image. It meant that using 4 LR images as input and the same hyperparameters and configuration used when 5 LR input images are used, the Deep Neural Network was still working well and it also produced output good results.

Despite that, it was also important to take care of the training and validation process to see exactly the behavior of the network during training. In the Figure 4.8. the training and validation curves are shown.

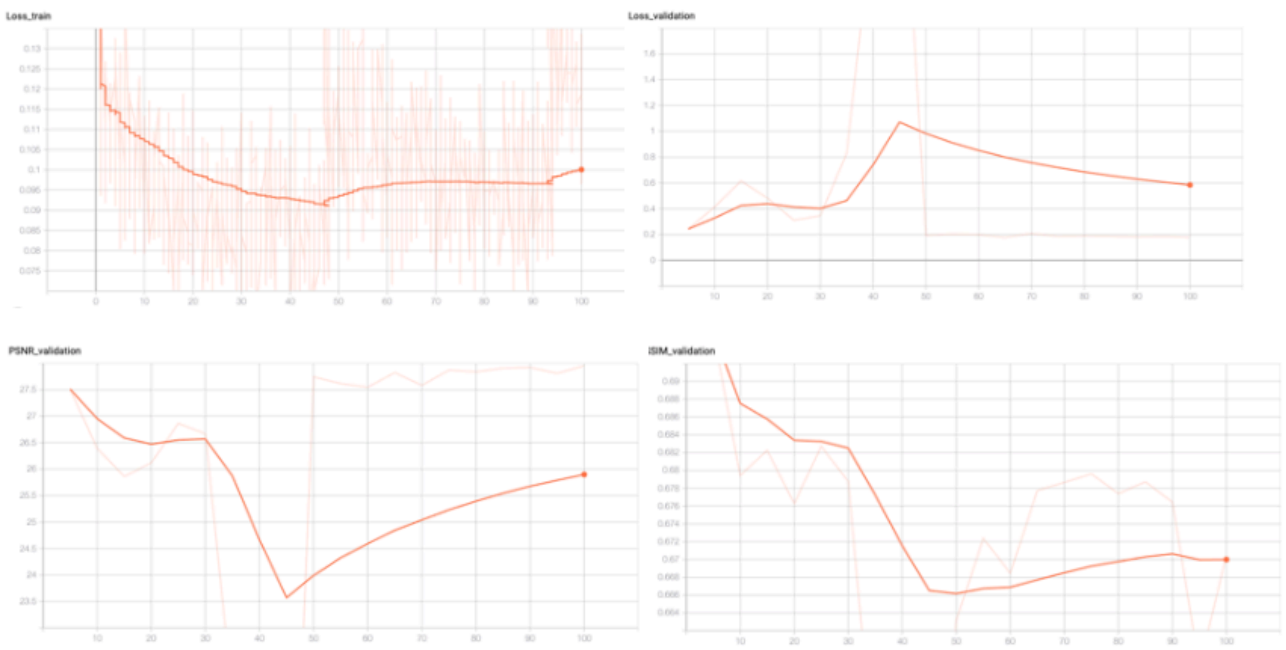


Figure 4.8. Left to right, upon to bottom; Training Loss, Validation Loss, PSNR and SSIM graphs. (horizontal axis represents epochs / vertical axis represents dB for the loss graphs).

During the training process, the training loss was decreasing until epoch 50 approximately, and then this value increased progressively but not a lot. So, it seems that the network was working well because the training loss was decreasing in each step. In the validation process, we observe the opposite behavior. Validation loss and metrics seemed to do not work well, but, from epoch 50 all of them made a turn in their trajectories going towards the good direction.

4.4. Deep Neural Network for Super-Resolution of 3 Multitemporal Remote Sensing Images

In this section, the results obtained with 3 LR input images are presented.

One example of the results obtained with 3 multitemporal images is shown in the Figure 4.9.

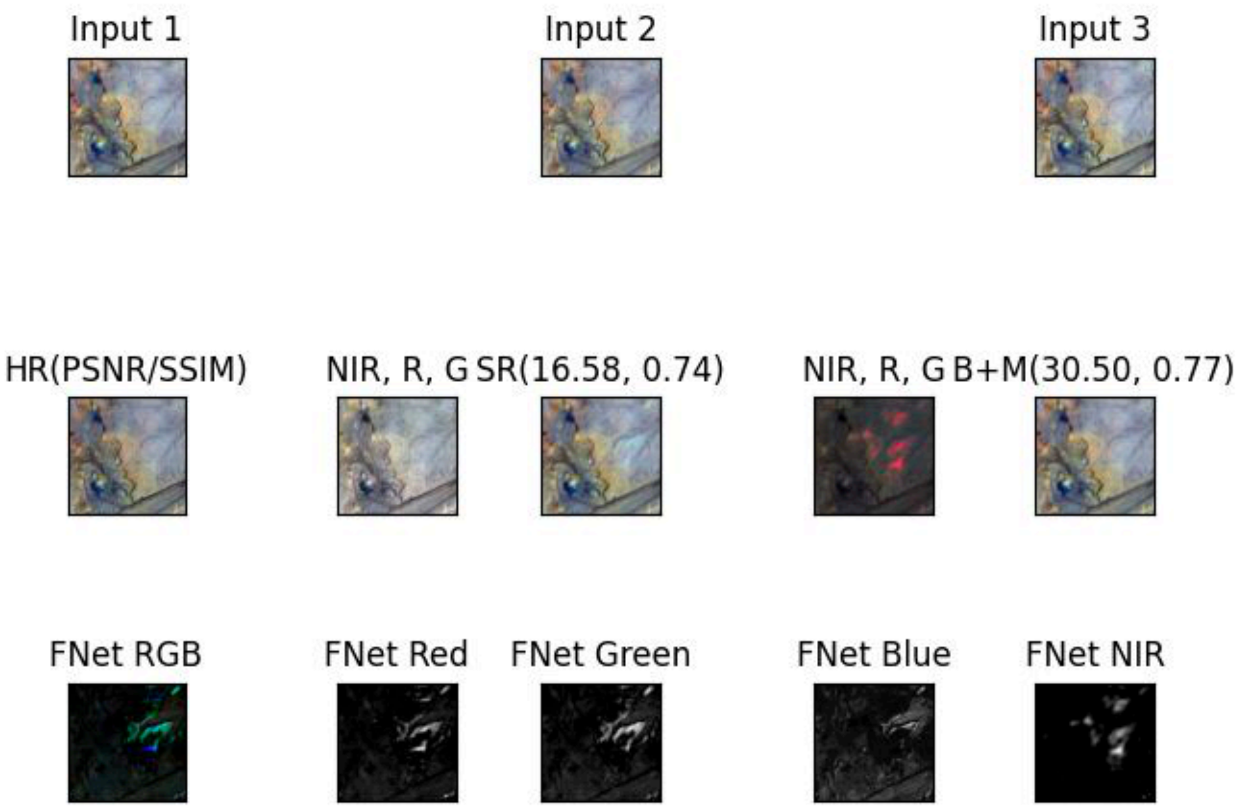


Figure 4.9. Left to right per rows, First Row: Three multitemporal low-resolution input images, Second Row: HR target, False Color target, SR output image, False Color SR image, Bicubically interpolated+Mean of the three LR inputs, Third Row: Fusion Net output images in different spectral bands.

Definitely, this last experiment, in which 3 LR images were used as input, was the one that produced the worst results. It has to be taken into account that the CNN was trained using the same configuration and hyperparameters which were used to train the base model (the one with 5 LR input images). As it is shown in the Figure 4.9., the SR predicted image has worse PSNR and SSIM metrics than the averaged bicubically-upasampled and registered one. The predicted SR image obtained PSNR and SSIM metrics approximately equal to half of the values that the averaged image, which has not been processed by the CNN, got. Moreover, the quality of the predicted SR image was visually worse than the input images used to train the Deep Neural Network. The obtained SR image is blurrier than the input ones.

The training and validation curves were used to finally understand the bad behavior of the network in this particular experiment. In the Figure 4.10. this behavior is shown.

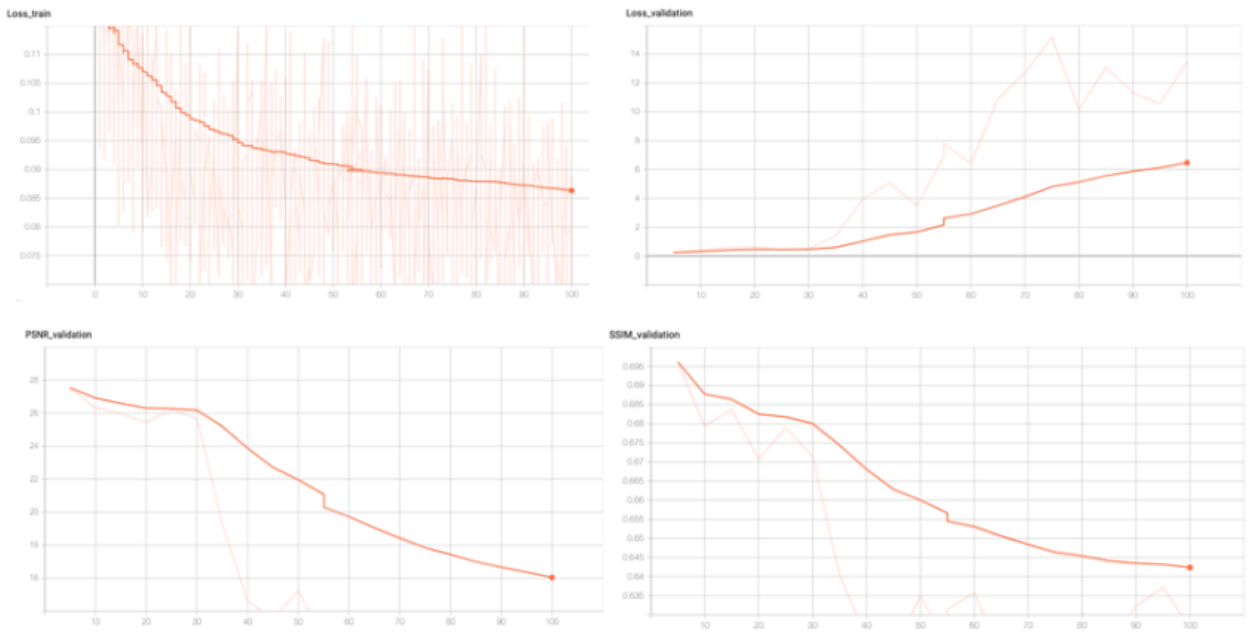


Figure 4.10. Left to right, upon to bottom; Training Loss, Validation Loss, PSNR and SSIM graphs. (horizontal axis represents epochs / vertical axis represents dB for the loss graphs).

At first, looking only at the training loss curve, it might seem that the CNN was training well but when the validation process is observed, it is possible to appreciate that the network had a very bad training process. Validation loss was increasing rapidly and PSNR and SSIM metrics were reducing their values. This meant that, on the one hand, the hyperparameters chosen to run this experiment were not the appropriates and, on the other hand, it also meant that the behavior of the network was not the desired and it was doing a very bad training. Unfortunately, we did not have more time to retrain the model with a different set of hyperparameters.

To sum up, we can compare the test results for all the experiments to be able to reach the final conclusions. In Table 4.4. these test results are shown.

DATASET/RESULTS	***TEST INPUT PSNR***	***TEST OUTPUT PSNR***	***TEST INPUT SSIM***	***TEST OUTPUT SSIM***
5 LR INPUT IMAGES	*29.75 dB*	*30.31 dB*	*0.73*	*0.78*
4 LR INPUT IMAGES	*30.45 dB*	*30.55 dB*	*0.77*	*0.78*
3 LR INPUT IMAGES	*30.43 dB*	*16.62 dB*	*0.77*	*0.74*

Table 4.4. Comparative table of the test results when a different amount of input images were used. (Test Input PSNR: averaged input images, target; Test Output PSNR: predicted SR image, target; Test Input SSIM: averaged input images, target; Test Output SSIM: predicted SR image, target).

Test results clearly demonstrate that with 5 LR input images, the network acquires better qualitative results. In this case, the difference between output and input metrics is higher than the difference between these metrics when 4 and 3 LR images are used. It means that a qualitative improvement is achieved using 5 LR input images.

When 4 LR images are used, the Deep Neural Network also works well, acquiring an improvement in the output metrics. Nevertheless, the quantitative and qualitative results are not as good as the ones obtained in the 5 LR input images case.

The worst case is produced with 3 LR input images. Taking a look in the test results, it is being able to see that output metrics are worse than input metrics. It means that the network does not work correctly. To solve this bad functionality of the Deep Neural Network, when 3 LR input images are used, a different configuration and hyperparameters have to be set to fit well the CNN.

5. Budget

The main goal of this section is to estimate the budget of the project taking into account that it is a research project and therefore, it does not involve a service or a product to be sold.

The SNAP software from ESA used to develop and visualize data is open-source. Nevertheless, the software used to manage data (ENVI) is not free. The annual licence fee is approximately 220 €.

The hardware needed for the development of this project was the computational resources provided by the GPI and the personal laptop used to work on the final thesis. The total estimation of the hardware is the cost of using the server and the purchase of the personal computer which is mandatory needed to work in the project. The monthly server use license has a cost of 50 € monthly and the laptop has a market price of 2500€.

Moreover, it has to be taken into account the salary of the member which will work and develop the project. Doing an estimation of the amount of time that each member of the project has destinated to do it and the standard salary for junior engineers (15€/hour, 20 hours a week), and two technical advisors (30€/hour, 2 hours a week); the costs are able to be computed as follows:

ITEM	PRICE	TIME	TOTAL (4 MONTHS)
SOFTWARE			
ENVI SOFTWARE	50€/month	all	200€
HARDWARE			
SERVER COMPUTATION	50€/month	all	200€
COMPUTER	2500€	all	2500€
WORKERS			
JUNIOR ENGINEER	15€/hour	20 hours/week	4800€
2 TECHNICAL ADVISORS	30€/hour	2 hours/week	2 x 960€ = 1920€

Table 5.1. Budget.

The final budget is:

200€ + 200€ + 2500€ + 4800€ + 1920€ = **9620€**

6. Conclusions and future development:

This work presents a Deep Neural Network architecture, in particular, a CNN to deal with super-resolution applied to multi-temporal images.

The results obtained with the base Deep Neural Network, the one which had as input 5 LR images, were successful but not optimal at all. It demonstrated that fusing multiple images from the same ground scene permits acquiring images with higher spatial resolution, despite the presence of relevant temporal variations.

In this work, it was not necessary to deal with images registration because during the selection of the dataset images, we choose the ones that each pixel of the images approximately corresponds to the same pixels of the other temporal images. The fact that the images in the dataset were as different as possible, in visual terms, made the training process more complex and complicated. Nevertheless, it leaves more room for improvement.

The comparison of the results obtained when the CNN had 5 LR input images with the other two that had 4 and 3 LR input images, using in all cases the same network configuration and hyperparameters, have been useful to understand the importance of having a certain amount of temporal input data and also to get conscious of the importance of well calibrating the model.

When 4 LR input images were used, it was seen that the network did not train as well as in the 5 images case, but the test results obtained during the network test were very similar. Nevertheless, this is not accomplished when 3 LR input images were used to train the network. In this case, an important deterioration in the training process and in the test results have been occurred. To sum up, to get good accuracy and better CNN results, each network has to be trained independently with a particular configuration and hyperparameters.

Future work could investigate an enhanced solution in which the images were registered inside the CNN architecture in order to be able to use datasets which have not been created taking into account the registration task among images. Moreover, several experiments using Adversarial Approach [36] or RRDB blocks can be proved with the aim of improving the final Deep Neural Network.

Bibliography:

[1] A. B. Molini, D. Valsesia, G. Fracastoro and E. Magli, "Deep Learning For Super-Resolution Of Unregistered Multi-Temporal Satellite Images," 2019 10th Workshop on Hyperspectral Imaging and Signal Processing: Evolution in Remote Sensing (WHISPERS), Amsterdam, Netherlands, 2019, pp. 1-5, doi: 10.1109/WHISPERS.2019.8920910.

[2] A. Bordone Molini, D. Valsesia, G. Fracastoro and E. Magli, "DeepSUM: Deep Neural Network for Super-Resolution of Unregistered Multitemporal Images," in IEEE Transactions on Geoscience and Remote Sensing, vol. 58, no. 5, pp. 3644-3656, May 2020, doi: 10.1109/TGRS.2019.2959248.Arxiv

[3] Geodim.es. 2020. SM GEODIM MODELOS DE INFORMACIÓN DE LA TIERRA SENTINEL 2. [online] Available at: <http://www.geodim.es/pdf/Geodim%20SENTINEL-2A.pdf>.

[4] Sentinel.esa.int. 2020. Orbit - Sentinel 2 - Mission - Sentinel Online. [online] Available at: <https://sentinel.esa.int/web/sentinel/missions/sentinel-2/satellite-description/orbit>.

[5] W. Yang, X. Zhang, Y. Tian, W. Wang, J. Xue and Q. Liao, "Deep Learning for Single Image Super-Resolution: A Brief Review," in IEEE Transactions on Multimedia, vol. 21, no. 12, pp. 3106-3121, Dec. 2019, doi: 10.1109/TMM.2019.2919431.Asdfdas

[6] M. K. Ng, H. Shen, E. Y. Lam, and L. Zhang, "A total variation reg- ularization based super-resolution reconstruction algorithm for digital video," EURASIP Journal on Advances in Signal Processing (JASP), vol. 2007, no. 1, p. 074585, 2007.

[7] J. Sun, Z. Xu, and H.-Y. Shum, "Gradient profile prior and its applica- tions in image super-resolution and enhancement," IEEE Transactions on Image Processing (TIP), vol. 20, pp. 1529–42, 11 201

[8] L. Wang, S. Xiang, G. Meng, H. Wu, and C. Pan, "Edge-directed single-image super-resolution via adaptive gradient magnitude self- interpolation," IEEE Transactions on Circuits and Systems for Video Technology (TCSVT), vol. 23, no. 8, pp. 1289–1299, Aug 2013.

[9] M. Protter, M. Elad, H. Takeda, and P. Milanfar, "Generalizing the nonlocal-means to super-resolution reconstruction," IEEE Transactions on Image Processing (TIP), vol. 18, no. 1, pp. 36–51, Jan 2009.

[10] K. Zhang, X. Gao, D. Tao, and X. Li, "Single image super-resolution with multiscale similarity learning," IEEE Transactions on Neural Networks and Learning Systems (TNNLS), vol. 24, pp. 1648–1659, 2013.

[11] GitHub. 2020. Diegovalsesia/Deepsum. [online] Available at: <https://github.com/diegovalsesia/deepsum>.

[12] R. Tsai and T. Huang, “Multiframe image restoration and registration,” in Advances in Computer Vision and Image Processing, vol. I. JAI Press: Greenwich, CT, USA, 1984, pp. 317–339.

[13] M. Irani and S. Peleg, “Improving resolution by image registration,” Graphical Models and Image Processing (CVGIP), vol. 53, no. 3, pp. 231 – 239, 1991.

[14] H. Stark and P. Oskoui, “High-resolution image recovery from image plane arrays, using convex projections,” Journal of the Optical Society of America, vol. 6, no. 11, pp. 1715–1726, Nov 1989.

[15] A. J. Patti, M. Ibrahim Sezan, and A. Murat Tekalp, “High-resolution image reconstruction from a low-resolution image sequence in the presence of time-varying motion blur,” in Proceedings of 1st International Conference on Image Processing (ICIP), vol. 1, Nov 1994, pp. 343–347 vol.1.

[16] S. Farsiu, M. D. Robinson, M. Elad, and P. Milanfar, “Fast and robust multiframe super resolution,” IEEE Transactions on Image Processing (TIP), vol. 13, no. 10, pp. 1327–1344, Oct 2004.

[17] H. Shen, M. K. Ng, P. Li, and L. Zhang, “Super-resolution reconstruction algorithm to modis remote sensing images,” The Computer Journal, vol. 52, pp. 90–100, 01 2009.

[18] Cs231n.github.io. 2020. Cs231n Convolutional Neural Networks For Visual Recognition. [online] Available at: <https://cs231n.github.io/convolutional-networks/#overview> .

[19] Brownlee, J., 2020. Gentle Introduction To The Adam Optimization Algorithm For Deep Learning. [online] Machine Learning Mastery. Available at: <https://machinelearningmastery.com/adam-optimization-algorithm-for-deep-learning/> .

[20] Medium. 2020. A Practical Guide To Relu. [online] Available at: <https://medium.com/@danqing/a-practical-guide-to-relu-b83ca804f1f7>.

[21] Aprende Machine Learning. 2020. Qué Es Overfitting Y Underfitting Y Cómo Solucionarlo. [online] Available at: <https://www.aprendemachinelearning.com/que-es-overfitting-y-underfitting-y-como-solucionarlo/>.

[22] Semiconductor Engineering. 2020. Peak Signal-To-Noise Ratio As An Image Quality Metric. [online] Available at: <https://semiengineering.com/peak-signal-to-noise-ratio-as-an-image-quality-metric/>.

[23] Imatest.com. 2020. SSIM: Structural Similarity Index | Imatest. [online] Available at: <https://www.imatest.com/docs/ssim/>.

[24] ESA SNAP software (7.0.0). (2018). [SNAP comes from the acronym Sentinel Application Platform and responds to a free program offered by the European Space Agency to process and analyze satellite images from the Sentinel satellite fleet.]. ESA. <http://step.esa.int>.

[25] D. Valsesia and E. Magli, "A novel rate control algorithm for onboard predictive coding of multispectral and hyperspectral images," IEEE Transactions on Geoscience and Remote Sensing (TGRS), vol. 52, no. 10, pp. 6341–6355, Oct 2014.

[26] D. Valsesia and P. T. Boufounos, "Universal encoding of multispectral images," in IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), March 2016, pp. 4453–4457.

[27] J. Yang, J. Wright, T. S. Huang, and Y. Ma, "Image super-resolution via sparse representation," IEEE Transactions on Image Processing (TIP), vol. 19, no. 11, pp. 2861–2873, Nov 2010.

[28] R. Zeyde, M. Elad, and M. Protter, "On single image scale-up us- ing sparse-representations," in Curves and Surfaces, J.-D. Boissonnat, P. Chenin, A. Cohen, C. Gout, T. Lyche, M.-L. Mazure, and L. Schu- maker, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 711–730.

[29] R. Timofte, V. De, and L. V. Gool, "Anchored neighborhood regression for fast example-based super-resolution," in IEEE International Confer- ence on Computer Vision (ICCV), Dec 2013, pp. 1920–1927.

[30] S. Schulter, C. Leistner, and H. Bischof, "Fast and accurate image up- scaling with super-resolution forests," in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2015, pp. 3791–3799.

[31] W. T. Freeman, T. R. Jones, and E. C. Pasztor, "Example-based super- resolution," IEEE Computer Graphics and Applications, vol. 22, no. 2, pp. 56–65, March 2002.

[32] A. Kappeler, S. Yoo, Q. Dai, and A. K. Katsaggelos, "Video super- resolution with convolutional neural networks," IEEE Transactions on Computational Imaging (TCI), vol. 2, no. 2, pp. 109–122, June 2016.

[33] J. Caballero, C. Ledig, A. P. Aitken, A. Acosta, J. Totz, Z. Wang, and W. Shi, "Real-time video super-resolution with spatio-temporal networks and motion compensation," CoRR, vol. abs/1611.05250, 2016.

[34] Docs.microsoft.com. 2020. Normalizar Datos: Referencia Para Los Módulos - Azure Machine Learning. [online] Available at: <https://docs.microsoft.com/es-es/azure/machine-learning/algorithm-module-reference/normalize-data>.

[35] Brownlee, J., 2020. How To Normalize And Standardize Your Machine Learning Data In Weka. [online] Machine Learning Mastery. Available at: <https://machinelearningmastery.com/normalize-standardize-machine-learning-data-weka/>.

[36] Wang, X., Yu, K., Wu, S., Gu, J., Liu, Y., Dong, C., Loy, C., Qiao, Y. and Tang, X., 2020. ESRGAN: Enhanced Super-Resolution Generative Adversarial Networks. [online] arXiv.org. Available at: <https://arxiv.org/abs/1809.00219> [Accessed 29 June 2020].

Appendices:

Appendix A: Work Packages

Project: **Project proposal and workplan**	WP ref: **WP 1**	
Major constituent: Documentation	Sheet 1 of 8	
Short description: Documentation of the first document of the Final Degree Project. This document consists on the basis, project goals, project organization, relevant information…	Planned start date: 09/03/2020 Planned end date: 12/03/2020 Start event: T1 End event: T3	
Internal task T1: Project definition and description **Internal task T2:** Project development plan **Internal task T3:** Document review and approval	Deliverables: Project_Proposal_And_Workplan_TFG_PolMasóAyats.doc	Dates: 12/03/2020

Project: **Information research and documentation**	WP ref: **WP 2**	
Major constituent: Information research	Sheet 2 of 8	
Short description: Learn about the basic recourses which I will have to use to implement the project.	Planned start date: 17/02/2020 Planned end date: 27/03/2020 Start event: T1 End event: T4	
Internal task T1: Do CS231n Stanford course about deep learning and machine learning and the Andrew Ng course about deep learning and CNN from Coursera. **Internal task T2:** Familiarization with Python and the library Pytorch. **Internal task T3:** Familiarization with ENVI/SNAP **Internal task T4:** Do research of Deep neural network for Super-resolution of multitemporal Remote Sensing Images	Deliverables:	Dates:

Project: **Download images to create the images dataset**	WP ref: **WP 3**
Major constituent: Image dataset	Sheet 3 of 8
Short description:	Planned start date: 16/03/2020 Planned end date: 20/03/2020

Downloading images captured by Sentinel-2 and create subsets with the bands 2, 3, 4 and 8, corresponding with the 10 meters spatial resolution.	Start event: T1 End event: T1	
Internal task T1: Check the images to create the dataset.	Deliverables:	Dates:

Project: **Software development**	WP ref: **WP 4**	
Major constituent: Software	Sheet 4 of 8	
Short description: Learning how to use the appropriate software and start its implementation.	Planned start date: 18/03/2020 Planned end date: 22/05/2020 Start event: T1 End event: T2	
Internal task T1: Use transfer learning and internet resources to build an adequate CNN for this project. **Internal task T2:** Use the images in the Sentinel-2 dataset to test the Software developed.	Deliverables:	Dates:

Project: **Critical Review**	WP ref: **WP 5**	
Major constituent: Document	Sheet 5 of 8	
Short description: Documentation of the second document of the Final Degree Project. This document consists in a review of the initial work plan and the actual state of the project. Including modifications and reviews according to the real project evaluation.	Planned start date: 10/04/2020 Planned end date: 14/04/2020 Start event: T1 End event: T2	
Internal task T1: Analyze and review the project state and compare it with the initial work plan. **Internal task T2:** Write and deliver the document.	Deliverables: Project_Critical_Review_TFG_PolMasóAyats.doc	Dates: 14/04/2020

Project: **Test and results assessment**	WP ref: **WP 6**	
Major constituent: Documentation and software	Sheet 6 of 8	
Short description: Get the final results and make upgrades and modifications in the software where it is needed. Compare these results to other approaches done in other Sentinel-2 Projects if it is possible.	Planned start date: 15/05/2020 Planned end date: 05/06/2020 Start event: T1 End event: T3	
Internal task T1: Test the performance of the system with the test images. **Internal task T2:** Fine-tune the results and look for improvements. **Internal task T3:** Get results.	Deliverables:	Dates:

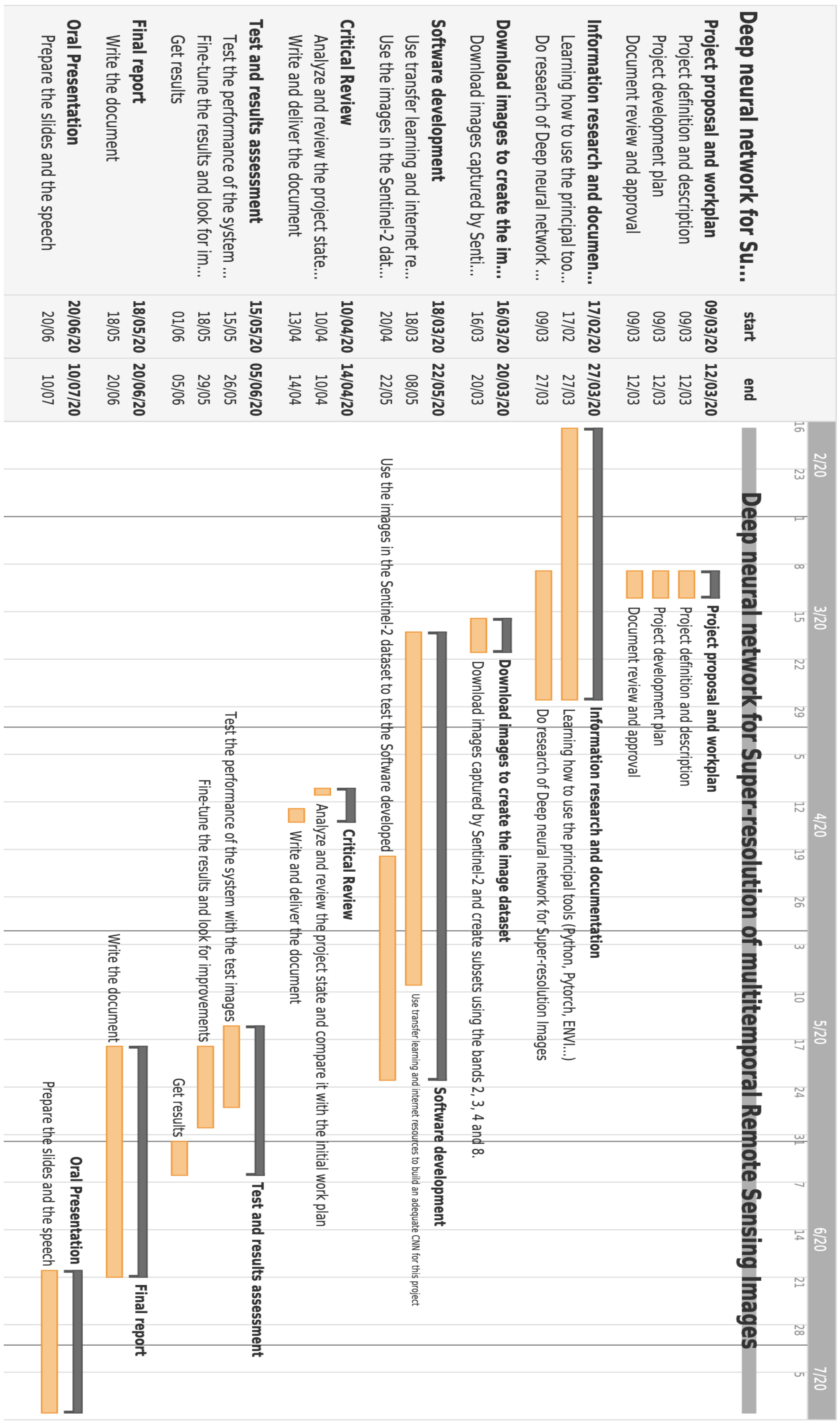
Project: **Final report**	WP ref: **WP 7**	
Major constituent: Documentation	Sheet 7 of 8	
Short description: Documentation of the third document of the Final Degree Project. This document consists on the final results and conclusions. It reflects all the main work done during the project.	Planned start date: 18/05/2020 Planned end date: 20/06/2020 Start event: T1 End event: T1	
Internal task T1: Write the document.	Deliverables: Final_Report_ TFG_PolMasóAyats.doc	Dates: 29/06/2020

Project: **Oral Presentation**	WP ref: **WP 8**	
Major constituent: Documentation	Sheet 8 of 8	
Short description: Oral Presentation of the project.	Planned start date: 20/06/2020 Planned end date: 10/07/2020 Start event: T1 End event: T1	
Internal task T1: Prepare the slides and the speech.	Deliverables: Presentation_TFG_ PolMasóAyats.pdf	Dates: 13/07/2020- 17/07/2020

Milestones

WP#	Task#	Short title	Milestone / deliverable	Date (week)
1	1	Project definition and description	Documentation	4
1	2	Project development plan	Draft	4
1	3	Document review and approval	Project_Proposal_ And_Workplan_TFG_ PolMasóAyats.doc	4
2	1-3	Learning how to use the principal tools (Python, Pytorch, ENVI…)	-	1-6
2	4	Do research of Deep neural network for Super-resolution of multitemporal Remote Sensing Images	Software	4-6
3	1	Download images captured by Sentinel-2 and create subsets using the bands 2, 3, 4 and 8.	-	5
4	1	Use transfer learning and internet resources to build an adequate CNN for this project	Software	5-12
4	2	Use the images in the Sentinel-2 dataset to test the Software developed	Software	10-14
5	1	Analyze and review the project state and compare it with the initial work plan	Review/Draft	8
5	2	Write and deliver the document	Critical_Review _TFG_PolMasóAyats.doc	9
6	1	Test the performance of the system with the test images	Software	13-14
6	2	Fine-tune the results and look for improvements	Software	14-15
6	3	Get results	Review	16
7	1	Write the document	Final_Report_ TFG_PolMasóAyats.doc	14-18
8	1	Prepare the slides and the speech		18-21

Gantt Diagram



Deep neural network for Su...	start	end
Project proposal and workplan	**09/03/20**	**12/03/20**
Project definition and description	09/03	12/03
Project development plan	09/03	12/03
Document review and approval	09/03	12/03
Information research and documen...	**17/02/20**	**27/03/20**
Learning how to use the principal too...	17/02	27/03
Do research of Deep neural network ...	09/03	27/03
Download images to create the im...	**16/03/20**	**20/03/20**
Download images captured by Senti...	16/03	20/03
Software development	**18/03/20**	**22/05/20**
Use transfer learning and internet re...	18/03	08/05
Use the images in the Sentinel-2 dat...	20/04	22/05
Critical Review	**10/04/20**	**14/04/20**
Analyze and review the project state...	10/04	10/04
Write and deliver the document	13/04	14/04
Test and results assessment	**15/05/20**	**05/06/20**
Test the performance of the system ...	15/05	26/05
Fine-tune the results and look for im...	18/05	29/05
Get results	01/06	05/06
Final report	**18/05/20**	**20/06/20**
Write the document	18/05	20/06
Oral Presentation	**20/06/20**	**10/07/20**
Prepare the slides and the speech	20/06	10/07

Appendix B: GitHub Code Link

https://github.com/pmaso98/TFG_SR

Glossary

GPI	Image and Video Processing Group
TSC	Signal Theory and Communication Department
UPC	Universitat Politècnica de Catalunya
ESA	European Space Agency
DL	Deep Learning
CNN	Convolutional Neural Network
AI	Artificial Intelligence
LR	Low Resolution
HR	Hight Resolution
SR	Super Resolution
SISR	Single Image Super Resolution
MISR	Multiple Image Super Resolution
VIS	visible spectrum
NIR	near-infrared
SWIR	mid-infrared