POLITECNICO DI TORINO

Master's Degree in Computer Engineering

Politecnico
di Torino

Master's Degree Thesis

Proba-V Super-Resolution in combination with Sentinel-2

Supervisors

Prof. Enrico MAGLI

Prof. Diego VALSESIA

Dr. Fabrizio NIRO

Candidate

Gabriele INZERILLO

October 2022

Abstract

In the field of Deep Learning, and more specifically Computer Vision, one of the tasks that covers quite a lot of interest is that of Super-Resolution: a set of techniques used to improve the resolution of digital images. Super-Resolution techniques have found room for application in many fields, among which one of the most interesting is that of remote sensing and earth observation; this is amply evidenced by the very numerous challenges on the subject organized by space entities such as ESA and NASA.

Being able to improve the spatial resolution, i.e. the physical measurement (meters) that represents the size of a pixel, of a satellite image can be particularly useful for a number of reasons including being able to make object classification and detection tasks easier to solve, or again, to monitor at a greater level of detail the earth's surface. High spatial resolution images, however, are produced by remote sensing satellites less frequently than low spatial resolution ones, which is why having models available that allow one to increase resolution from one or more low resolution images turns out to be a critically important task.

PIUnet is a model that performs MultiTemporal Image Super Resolution, created by a research group at Politecnico di Torino as a result of a challenge convened by ESA, which increases the spatial resolution of images from the Proba-V satellite from 300 to 100 meters. Starting from the existing Super-Resolution architecture PIUnet, the purpose of this thesis is to evaluate how the Super-Resolution task can be performed using images from two different missions, Proba-V and Sentinel-2. Specifically, we want to make sure that the low-resolution model training images remain those from Proba-V while the ones used as Ground-Truth (i.e., high resolution) are instead selected from Sentinel-2.

To do this, the work presented was divided into several phases: a first phase of creation of the dataset comprising low-resolution images from Proba-V and high-resolution images from Sentinel-2, a second phase consisting in the training of the model on the new dataset, a third phase of evaluating the results obtained from the standard version of PIUnet, and finally a fourth phase of modifying PIUnet to make it suitable for working with images from two different satellites. Each of these stages is described in details within this thesis. Furthermore, at the end of this work, both the results obtained using the standard version of PIUnet and the results obtained from the modified version of PIUnet are also presented, analyzed, and compared in depth, showing how the changes made on PIUnet resulted in better quality, radiometrically-consistent, super-resolved images.

A mia sorella, la persona più importante della mia vita, con l'augurio che possa ottenere tutto ciò che desidera e con la consapevolezza che nel lungo, e a tratti burrascoso, viaggio della vita suo fratello sarà sempre un porto sicuro in cui attraccare.

Acknowledgements

First of all, I would like to thank Prof. Enrico Magli who, although he did not know me directly, not having been his student directly, gave me the confidence and the opportunity to embark on this thesis path, and Prof. Diego Valsesia who helped me and clarified many technical and non-technical doubts throughout the thesis.

Secondly, I sincerely thank Serco Italia for giving me the opportunity to start working in a corporate environment full of competent people and for providing me with part of the resources necessary to carry out this work; in particular, I thank Dr. Fabrizio Niro and Dr. Erminia De Grandis, who with great reliability and willingness have constantly followed me in the development of the thesis, helping me to understand many concepts, and to solve many doubts about remote sensing, a field unknown to me until seven months ago.

Last but not least, I would like to thank my two families: the biological family that has supported and loved me for 24 years, and the *kyssene* family, a group of students that was initially born for study purposes and that, as time has passed, has increasingly turned into a second family without which I would never have become what I am now.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

In recent years, applications of Deep Learning, and more specifically Computer Vision, have been gaining popularity in many fields of research due to their capability to solve very complex tasks more quickly, accurately and easily than the traditional programming-based approach of computer science. Among the many Computer Vision techniques subject to study and research, those of Super-Resolution are of particular interest, especially in the fields of space, remote sensing and earth observation. Super-resolution techniques aim to improve the quality (i.e., resolution) of digital images; it is therefore clear that having access to accurate techniques of this kind can be particularly useful in the field of remote sensing since the spatial resolution, i.e., the physical measurement representing the size (in meters) of a pixel, of a satellite image is a crucial parameter in determining the overall quality of the image and, thus, leading to more or less accurate analyses.

Starting from an existing Super-Resolution architecture called PIUnet, created by a research group at Politecnico di Torino as a result of a challenge convened by ESA, the purpose of this thesis is to evaluate how the Super-Resolution task can be performed on the same architecture using images from two different missions, Proba-V and Sentinel-2, and possibly improve the results by modifying the network itself. In fact, in its initial and original implementation, PIUnet was designed to work with, and increase the resolution of images from, only one satellite: Proba-V, while with this work we want to make sure that the low-resolution model training images remain those from Proba-V while the ones used as Ground-Truth (i.e., high spatial resolution) are instead selected from Sentinel-2. The choice of these two missions is mainly due to two considerations:

1. Sentinel-2's data are sufficiently consistent with Proba-V in terms of overpass time, radiometry and spectral coverage in the visible and near-infrared spectral range; this, in theory, should ensure a smooth data fusion process, at the same time, however, it should be considered that the two satellites have different

sensors, spatial resolutions and, in general, different characteristics, so the result obtained using the standard PIUnet architecture could present various problems and thus not be accurate.

2. Sentinel-2's images achieve, in the near-infrared and visible range, a spatial resolution of 10 meters while Proba-V data reach up to a maximum of 100 meters, consequently the spatial resolution of Super-Resolved images could be further improved in the future (compared with standard PIUnet architecture, which increases the spatial resolution of images from 300 to 100 meters).

1.1 Structure of the document

The following thesis is structured as follows:

- Chapter 1, *Introduction*, contains a brief introduction to the thesis.
- Chapter 2, *Remote Sensing and Earth Observation*, aims to introduce the reader to some fundamental concepts about remote sensing and earth observation. This is made necessary by the fact that the thesis makes strong use of terms, concepts and techniques of remote sensing; it is therefore crucial to understand what are the main issues of this domain and how artificial intelligence can contribute to its development.
- Chapter 3, *Deep Learning Background*, introduces the reader to the main knowledge of machine learning and deep learning. In addition, the Super-Resolution architecture, the basis of this thesis work, called PIUnet, is analyzed and described in detail.
- In Chapter 4, *Datasets*, is presented the used dataset and the related process of creation as well.
- Chapter 5, *Methods and Trainings*, contains a detailed description of the changes made to PIUnet during the thesis and of all the experiments made.
- In Chapter 6, *Results*, the results obtained are presented and discussed.
- Chapter 7, *Conclusion*, contains the conclusion of the work and some possible future ideas to improve or extend the results.

Chapter 2

Remote Sensing and Earth Observation

This chapter aims to provide the basic concepts of Remote Sensing and Earth Observation.
This thesis makes strong use of terms, concepts and techniques related to the domain of Remote Sensing and Earth Observation; for this reason it is important to understand the basic notions and the main issues related to this specific domain.

2.1 Remote Sensing

Remote Sensing is the process of acquisition of information about objects and geographical areas by means of sensors, usually placed on planes and satellites, that are capable to measure emitted and reflected radiation by the analysed area or object.

This technical-scientific discipline is used in numerous fields, some practical examples and uses of Remote Sensing include:

- Environmental and crop monitoring.
- Tracking of deforestation and desertification.
- Monitoring and responding to natural disasters and catastrophes.
- Weather forecasting.
- Monitoring of urban settlements.

2.2 Electromagnetic Spectrum and Radiation

The electromagnetic spectrum (Fig. 2.1) is the range of frequencies of electromagnetic radiation and their respective wavelengths and photon energies. It contains the range of all the electromagnetic radiation and is divided into several subranges according to the frequency of the waves and the wavelength.

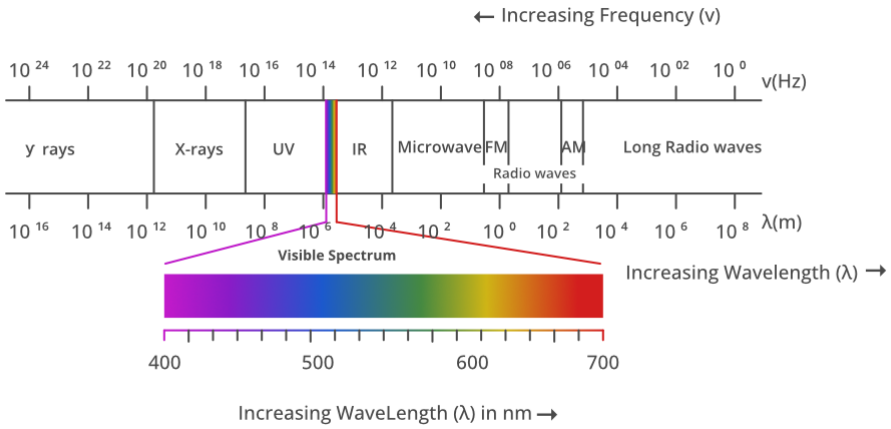


Figure 2.1: Electromagnetic Spectrum

The main sources of electromagnetic radiation used in Remote Sensing are three:

- **Solar Radiation**: electromagnetic radiation emitted by the sun.
- **Longwave Radiation**: electromagnetic radiation emitted by the earth's surface and Earth's atmosphere in the form of thermal radiation.
- **Artificial Radiation**: electromagnetic radiation generated by an artificial source (e.g. radars).

The spectral response of a certain object/surface, measured by a sensor, is determined by the way the electromagnetic radiation interacts with that specific object/surface, thus, we can distinguish different elements on surface such as water, soil, vegetation etc. just by looking at the reflected or emitted radiation, in fact, there are three ways of interaction that can take place when energy strikes a certain surface; these are: absorption, transmission and reflection. Notice how the proportions of each will depend on multiple factors such as the wavelength of the radiation and the target surface's material.

2.2.1 Interaction with the atmosphere

Before the electromagnetic radiation reaches the sensor on the satellites/planes it has to travel some distance, more precisely:

1. The electromagnetic radiation is radiated by a source, as we have seen above.
2. It propagates through space reaching the Earth.
3. It interacts with the Earth's atmosphere before, and the Earth's surface after.
4. Once reflected, the radiation interacts again with the atmosphere.
5. At the end it reaches the remote sensors mounted on satellites.

During this trip, when passing through the Earth's atmosphere, the electromagnetic radiation is subject to three physical phenomena:

- **Scattering**: scattering (Fig. 2.2) happens every time the radiation changes its direction after interacting with particles and gas molecules in the atmosphere. Wavelength of the radiation, quantity, type of particles and other factors lead to different kinds of scattering, there are three main types: *Rayleigh scattering*, that is caused mainly by oxygen and nitrogen molecules and occurs when the radiation hits small-sized particles, compared with its wavelength; *Mie scattering*, that occurs when the radiation wavelength and the particles have similar sizes; *Nonselective scattering*, that occurs when the particles are way larger than the wavelength of the radiation.
- **Absorption**: part of the radiation interacting with the atmosphere is absorbed by the gas molecules, leading to a decrease in intensity of the electromagnetic radiation.
- **Refraction**: when a wave passes from one medium to another, refraction occurs. In Remote Sensing the atmospheric refraction is a well-known phenomenon that causes the deviation of electromagnetic waves due to changes in air density.

2.2.2 Reflectance

The **reflectance** of a material is its ability to reflect the radiant energy incident on its surface. In a more formal way the hemispherical reflectance R of a surface can be defined as:

$$R = \frac{\phi_r}{\phi_i}$$

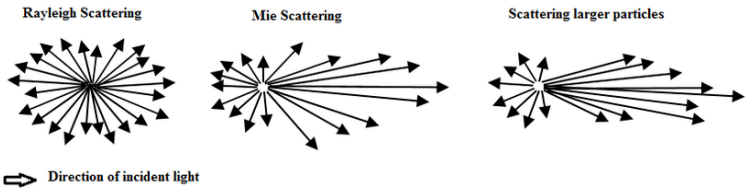


Figure 2.2: Types of scattering
[1]

Where ϕ_r is the flux reflected by the surface and ϕ_i is the incident flux on the surface. Since this measure is a ratio of radiant flux it has no unit of measure and ranges between 0 and 1.
The reflectance is probably one of the most important measures used in Remote Sensing because from it one can derive what type of surface is been analysed, it is indeed crucial to note that the amount of electromagnetic radiation that will be reflected depends mainly on the nature and properties of the material; every object or surface has a specific **spectral response pattern** (Fig. 2.3): in a certain wavelength region a specific surface or object has a spectral response pattern that is different from other objects, thus it is possible to distinguish different surfaces and objects just by looking at their spectral signature.

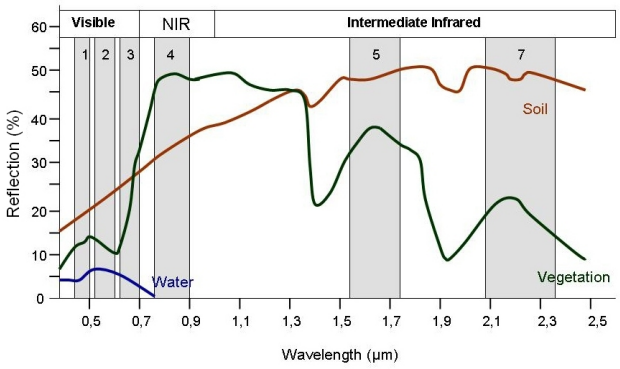


Figure 2.3: Reflectance response patterns of different surfaces, respectively vegetation, soil and water. The vertical gray bands represent the different band of acquisition of the sensor (*Landsat7*).
[2]

2.3 Sensors

Remote Sensing sensors are the instruments used to measure and collect data about the electromagnetic radiation incident to a certain object or surface. We can distinguish among two main macro-categories of sensors: active sensors and passive sensors. An **active sensor** provides and emits the waves needed to scan an object or a surface in order to detect the reflected electromagnetic radiation and measures its spectral signature, while, a **passive sensor** detects and measures the spectral signature of the radiation emitted by a natural source (mainly the sun) and then reflected by the target surface.
One of the main components of a Remote Sensing sensor is the scanner: sensors make use of electro-optical scanners to produce images, using detectors that measure reflected or emitted electromagnetic energy by a surface. The width of the scanned area is called *Swath*, while the *FOV* (Field Of View) represents the angle that the scanner can capture from a certain distance. There are two types of scanners:

- **Whiskbroom scanner**: also called across-track scanner (Fig. 2.4, *A*), it uses rotating mirrors to scan a surface. The scan is perpendicular to the motion of the sensor, moving from one side to another, scanning the target area cell by cell.
- **Pushbroom scanner**: also called along-track scanner (Fig. 2.4, *B*), it uses an array of detectors to scan parallel to the surface. Instead of scanning the surface from one side to another, cell by cell, the pushbroom scanner scans the entire swath line at once.

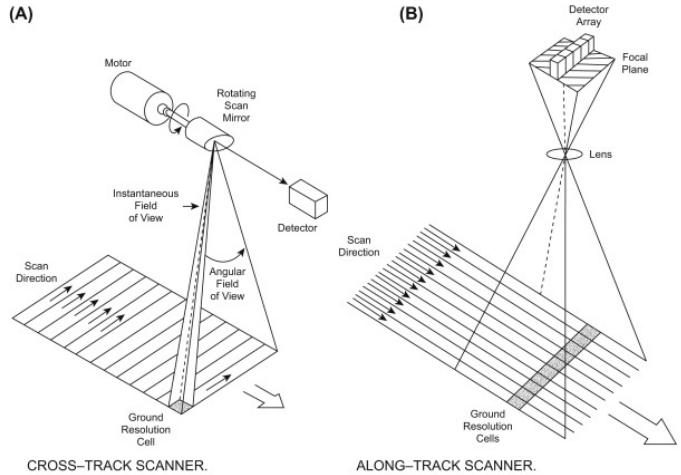


Figure 2.4: Whiskbroom scanner (*A*) and Pushbroom scanner (*B*) [3]

2.3.1 Sensors resolutions

Four resolutions are used to describe a remote sensing sensor:

- **Spatial Resolution**: it defines the dimension (often measured in meters) of a pixel of the image recorded by the sensor, an image whose spatial resolution is equal to 60 meters represents pixels whose sides measure 60 meters. Keep in mind that this is an approximation, because in satellite imagery a pixel is never a perfect square. More information about spatial resolution and sensor quality are provided later in Section 2.6 and 2.7.
- **Radiometric Resolution**: it defines the number of intensities of electromagnetic radiation that the sensor is able to distinguish. In other words we can define the radiometric resolution as the sensitivity of a sensor in perceiving and codifying in signal the differences given by the radiant flux reflected or emitted by a surface. The higher the radiometric resolution, the more accurate the sensed image will be; Figure 2.5 clarifies this concept.
- **Spectral Resolution**: it defines the number of spectral bands and the relative wavelength that the sensor is able to measure. A spectral band is defined by the central wavelength λ and the band width $\Delta\lambda$.
- **Temporal Resolution**: it defines how much time passes from the acquisition of a certain area and the subsequent acquisition of the same area, or simply the time that a satellite takes to fly over a certain point again; it can be decreased by using a constellation of satellites instead of a single satellite. This resolution is crucial to monitor changes in the landscape.

2.4 Satellite orbits

Since all the data used in this work come from satellites, a distinction between the types of satellite orbits is needed. We can distinguish between two major orbits:

- **Polar orbit**: polar orbits are for ensuring global coverage and are generally *Sun-Synchronous* (SSO), meaning that the crossing time at the equator is at the same local time, this is essential for multi-temporal studies, since the illumination conditions are the same. All satellites of the Copernicus suite (such as Sentinel-2, Sentinel-3 etc..) and Proba-V follow Sun-Synchronous Orbits. This orbit goes from north to south, usually passing over the poles, and has a very low altitude (between 600 and 800km).
- **Geostationary orbit**: the satellites that are in this orbit move from west to east, above the equator, and are usually used for sensing the same area on

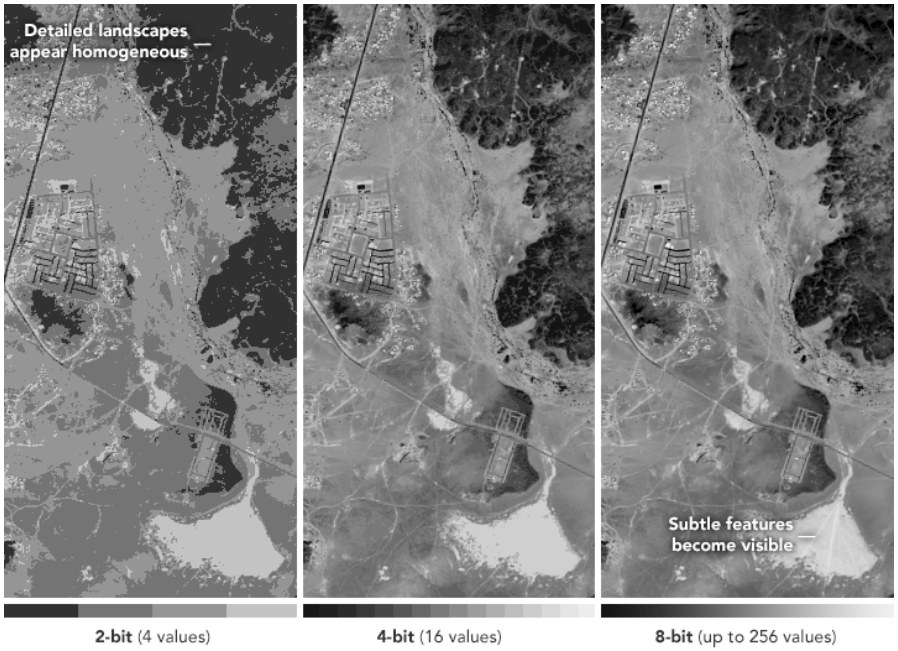


Figure 2.5: Same image with different radiometric resolutions [**Credit: NASA Earth Observatory images by J. Stevens, using Landsat data**]

Earth with a very high revisit (order of minutes), allows therefore to monitor very dynamic processes (e.g. clouds) notably for meteorological applications.

2.5 Image Processing

A fundamental part of the work related to remote sensing consists in the subsequent processing and analysis of the acquired images. Normally the generic dimensionless measure to represent the value of a pixel of a remote sensed image is called **Digital Number** (**DN**); purpose of the DN is to represent the values of the pixels that have not yet been converted into physical units of measurement (such as radiance and reflectance). Obviously having the data expressed in DN is of little use as in any physical measurements, the first step consist in moving from the sensor specific output to a physically meaningful value expressed in standard scientific units, this is done through the calibration (further information in the next section), that's why the products of the satellites that perform remote sensing represent the values of the pixels in radiance or reflectance, specifically we distinguish between two distinct processing stages of radiance and reflectance:

- **TOA** (**Top-Of-Atmosphere**): as we said when we talked about radiance

and reflectance we must consider that the radiant energy reflected when taking satellite images is not given only by the reflection given by the Earth's surface, but also by some physical phenomena caused by the atmosphere, such as scattering and absorption. The TOA measures this kind of *raw* radiance/reflectance where the contribution of the reflected light is given by both the atmosphere and the Earth's surface.

- **BOA** (**Bottom-Of-Atmosphere**): in this quantity the contribution from atmospheric scattering and absorption is removed through an *Atmospheric Correction* process; so basically, this reflectance/radiance does not measure the contribution made by the atmosphere or other atmospheric elements such as clouds and it preserves only the part of radiant energy reflected by the surface below (Earth's surface).

2.5.1 Calibration

When talking about **calibration** we mean the operation in which a measuring instrument is adjusted in order to improve its accuracy, specifically with the calibration of the sensors we want to relate the value measured by a sensor with the corresponding uncertainty measurement of that instrument. It is therefore crucial in Remote Sensing to have an idea of what calibration is, and what types of calibration exist because when detecting multispectral images even a minimum error in the accuracy of the instrument leads to incorrect results (physical quantities).

As strongly underlined by R.Müller [4], calibration has to be applied in order to relate the digital counts given by the sensor to the incoming radiances, consequently, the physical units of interest. The relation between the digital numbers and the radiances can be derived by comparison of the sensor signal with an absolute standard reference prior to launch. Nowadays, satellite instruments are usually well designed and calibrated prior to launch. Unfortunately, no matter how sophisticated the instruments are, once in space they degrade with time, e.g., due to thermal, mechanical or electrical effects or exposure to UV radiation. In order to account for such ageing, on-board calibration devices are generally placed on board, allowing to monitor and periodically correct for instrument drifting. Likewise, vicarious well characterized and stable Earth's and planetary targets are used to monitor sensor performances during the mission lifetime.

There are two main types of calibration in Remote Sensing that can be performed:

- **Geometric Calibration**: the goal of the geometric calibration is to obtain a geometrically correct image, that is where each pixel is mapped to its corresponding geographical location.
- **Radiometric Calibration**: this calibration is needed to convert raw data (Digital Number, as we have seen before) measured by the sensors in meaningful

physical scale-based units of measure such as TOA reflectance and thus obtain physical values that give us information about the image.

Another important technique is the vicarious calibration. The **Vicarious Calibration** makes use of natural (e.g. deserts, ocean etc.) or artificial sites on earth to perform on-board calibration; these sites used for calibration are not chosen randomly but have very specific characteristics from both a spatial and spectral points of view. By measuring the radiance and reflectance from these sites and comparing them with the actual standard measurements obtained on the ground prior to launch, you can see how much accuracy the sensors have lost and work accordingly to recalibrate them.

2.5.2 Orthoimagery and Orthorectification

When we take a photo, the image we are in front of obviously gives us a sense of perspective, even if it is two-dimensional. An **orthoimage**, on the other hand, is an image that has been geometrically corrected (through an orthorectification process) and georeferenced in such a way that the scale of representation of the photograph is uniform, i.e. the photo loses perspective and can therefore be considered as a geographical map (Fig. 2.6).

Orthoimages are particularly useful as they can be used to measure real distances as they correctly represent the surface of the earth. To obtain an orthoimage, the **orthorectification** process must be applied: it consists of a series of transformations such as projections, rotations, translations and more that have the purpose of correcting an image from various deformations due to both shooting (it is indeed hard to always take images from the same quota) and the optical instrumentation used.

2.5.3 Digital Evaluation Model

Another important concept in remote sensing imagery is the **Digital Evaluation Model** (**DEM**). A DEM is a digital model that represents the distribution of altitudes of a territory/surface. This type of model is produced by associating to every pixel of an image its absolute quota. More specifically we can distinguish between **Digital Surface Model** (**DSM**) and **Digital Terrain Model** (**DTM**). The difference between these two models is that with the DSM we get a 3D model that takes into account the height of the surface including all the objects that are placed above (houses, trees etc.) while the DTM takes into account only the absolute height of the ground (i.e. Earth surface).

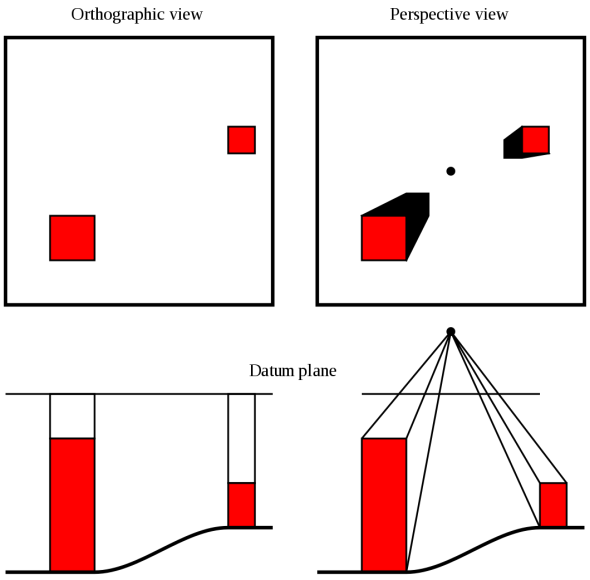


Figure 2.6: Comparison between orthographic and perspective view [**Credit: Wikipedia**]

2.5.4 Image co-registration

The process of **image co-registration** has the goal to align two images geometrically. The images that are used for remote sensing are often detected at different temporal moments, furthermore, the satellite may have different crossing time at the Equator so the opportunity of over-flight the same spot at the same time is generally very limited; also, the local time descending node (LTDN) changes across missions, for land missions is usually 10:00 or 10:30 AM, because there is less probability of cloud coverage, but there are also mission crossing at 13:00PM (for example: MODIS AQUA). All of the above imply that two images of the same area have offset positions of the pixels even if the surface/area of the images is the same, so, co-registration is needed to minimize these pixel shifts and align the images.

2.6 Ground Sampling Distance

In an orthoimage, the **Ground Sampling Distance** (**GSD**) represents the distance between the center of two consecutive pixels in the territorial unit of measurement such as meters. It is clear how there is an inverse proportionality relationship between the GSD value and the definition of an image: the larger the GSD, the lower its level of detail. It is moreover easy to understand that the GSD is strictly connected to the spatial resolution of a satellite.

2.7 Point Spread Function

The **Point Spread Function** (**PSF**) represents the impulsive response of a system to a point object (Fig. 2.7), in fact, an image can always be described as a blurred representation of a certain object; the degree of blurring of the point object is the PSF of that specific imaging system and it is an important measure for the quality of the latter.

It is clear how it can be used as a quality measure for an optical system, like the remote sensing sensors: the higher the blurring, the less the quality of the acquired image.

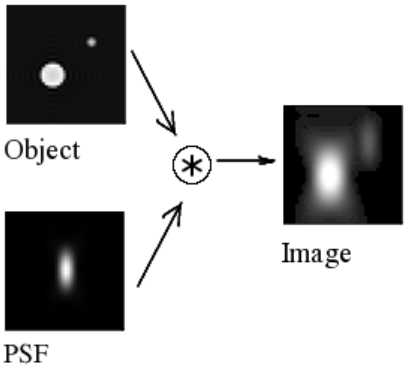


Figure 2.7: Point Spread Function
[**Credit: Wikipedia**]

2.8 Proba-V and Sentinel-2 missions

The goal of this last section is to provide an overview of the two missions Proba-V and Sentinel-2. The datasets used in this work contains data acquired by the two missions previously mentioned; more details will be provided later in Chapter 4.

2.8.1 Proba-V

Proba-V is a satellite launched in 2013 by the European Space Agency. It is located at a distance of 820km from the Earth's surface and acquires data in 4 spectral bands: *Blue*, *Red*, *Near-Infrared* (NIR) and *Short Wave Infrared* (SWIR). Three sensors (Fig. 2.8) are mounted in the satellite, arranged in such a way that there is one in the center and the other two on the sides. Each sensor is equipped with two focal planes respectively to capture the SWIR and visible waves to the human eye (Blue, Red) and NIR [5].

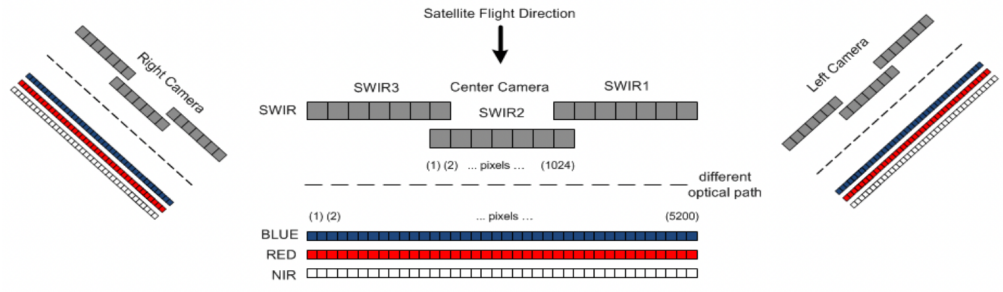


Figure 2.8: Proba-V Instrument layout
[5]

Proba-V produces products in three different spatial resolutions: 1000m, 300m and 100m. Figure 2.9 summarize both the radiometric and geometric specifications of the Proba-V satellite at launch.

Radiometric specifications		
Band centre (nm)	Bandwidth (nm)	SNR at L_{ref}
463	46	155 at 111 W m^{-2} sr^{-1} μm^{-1}
655	79	430 at 110 W m^{-2} sr^{-1} μm^{-1}
845	144	529 at 106 W m^{-2} sr^{-1} μm^{-1}
1600	73	380 at 20 W m^{-2} sr^{-1} μm^{-1}
Radiometric performance		
Absolute accuracy	5%	
Inter-channel accuracy	3%	
Stability	3%	
Spectral misregistration	0.9 nm VNIR, 2 nm SWIR	
Polarization sensitivity	4% for the Blue band; 1% otherwise	

Geometric specifications	
Field-of-view and swath	102.4° and 2295 km
Ground sampling distance (GSD)	1/3 km × 1/3 km HR, 1 km × 1 km LR
Modulation transfer frequency (MTF)	> 30% at Nyquist frequency
Absolute geo-location accuracy	<1 HR GSD
Inter-band geo-location accuracy	<1/3 HR GSD
Multi-temporal geo-location accuracy	< 1/2 HR GSD

Figure 2.9: Proba-V specifications at launch
[6]

Products

As we can see in Figure 2.10 Proba-V products are divided in several levels, and only some of these levels are available to the end users. The products are divided into two categories:

- **Segment products**: are those obtained from levels 1C and 2A. The Level 1C (L1C) product contains the raw, unprojected observations in segments, as well as calibration information, while the Level 2A (L2A) products contain the projected segment data [5].

- **Synthesis products**: are the level 3 products. These products contain daily (S1, available at all resolutions) and multi-daily (S5 for 100 m and S10 for 300 m and 1 km) TOA reflectances that are composed of cloud, shadow, and snow/ice screened observations [5]. Additionally, Top-of-Canopy (TOC) reflectance (corresponding to the BOA reflectance we talked about earlier in this chapter) are available in this level.

Processing levels

This subsection describes the various Proba-V products, following all the processing levels visible in Figure 2.10.
In Level 1 the main steps performed are the *geometric* and *radiometric calibration*; this level is divided into the following sub-levels:

- **Level 1A**: at this level data are raw and uncompressed; here a geolocation step is performed in order to associate each pixel to its relative longitude and latitude. Geometric calibration operations are performed with the support of the Instrument Calibration Parameters (IPCs) that are regularly updated.

- **Level 1B**: at this level TOA reflectance is calculated starting by the raw data (measured in Digital Numbers) obtained in the previous level. First of all, DN^k associated to the k spectral band is corrected in order to remove pixel uniformities, dark currents and non-linearities. Then, using ICPs the DN^k is converted in L^k radiance. Finally, TOA reflectance is calculated for each k spectral band starting from the L^k radiance and other parameters such as the Earth-Sun distance, the mean atmospheric irradiance and the Solar Zenith Angle (SZA).

- **Level 1C**: the first end user segment product is available in this level.

In Level 2, products from Level 1C are further processed going through two sub-levels:

- **Level 2A**: level 2A products are the results of a series of processing steps. It follows their description:

    1. **Mapping**: in this processing step the data are mapped onto the WGS84 latitude-longitude projection.

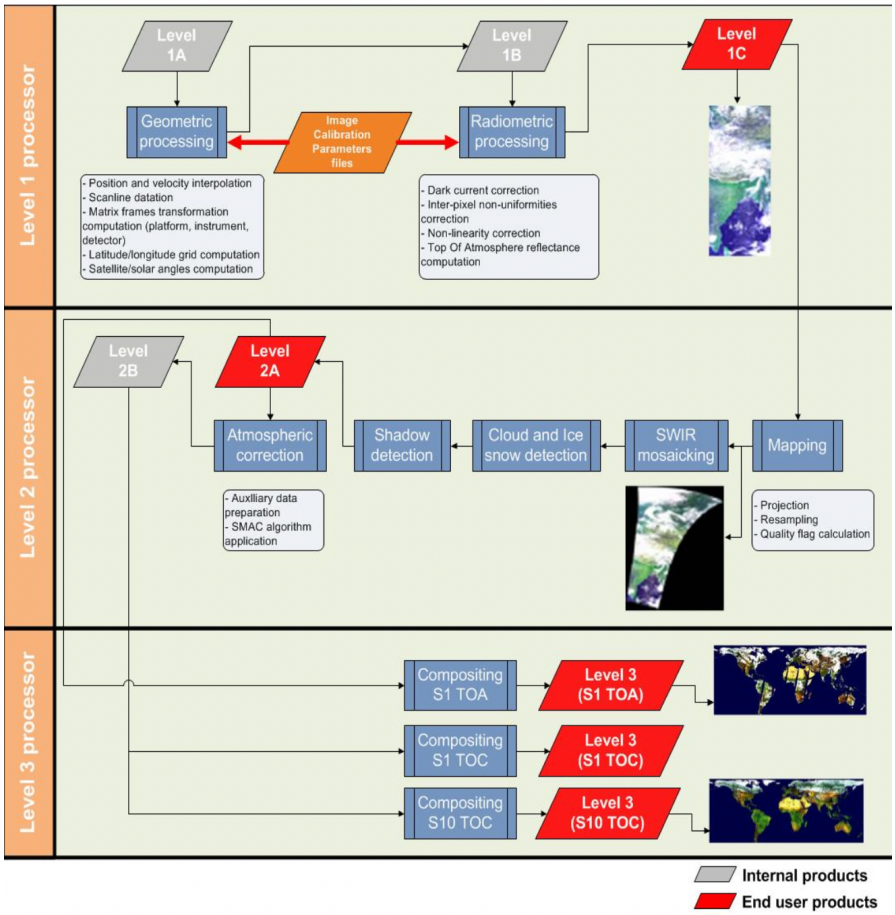


Figure 2.10: Proba-V levels and processing flowchart [5]

2. **SWIR Mosaicking**: after the mapping, there are still three separately projected SWIR strips. Therefore a mosaicking step is applied to compose a single SWIR band image [5].
3. **Cloud detection**: at this point we can guess that clouds are a big obstacle to the analysis of satellite imagery, so it is necessary to identify clouds from images before continuing with data analysis. For this reason a *cloud detection algorithm* is implemented: briefly, this algorithm consists in an extended set of thresholds and similarity checks, detecting the radiometric contrast between surface and clouds, performed on the values of the Blue and SWIR spectral bands , if the threshold value is smaller than the pixel's value then the pixel is marked as 'cloud'. The output of this algorithm is the **cloud mask**.
4. **Snow/Ice detection**: an algorithm for identifying snow and ice that

works similarly to that of cloud detection is implemented using the data of all four spectral bands and four different values of threshold.

5. **Cloud Shadow detection**: as with clouds, cloud shadows are also a possible source of data error; for this reason a cloud shadows detection algorithm is implemented and a **shadow mask** is generated.

- **Level 2B**: the Level 2A TOA reflectance observations are the resultant of surface reflectance and scattering, absorption, and multiple reflections within the atmospheric column below the satellite (clouds, gases, aerosols). In order to obtain the directional TOC reflectance values, atmospheric correction is performed. This model converts the observed TOA reflectance into TOC reflectance using auxiliary water vapour, ozone, and surface pressure data [5].

At Level 3 we have as final products those obtained from level 2B processing steps and which pass through the composition step; purpose of this step is to combine, in an optimal way, observations made in different time intervals in a single image called synthesis image that does not contain clouds. More specifically, level 3 products differ in S1, S3 and S10 based on how many days of data the product summarizes.

The outputs of the previously mentioned algorithms are used to generate the so called **status map** (Fig. 2.11); in the next chapters I will discuss in detail the crucial role of this mask for this work.

Bit (LSB to MSB)	Description	Value	Key
0-2	Cloud/Ice Snow/Shadow Flag	000	Clear
		001	Shadow
		010	Undefined
		011	Cloud
		100	Ice
3	Land/Sea	0	Sea
		1	Land
4	Radiometric quality SWIR flag	0	Bad
		1	Good
5	Radiometric quality NIR flag	0	Bad
		1	Good
6	Radiometric quality RED flag	0	Bad
		1	Good
7	Radiometric quality BLUE flag	0	Bad
		1	Good
8*	SWIR coverage	0	No
		1	Yes
9*	NIR coverage	0	No
		1	Yes
10*	RED coverage	0	No
		1	Yes
11*	BLUE coverage	0	No
		1	Yes

Figure 2.11: Proba-V Status Map pixel's values [5]

2.8.2 Sentinel-2

The *Sentinel-2* is an Earth Observation mission developed by the European Space Agency (ESA). Specifically, Sentinel-2, as part of the Copernicus Programme, aims to monitor the green areas of the planet, coastal areas, arable areas and much more in order to support the process of managing natural disasters, monitoring land changes, monitoring crops etc. To ensure a low temporal resolution and continuous image availability, two twin satellites, the Sentinel-2A and the Sentinel-2B, operate simultaneously on the same polar-orbit at a distance of 786 km from Earth, offset by 180 degrees. Each of the satellites acquire data by the *MultiSpectral Instrument* (MSI), an optical sensor that relies on a pushbroom scanner to scan the Earth's surface, with an orbital swath width equal to 290km.

The images captured by the Sentinel-2 satellites are 13-bands multispectral images ranging from visible infrared (NIR) to short-wave infrared (SWIR) with spatial resolutions of 10, 20 and 60 meters, according to the specific spectral band. More details about the spectral bands of the Sentinel-2 missions in Figure 2.12.

Band number	Central wavelength (nm)	Band width (nm)	Lref ($Wm^{-2}\,sr^{-1}\,\mu m^{-1}$)	SNR @ Lref
1	443	20	129	129
2	490	65	128	154
3	560	35	128	168
4	665	30	108	142
5	705	15	74.5	117
6	740	15	68	89
7	783	20	67	105
8	842	115	103	174
8b	865	20	52.5	72
9	945	20	9	114
10	1380	30	6	50
11	1610	90	4	100
12	2190	180	1.5	100

Figure 2.12: Sentinel-2 spectral bands
[7]

The Sentinel-2 mission includes two segments (i.e. main components): the space one, consisting of the two satellites that detect the images, and the ground one, which aims to facilitate the acquisition of data from the satellites, the processing and storage of data and mission control in general.

Products

Two products are generated by the Sentinel-2:

- **Level-1C products**: TOA reflectances in cartographic geometry.

- **Level-2A products**: BOA reflectances in cartographic geometry.

Products are a compilation of elementary granules (25km × 23km images) of fixed size, within a single orbit. A granule is the minimum indivisible partition of a product (containing all possible spectral bands). For Level-1C and Level-2A, the granules, also called tiles, are $100 \times 100 km^2$ orthoimages in UTM projection: the UTM (Universal Transverse Mercator) system divides the Earth's surface into 60 zones. Each UTM zone has a vertical width of 6° of longitude and horizontal width of 8° of latitude.

Processing levels

Similarly to what happens with ProbaV, also the Sentinel-2 products are obtained by passing through a series of processing levels (Fig. 2.13), however, end users have access to Level1C and Level2A products only. Level-0 and Level-1A take care of collecting compressed and subsequently decompressed data. In level 1B, radiometric and geometric corrections and identification of defective pixels are applied to the decompressed data. Subsequently, at level 1C, further radiometric and geometric corrections are applied, furthermore here the conversion to reflectance is performed and the cloud and land/water mask are generated.

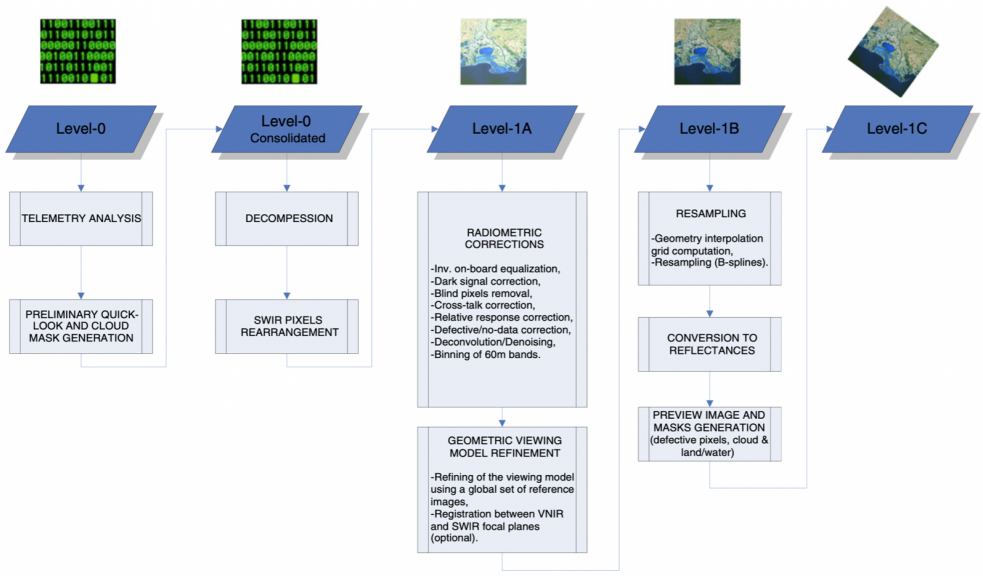


Figure 2.13: Sentinel-2 processing levels
[**Credit: ESA**]

The data processing steps at **Level-0** are performed in real-time by the PDGS (Payload Data Ground Segment) during the data reception. The operations done at

this level are necessary to subsequently store the data, metadata and subsequently produce higher level products.

Level-1 is divided into the following sub-levels:

- **Level-1A**: at this level the decompression of the data (granules) obtained from level-0 is performed, therefore, each pixel is localized. The output of this layer is a decompressed granule.
- **Level-1B**: radiometrically corrected data measured in TOA radiance are obtained as output of this level. In addition, geometric corrections are applied to the images in this layer.
- **Level-1C**: level-1C products are $100 \times 100 km^2$ orthoimages called tiles. Using a DEM (Digital Elevation Model) is fundamental in order to reproject these products in UTM. Per-pixel radiometric measurements are provided in TOA reflectances. Additionally at this level some useful masks, such as cloud and land/water masks, are computed.

Finally, the **Level-2A** mainly deals with performing atmospheric corrections in such a way as to change the images (tiles) of level-1C, which measure the reflectance in TOA, and thus convert it in BOA reflectance. In addition to this, in this level there is an algorithm for the Scene Classification (classification of the elements visible in the image) that allows us to identify, pixel by pixel, clouds (of various kinds and their respective shadows), snow and other pixel classes. The output of this algorithm is the Scene Classification Map that, similarly to what we have already seen with the ProbaV status map earlier, distinguishes the elements present in the image between clouds, clouds shadows, vegetation, non-vegetation, water and snow.

Chapter 3

Deep Learning Background

This chapter introduces you to some of the basic concepts about machine learning and, more specifically, deep learning. Furthermore, the end of this chapter provides an in-depth analysis of *PIUnet (Permutation Invariance and Uncertainty Network)*, the neural network that has been used, and modified, for the development of this thesis. Given the crucial role of *PIUnet* in this thesis, the most important concepts related to the latter will be covered in depth through this chapter, while, other DL and ML's concepts that are not strictly connected to *PIUnet* will be presented more superficially.

3.1 Artificial Intelligence, Machine Learning and Deep Learning

Artificial Intelligence (**AI**) refers to a specific branch of computer science that aims to give to machines, the ability to 'mimic' human-cognitive skills and carry out tasks in a human-way. **Machine Learning** (**ML**) is a subset of AI and is defined as the science that focuses on building systems that are able to perform a task by learning from data and without being explicitly programmed, unlike to what happens in the classic programming and computer science paradigms. **Deep Learning** (**DL**) is a subset of ML that aims to build mathematical models inspired by the human brain, so called neural networks, organized in different layers and where each layer calculates and provides the input for the next one.

3.1.1 Types of Machine Learning

Machine Learning algorithms can be classified according to various criteria, however, the most important classification is related to the type of data that models use and how they are supervised while learning:

Supervised Learning These kind of algorithms are fed with data that contain also the solutions of the problem (so called *labels*). Since the model's goal is to find a function to map input values with a target label, these systems learn by setting their parameters in a way that, once the training phase is concluded, they will be able to associate each input value with the correct target label.

Unsupervised Learning The data fed to unsupervised learning models are unlabeled, so in this case the model does not try to associate every input data with a label, instead the model's goal is to find recurrent patterns and hidden information in the data.

Semi-Supervised Learning The Semi-Supervised Learning algorithms stand between Unsupervised and Supervised Learning; these models use both (few) labeled data and (many) unlabeled data. Usually these models are trained, first, in an unsupervised manner and then, once all the labels have been created, in a supervised way.

Reinforcement Learning Unlike what happens in the three previous approaches, in reinforcement learning algorithms, the learning system, also called *agent*, try to learn how to perform a certain task by maximizing its reward. To do this the agent uses feedback coming from the external environment; good feedbacks will lead to an increase of the reward while bad feedbacks to a decrease of the reward.

3.1.2 Datasets

The key concept that emerges in the definition of Machine Learning at the beginning of this chapter is that an ML system *learns from data, without being explicitly programmed*. Thus, It is clear the crucial role of data used to build AI system. Usually, these data are divided into three different datasets:

- **Training set**: as we can easily guess, the model is trained and learn using the training set data.
- **Validation set**: the validation set is used to evaluate the model and to fine-tune and find the best hyperparameters, for this reason we could say that the validation set indirectly affects the training of the model.
- **Test set**: the test set is used to test the performance of a given model. It is used only after the model has been trained using both the training and validation sets to check how well the model performs with never-seen data.

3.1.3 Loss Function

A **Loss Function** is a mathematical function used to evaluate how well a ML model perform. When we finetune a ML model in order to achieve the best results, what we do is, simply, try to minimize as much as possible the loss function: the lower the function, the better the model's performance. To minimize as much as possible a loss function, ML models use the so called *optimizer algorithm* like the *Gradient Descent*, *Adam* and much more.
But a Loss Function is much more than a mathematical representation of a model's performance; they are also used to optimize an algorithm and to find the best parameters that fit the data. In other words, the loss function is an integral and crucial part of the training of a model.

The most common losses functions are:

- **Mean Squared Error** (**MSE**): also called *L2 Loss*, it measures the average squared difference between model's predictions (y) and actual observations ($\hat{y}$). It follows the mathematical formulation of the MSE:

$$MSE = \frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{n} \tag{3.1}$$

- **Mean Absolute Error** (**MAE**): also called *L1 Loss*, it measures the average of sum of absolute difference between model's predictions (y) and actual observations ($\hat{y}$). It follows the mathematical formulation of the MAE:

$$MAE = \frac{\sum_{i=1}^{n} |y_i - \hat{y}_i|}{n} \tag{3.2}$$

- **Cross-Entropy**: it is a loss often used in classification problems; it measures how much the predicted probability (y) of a label diverges from the actual label ($\hat{y}$), a.k.a. Ground Truth.

$$CrossEntropy = -\sum_{i=1}^{n} (y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)) \tag{3.3}$$

3.1.4 Problems of Machine Learning

Usually, all the things that can go wrong in ML and DL algorithms can be traced back to these two main problems, overfitting and underfitting (Fig. 3.1):

Overfitting Overfitting occurs every time the model performs very well on training data but it does not generalize well with other data. The main cause of overfitting is the high model complexity: if the model has too many features, the learned hypothesis may fit the training set well but it fails to generalize on the test set. The possible solutions to overfitting are:

- Simplify the model by reducing the number of features.
- Apply *regularization*, a technique that allows to keep all the features but reduce their impact.
- Feed the model with more training data.

Underfitting Underfitting occurs when the model has poor performance on both training and test data, and it is often caused by a model that is too simple and is not able to learn. The solutions to underfitting are:

- Increase the model complexity (i.e. increase the amount of information the model uses to learn).
- Increase the number of features (feature engineering).
- Clean training data, avoiding as much noise as possible.

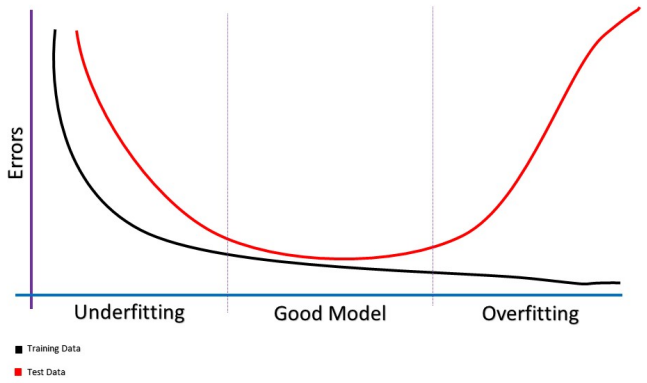


Figure 3.1: Underfitting and Overfitting
[8]

3.2 Neural Networks

The **Neural Networks** (**NN**) are a group of algorithms that try to mimic the human brain. In the last years they are becoming increasingly fundamental in the field of machine learning as they outperform the majority of classic ML techniques, especially in the resolution of the hardest tasks (e.g. computer vision). As in

the human brain, a Neural Network is composed mainly by 2 elements: *neurons* (computational units) and *synapses* (connections between neurons).

A neuron takes a certain number of parameters as input, multiplies them by weights, combines them together and finally applies a non-linear function called **activation function**. Figure 3.2 shows a simple neural network, so called **Multi-layer Perceptron** (**MLP**), composed by an input layer, an hidden layer and an output layer.

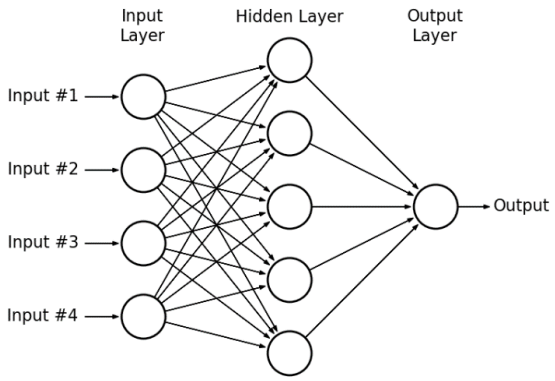


Figure 3.2: Multi-layer Perceptron, one of the simplest NN [9]

The number of neurons in each layer may vary, except for the level of input and output. The number of neurons of the input layer depends on the number of inputs the model takes; the number of neurons of the output layer depends on the number and type of output we expect (e.g. in a classification task the number of neurons of the ouput layer will be equal to the number of classes/labels we want to classify).

3.2.1 Convolutional Neural Networks

The **Convolutional Neural Networks** (**CNNs**) are among the most used neural networks, especially in the field of computer vision, however, they have been also successfully used at many other tasks. The basic building block of a CNN is the convolutional layer: the convolution operation allows to decrease the number of features of the model without losing information. As explained by A. Géron in [10], the convolution allows the network to concentrate on small low-level features in the first hidden layer, then assemble them into larger higher-level features in the next hidden layer, and so on. This hierarchical structure is common in real-world images, which is one of the reasons why CNNs work so well in performing computer vision tasks. The main layers of this type of NN are:

Convolutional layer This layer convolves the input data and it creates an output feature map. The parameters that defines a convolution are: the kernel, the padding and the stride. Given an input with the following shape: [input height n]×[input width m]×[input channels] and a convolutional filter defined by a *filter f*, a *stride s* and a *padding p*, the result of the convolution will be performed as shown in Figure 3.3, where the shape of the convolutions between input and the filters is:

$$[\frac{n+2p-f}{s}+1] \times [\frac{n+2p-f}{s}+1]$$

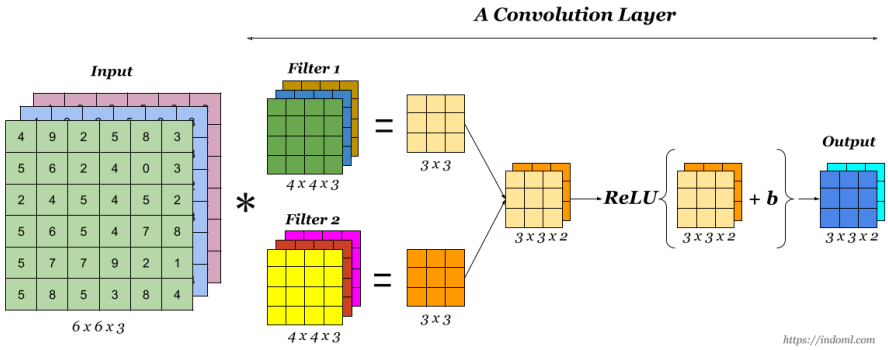


Figure 3.3: Convolutional layer [11]

Furthermore, a non-linear activation function is applied to reproduce a non-linear behaviour since the convolution is a linear operation. In Figure 3.3 the *ReLU* is the activation function used, but usually also the *sigmoid* or the *hyperbolic tangent (tanh)* are common functions used to introduce non-linear behaviours.

Pooling layer Pooling layers are used to significantly reduce the number of parameters (i.e. the dimension of the output feature map obtained by a convolution) of the CNN by dividing the feature map in several blocks of $n \times m$ shape. The most common pooling layers (Fig. 3.4) used are:

- **Max Pooling**: it returns the maximum value of each block of the feature map.
- **Average Pooling**: it returns the mean of each block's values of the feature map.

Fully Connected layer Usually some fully connected layers (i.e. MLP networks) are added at the end of a CNN in order to perform a non-linear combination of the features extracted in the previous layers and, lastly, classify the image.

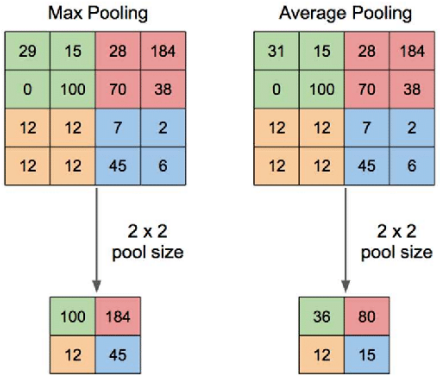


Figure 3.4: Pooling layers
[12]

3.3 ResNet and Residual Blocks

CNNs and NNs in general can easily become very deep and complex, furthermore, during the training of these networks can arise various problems related to gradients, such as vanishing (gradient approaches 0) and exploding gradients (gradient approaches ∞).

K. He et al. [13] were the first to tackle the complexity of the deepest NNs by using a novel approach based on new computational blocks called **Residual Blocks**. A Residual Block (Fig. 3.5) is composed by the so called **skipping connections**: 'shortcuts' that allow the forwarding of data obtained from a level to a deeper level by adding it to a linear component before applying a non-linear function like the ReLU. The NNs that implement the Residual Blocks are called **Residual Networks** (**ResNet**).

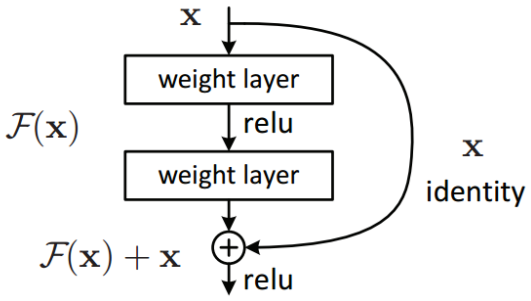


Figure 3.5: Residual Block
[13]

In Figure 3.6 is shown that the plain NNs have a performance degradation

problem as the number of levels increase and how ResNet strongly mitigates this phenomenon.

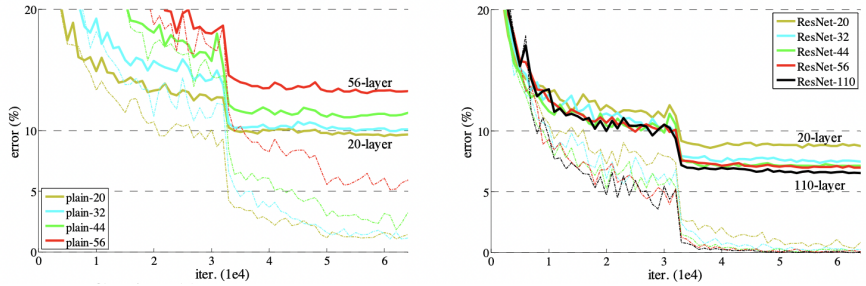


Figure 3.6: Performance degradation of Plain NNs and ResNets [13]

3.4 Super-Resolution

We have seen how the use of high-resolution satellite imagery turns out to be very useful for a number of practical applications such as environmental monitoring, city mapping, disaster management, and much more. Unfortunately we have also seen how the instruments aboard satellites are subject to very stringent constraints that in some cases can limit the spatial resolution of the images acquired by them; appropriate **Super-Resolution (SR)** techniques, however, allow us to obtain high-resolution (HR) images from one or more low spatial resolution (LR) images.

It becomes clear how the availability of many images of the same scene is particularly important since if properly combined they allow us to greatly increase the spatial resolution. In fact, when we talk about Super-Resolution applied to Remote Sensing, we must take into account the fact that there are two main methods of applying SR techniques:

- **Single Image Super-Resolution (SISR)**: these techniques use a single image to construct its super-resolution counterpart; however, the amount of useful information in the case of a single LR image is not very high, and the capacity of these models is therefore limited.
- **Multitemporal Image Super-Resolution (MISR)**: in this case multiple images of the same scene (i.e., multiple images of the same area taken at different times) are used to generate the super-resolution version of the image. The great advantage of this approach is due to the fact that by having multiple images of the same scene we will have much more useful information and also be able to increase the spatial resolution, at the same time, however, it

should be considered that using temporally distant images leads to having images that, in fact, often change (for example, due to lighting, clouds, human activity and others).

3.5 PIUnet

PIUnet (Permutation Invariance and Uncertainty estimation Network) is a neural network architecture presented by D.Valsesia and E.Magli in [14], for multitemporal super-resolution which is invariant to the temporal permutations of the images and that requires smaller dataset for training (in comparison with other models performing MISR). The new features proposed in PIUnet can be summarized as:

- Invariance to temporal image permutations.
- A second output, in addition to the SR image, called *uncertainty map* which estimates the uncertainty for each pixel of the SR image based on temporal variations in images and true error.
- The ability to take a variable number of LR images as input (previous architectures performing MISR take a fixed number of LR images, usually 9).

The dataset used for the training is a Proba-V dataset made available by ESA; the reason this dataset was chosen is that it has both 300m spatial resolution LR images (generated every day) and 100m spatial resolution HR images simultaneously. However, the HR images are generated every 5 days, making them limited in number. Each image in the dataset represents a Level 2A product, ergo radiometrically and geometrically corrected and quantified in TOA reflectance; the bands used are the NIR (near-infrared) and RED bands.The images are also preprocessed in such a way as to select only those LR images that, according to the relative status map, have less than 15 percent concealed pixels in them; in addition, the number of 9 LR images was fixed for each scene even though there are no constraints on the number of images to be used. Finally, the images were appropriately normalized by subtracting the mean intensity from the training set and dividing everything by the standard deviation.

3.5.1 Model architecture

The PIUnet architecture is shown in Figure 3.7. The model takes as input a stack of LR images and outputs two results: the uncertainty map (top output) and the super resolution image (bottom output); both of these outputs share part of the

network (backbone) although we can see that convolutional, Batch Normalization and Upsampling blocks are used to obtain the uncertainty map, while for the SR image a PixelShuffle and a skipping connection are used. We emphasize an important difference with other models that perform the same task (MISR), which is that PIUnet can take as input any number of LR images, vice versa many other MISR models take as input a fixed number of LR images.

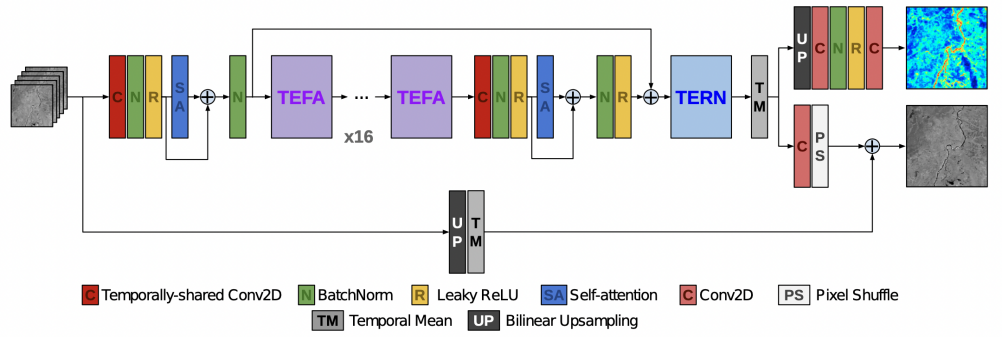


Figure 3.7: PIUnet Architecture [14]

Notice how the architecture is also composed by two modules: *TEFA* (Fig. 3.8) and *TERN* (Fig. 3.9), which will be discussed in depth in the next section.

3.5.2 Invariance to temporal image permutations

Before understanding how invariance to temporal permutations works, two concepts of algebra must be introduced.

Equivariance: A function $f : X \to Y$ is equivariant to the actions g of a group G if:

$$f(g \circ x) = g \circ f(x) \qquad \forall x \in X, g \in G \tag{3.4}$$

Invariance: A function $f : X \to Y$ is invariant to the actions g of a group G if:

$$f(g \circ x) = f(x) \qquad \forall x \in X, g \in G \tag{3.5}$$

As explained in [14], in PIUnet, we are dealing with the permutation group and its actions are all the possible temporal permutations of the input images, more specifically, with the invariance property the order in which we give LR images to PIUnet does not affect the SR image. If we have an invariant function, the output will always be the same, no matter the permutation of the input, while for

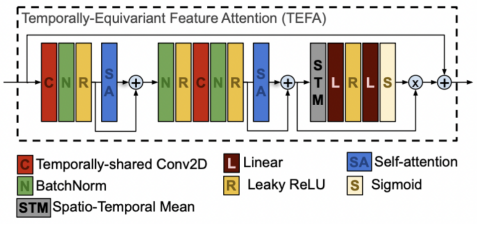


Figure 3.8: TEFA Module [14]

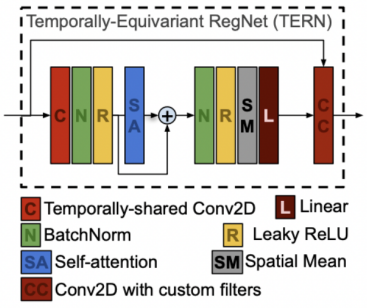


Figure 3.9: TERN Module [14]

equivariant functions we will get an output that is exactly a permuted version of the output we would get without the input permutation.

Being able to implement the invariance property in the layers is the key to obtain a model that is invariant to temporal image permutations. Unfortunately the mathematical operations that have the latter property are few (e.g. the mean), and often too simple for solving such a complex task. However, D. Valsesia and E.Magli proposed in [14] an alternative way to build an invariant neural network: the idea is to concatenate multiple equivariant operations (layers) followed finally by a global invariant operation. The equivariant operation that was chosen to implement the model is the *self-attention* [15].

The **Self-Attention** is an operation that uses three matrices to generate three vectors called: *key (K)*, *value (V)* and *query (Q)* and once obtained it calculates the cross-correlation matrix between key and query used later to appropriately weight the values of the value vector and generate the output. More formally, we define the self-attention function as:

$$\boldsymbol{Q} = \boldsymbol{X}\boldsymbol{W}_q \quad \boldsymbol{K} = \boldsymbol{X}\boldsymbol{W}_k \quad \boldsymbol{V} = \boldsymbol{X}\boldsymbol{W}_v$$

$$\boldsymbol{Y} = softmax(\frac{\boldsymbol{Q}\boldsymbol{K}^T}{\sqrt{T}}) \cdot \boldsymbol{V} = \boldsymbol{A}\boldsymbol{V}$$

Where **X** is the representation of a pixel with F features and T temporal channels. It can be mathematically proved in the above formula that a permutation of T corresponds to a permutation of the columns of A, ergo the self-attention function is equivariant. This operation is performed on all pixels of all images in the batches we train.

Therefore, in [14], the Self-Attention function was used to build the **TEFA (Temporally-Equivariant Feature Attention)** block (Fig. 3.8); the latter is an

extension of the classic residual feature attention proposed in [16] and is used in repetition as the backbone of the neural network. Specifically, the TEFA module computes attention scores to weigh the feature channels by extracting spatial and temporal features in an equivariant way, by means of shared 2D convolutions and temporal self-attention, and averaging them over space and time [14]. Now seeing the whole architecture (Fig. 3.7) we notice that all its blocks perform equivariant functions (to temporal permutations), consequently to make the whole model invariant, as we said before, we need to apply a global invariant function: we do this performing an average of the TERN output along the temporal axis.

As for **TERN (Temporally-Equivariant RegNet)** module, it is an extension of the RegNet module presented in [17]. The goal of the original RegNet was to dynamically compute small K $\times$ K spatial kernels from the input features to be used as filters over the input itself; The TERN module retains this function but has been modified in [14] to implement the temporal-equivariance property.

3.5.3 Uncertainty estimation

The second output of PIUnet is the so-called **Uncertainty Map** which is particularly useful since it gives us an estimation of the aleatoric uncertainty of the SR image, useful to judge the reliability of regions of SR image. Figure 3.10 shown a comparison between one of the LR images provided to PIUnet, the SR output and the uncertainty map; notice how the uncertainty map has higher value in areas with various and different spatial features.

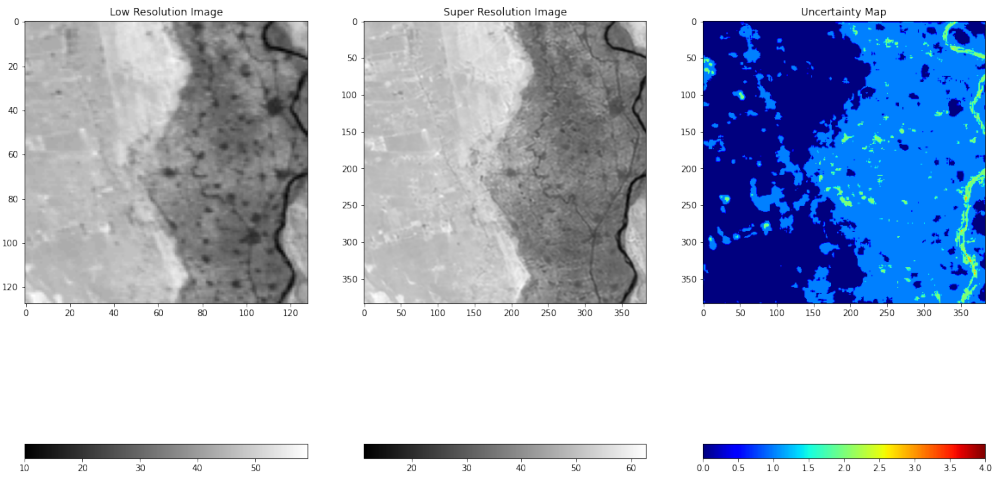


Figure 3.10: From left to right: LR image, SR image, Uncertainty Map

To estimate SR image uncertainty PIUnet focuses on *aleatoric uncertainty*, which is calculated as a function of perturbations on the input data such as noise

or, in this specific case, temporal variation in the images. Being able to measure this uncertainty allows us to determine when a portion of an image has been 'super-resolved' crudely and when there is some degree of certainty. To calculate the uncertainty we need to model each pixel of the SR image as a random variable and whose distribution can vary pixel by pixel. The **NLL (Negative Log-Likelihood)** is used as the loss function to be minimized where, however, the distribution of pixels is calculated using the following Laplacian formula:

$$p(x_i) = \frac{1}{2\beta_i} exp(-\frac{|x_i - \mu_i|}{\beta_i})$$

$$E[x_i] = \mu_i \qquad Var[x_i] = 2{\beta_i}^2$$

Where μ_i are the pixels of the SR image obtained in output, or more precisely the version where we compensate with the average brightness to handle the fact that the Ground Truth image and the SR image can have different absolute brightness values, and β_i which is proportional to the standard deviation and thus will represent our aleatoric uncertainty (from the second head of the network we in fact get $\delta = \log \beta$). Having made this assumption, the NLL loss can be generalized as if it is the following *L1 Loss*:

$$L = -\frac{1}{NB} \sum_{b,i} \log p(x_i) = \frac{1}{NB} \sum_{b} \left[\sum_{i} \left({\delta_i}^{(b)} + e^{-\delta_i^{(b)}} \left| x_i^{\mathrm{HR}(b)} - \mu_i^{(b)} \right| \right) \right] \tag{3.6}$$

Where $i = 1...N$ are the pixels and $b = 1...B$ are the images; also notice that μ and $\delta = \log \beta$ are the two outputs of the model, respectively the SR image and the aleatoric uncertainty.

3.5.4 Results

The metric used to evaluate the results of PIUnet is a variation of the classic **PSNR (Peak Signal-to-Noise Ratio)**, the **cPSNR (Corrected PSNR)**, which is insensitive to absolute brightness and takes into account the pixel shifts between the SR image and the Ground Truth. It follows the definition of the cPSNR:

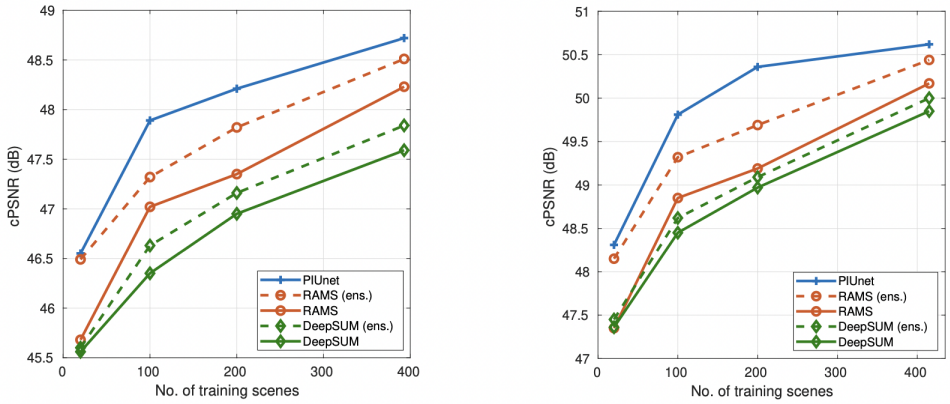
$$cPSNR = \max_{u,v \in [0,6]} 10 \log_{10} \frac{(2^{16} - 1)^2}{MSE_{u,v}} \tag{3.7}$$

Where $MSE_{u,v}$ is:

$$MSE_{u,v} = \frac{||x^{\mathrm{HR}(u,v)} \odot m - (x^{\mathrm{SR}} \odot m + b \odot m)||_2^2}{||m||_1}$$

In the latter m represent the quality masks, x^{HR} is the high-resolution image, x^{SR} is the super-resolved image, b is the bias calculated to make the PSNR insensitive to absolute brightness differences, finally u and v indicate the amount of horizontal and vertical shift applied to the HR image. Notice that in the previous formula the $\odot$ symbol denotes an elementwise product.

In Figure 3.11 is shown the performance comparison between PIUnet and other state-of-the-art models like DeepSUM [17] and RAMS[18].



(a) NIR (b) RED

Figure 3.11: Performance as function of no. of training scenes [14]

Furthermore, Figure 3.12 shows an example of the two outputs of the network, respectively the SR Image and the uncertainty map, and the Ground Truth (HR Image).

Notice how in the uncertainty map (Fig. 3.12), the highest aleatoric uncertainty values, denoted by warmer colors in the color map, are near areas where spatial and radiometric features are particularly variable.

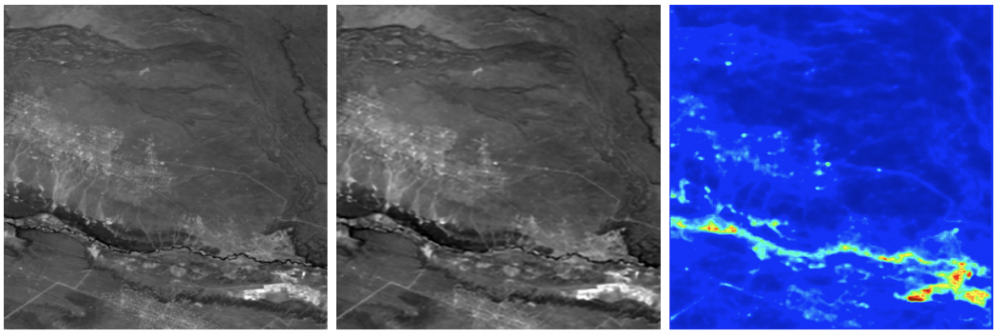


Figure 3.12: From left to right: HR Image, SR Image, Uncertainty Map [14]

Chapter 4

Datasets

One of the most important and time-consuming part of this work was related to the creation process of the dataset needed for this thesis. This chapter provides an in-depth analysis of the latter and its creation processes; since in order to create the final dataset it was first necessary to create two other datasets, one from Proba-V data and one from Sentinel-2 respectively, this chapter will also describe the process that led to the creation of the previous two and finally to the creation of the final dataset.

4.1 Datasets specifications

After analysing the problem, we defined the specification of the dataset:

- The new dataset must contain images from both Sentinel-2 and Proba-V missions. Those from Proba-V must be used as LR images, while those from Sentinel-2 must be used as Ground Truth.
- Obviously, the images (LR and HR) must represent the same area and must be coregistered in such a way that the shifts between the images are, at most, of 3/4 pixels.
- Since the images used with PIUnet measure the TOA reflectance, the images in the new dataset must be quantified in TOA reflectance, consequently for the Proba-V mission the data need to be extracted from level 2A, while for the Sentinel-2 mission the data need to be extracted from level 1C.
- The bands on which we will work on are the RED and NIR bands; specifically for each scene, the same image is extracted from both bands.
- For each scene, there must be at least 16 LR images and one HR image.

- Images should be as clean as possible; ideally LR images should have at least 70% clear pixels while HR images should have at least 85%.

- Acquisitions should not be time-spaced more than two months apart, otherwise there could be substantial spatial and radiometric differences due to various phenomena, both natural (e.g., change of season) and man-made (e.g., cultivation, buildings construction, etc..). Ideally, one would choose the latter as small as possible for the previously mentioned reasons, but since the availability of data varies a lot and high-quality images (i.e. with few dirty pixels) are quite rare, especially in certain areas, we decided to enlarge it and set it at 2 months. This 2-months time-window is, in our opinion, the minimum necessary trade off so that there are at least 16 downloadable LR images of the same area and at the same time with good quality.

I also specify that, following the same convention to the one already adopted in the European Space Agency challenge addressed in [14], the LR images have dimension of 128x128 pixels with a spatial resolution of 300 meters, following Proba-V specifications, consequently covering an area of about 38400 meters; while the HR images are 384x384 pixels in size with a spatial resolution of 100 meters, thus as before covering an area of 38400 meters per side.

4.1.1 Selections of Regions of Interest

The first step necessary to define the dataset was to choose the coordinates of the scenes, also known as *Regions Of Interests (ROI)*, to be acquired. For this purpose I selected, manually, 636 different coordinates from all over the world. Since both the NIR and RED bands had to be downloaded for each scene, the total number of different sets of images (*imgsets*) in the dataset is 1272. It is crucial to emphasize that these latter coordinates were not chosen randomly; rather, criteria were followed with the aim of obtaining as heterogeneous a dataset as possible, with the presence of a wide variety of biomes and spatial features, thus including coastal areas, urban settlements, deserts, vegetation-rich areas, mountains and more.

In addition to the previous criterion, strong consideration was also given to the fact that the images should contain as few clouds as possible, which is why acquisitions made in particularly sunny months, and in areas of the Earth where the presence of clouds is often reduced, were chosen.

Figure 4.1 shows a map of the world where the positions of the previously mentioned coordinates are identified with a cross.

The list of coordinates, and other useful information, regarding the acquisitions chosen for the new dataset can be found in Appendix A.

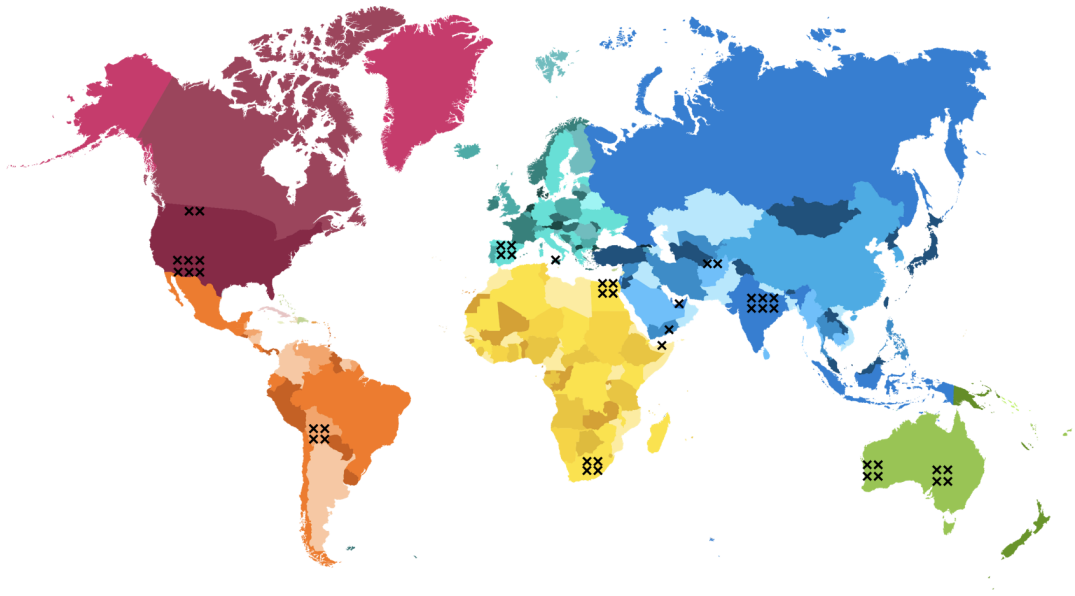


Figure 4.1: Coordinates of the 636 scenes

4.1.2 Image quality

As we have seen in previous chapters, for the model to work well, it is essential that the images are reliable and have high pixel quality. In particular, it is necessary that in addition to the LR and HR images, information about each pixel in the images should also be provided to the model to understand whether it is a *clear* or *dirty* pixel; in the rest of this thesis I will refer to the latter as **Quality Map (QM)** (Fig. 4.2). Since each *dirty* pixel corresponds to a loss of information then it is critical that the images in the dataset have as few *dirty* pixels as possible.

Following a similar, but not the same, approach to the one followed in [19], I decided to compute and add to the datasets (of both Proba-V and Sentinel-2), for each image, its corresponding Quality Map. More specifically I downloaded, for both missions, all the status map (Fig. 2.11 represents the Proba-V's status map notation), scene classification map and all the files containing acquisition quality metrics such as radiometric quality, geometric quality, saturated pixels, burned pixels and so on.

Finally, I computed the quality map for each LR and HR image following these general rules:

- Pixels identified as clouds, cloud shadows or undefined in the status maps of a certain image are set as *dirty* in the corresponding quality map.
- Using files with image quality information, I identified saturated, burned, and

low-quality pixels, both radiometric and geometric, as *dirty* in the corresponding quality map.

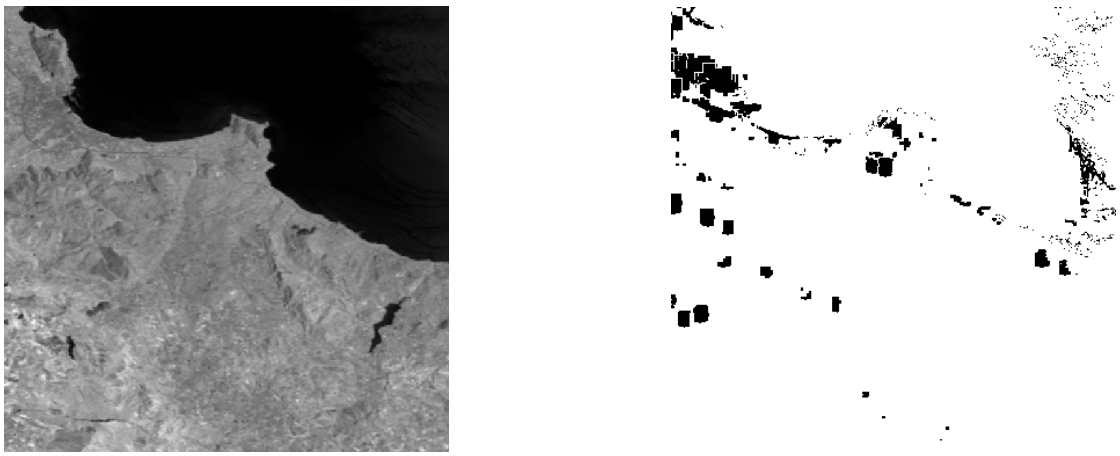
- All other pixels that do not fall into the previous two cases were identified as *clear* in the quality map.

Following the same convention adopted in [19], I defined the **clearance** of an image as the ratio of *clear* pixels to the total number of pixels in an image:

$$Clearance\ (\%) = \frac{no.\ clear\ pixels}{tot.\ image\ pixels}$$

With *tot. image pixels* that can be equal to 16.384 (128×128) in the case of LR images and 147.456 (384×384) in the case of HR images.

Figure 4.2 shows an HR image (4.2a) from the Proba-V dataset and its corresponding quality map (4.2b), computed following the previous rules.



(a) HR Image **(b)** Quality Map

Figure 4.2: HR Image and its corresponding quality map; black pixels (i.e. value = 0) represent *dirty* pixels while the white ones (i.e., value = 1) represent *clear* pixels. The image was extracted from the RED data band at the coordinates *(latitude, longitude)* of *38.21130, 13.31250* in May 2020.

4.1.3 Dataset structure

To summarize, I ended up creating a dataset consisting of 1272 *imagesets*. Each *imageset (imgset)* represents the same geographical area and contains the following:

- At least 16 low-resolution Proba-V images (16 bits grey scale) of size 128×128 pixels (Fig. 4.3).

- For each low-resolution image, the corresponding quality map of the same size. The quality map is a binary image, so it encodes only two possible values: 1 for *clear* pixels, 0 for *dirty* pixels.

- One single high-resolution Sentinel-2 image (16 bits grey scale) of size 384×384 pixels. This image is used as Ground Truth for the Super-Resolution task.

- A single quality map relative to the HR image. As with LR images, this quality map is a binary image of the same size as the HR image, in which *dirty* pixels are identified with 0 and *clear* pixels with 1.

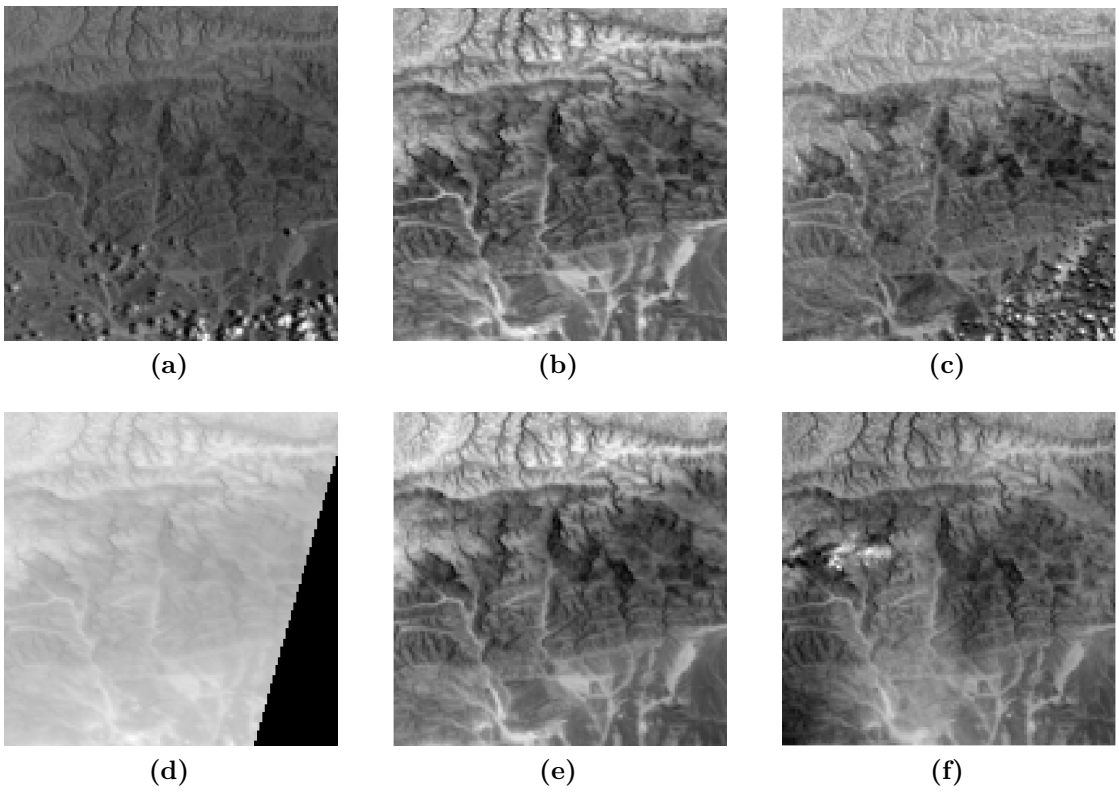


Figure 4.3: Different LR images representing the same area extracted at different times.

4.2 Data collection

As mentioned before at the beginning of this chapter, to create the dataset needed for this thesis the idea was to create and then merge two smaller datasets, one obtained from the Proba-V data and containing the LR images, and the other containing the corresponding HR images obtained from the Sentinel-2 data.
In this section is explained the process of creation of the two datasets already mentioned.

4.2.1 Proba-V

To collect Proba-V data and create the related dataset I relied on the terrascope[1] platform, specifically using one of the virtual machines provided by them. In the latter, all data in the time frame from 2014 to 2020 from the Proba-V mission in Europe are available, while less data are available outside European soil in the same time frame.

To collect the LR images I used the products available at product level L2A, which, as previously explained in Chapter 2, is composed of radiometrically and geometrically corrected Top-Of-Atmosphere reflectances in Plate-Carré projection. These products are saved in the HDF5 format in the Terrascope VM; the HDF5 format is a format that allows particularly complex and heterogeneous data to be grouped into an intuitive and easy-to-manage hierarchical structure, all of which made Proba-V's data extraction work relatively easy to do. Specifically, part of the hierarchical structure of the latter is presented in Table 4.1; for the sake of clarity and simplicity I specify that in Table 4.1 I report only part of the hierarchy of the HDF5 file, in particular, showing the attributes and data used for data extraction and not reporting instead all the data that were not used. Further information about the format and Proba-V data and the HDF5 format are available respectively in the Terrascope website and in the HDF Group Website[2].

LR Images extraction Regarding the extraction and collection of LR images, I made a script that, taking as input a latitude, a longitude and a time interval of two months, would search among all the 300-meters spatial resolution Level2A products present in the Terrascope VM those that within them had the latitude and longitude given as input and in the specified time interval; once these products were found I would go to check the quality of the image (more information in the next paragraph) and finally if the quality checks were passed patches of size 128x128

[1] https://terrascope.be/en

[2] https://www.hdfgroup.org/solutions/hdf5/

Name	HDF5 Path	Description
lat	/lat	Dataset containing the latitudes
lon	/lon	Dataset containing the longitudes
Status Map	/LEVEL2A/QUALITY/SM	Dataset containing the quality flags and status pixels
TOA RED	/LEVEL2A/RADIOMETRY/RED/TOA	Dataset containing TOA reflectances
TOA NIR	/LEVEL2A/RADIOMETRY/NIR/TOA	Dataset containing TOA reflectances

Table 4.1: Hierarchical structure of Proba-V HDF5 files.

were extracted, using the input latitude and longitude as left-upper starting pixel of the image. In the indicated time period as many images as possible were extracted, again if the quality control is passed, in such a way as to have as many LR images as possible; in case more than 15 LR images could not be extracted within the indicated time period then the script was aborted with an error report.

This procedure was applied for both NIR and RED bands; in addition, although not strictly required and not necessary, I also extracted for each imageset the corresponding Proba-V image in HR (i.e., with spatial resolution of 100 meters). As we will see later, the HR images from Proba-V, will also serve us to calculate the cPSNR using the Proba-V images as ground truth.

Quality Check and generation of QMs Since the L2A products are already geometrically and radiometrically corrected the quality checks were quite straightforward to perform. To create the quality map of an image I checked in the corresponding status maps which pixels had the label *undefined*, *cloud* and *shadow*, then I created a matrix by setting to 0 (dirty) the pixels that fell into the 3 previous categories and to 1 (clear) all others. From this last matrix I generated a binary image, i.e., the quality map.

Finally, I extracted only LR images that had a *clearance* $\geq 70\%$, while in the case of HR images this threshold was set at 85%.

4.2.2 Sentinel-2

While creating the Proba-V dataset was a straightforward process, creating the Sentinel-2 dataset was more complicated for a number of reasons; I list the main ones below:

- The data format of Sentinel-2 data is .SAFE; this format does not group all the useful information into a single easy-to-open file (which is what happens in HDF5), instead the information and data are distributed among a hierarchy of folders where there are multiple files and extensions (such as .tif, .jp2, .shp and more), depending on what type of information the specific file contains. The naming convention for the Sentinel-2 is presented in Figure 4.4 and Table 4.2. More information about Sentinel-2 data can be found in [20].
- For both Level-1C and Level-2A, data are distributed in $100 \times 100 km^2$ adjacent tiles. This turned out to be the main problem since it often happens that the HR image to be extracted was distributed over several tiles that need to be merged before (i.e., the image of the area we are interested in extracting is not present entirely within a single tile, but on multiple adjacent tiles). To solve this problem and merge multiple tiles graphically, I had to implement a *mosaicking procedure*.
- Another problem was related to the coordinates system, in fact in Proba-V data it was possible to locate points by latitude and longitude, while in Sentinel-2 the images are projected according to the UTM system (Universal Transverse Mercator, Fig. 4.5). The UTM is a coordinate system that divides the world, except for the polar zones, into 60 different regions each identified by a number and a letter.

<MISSION>_<DATE>T<UTCTIME>Z_<GRIDID>_<CONTENT>_<RESOLUTION>_V<VERSION>.tif

Figure 4.4: Sentinel-2 products naming convention

To extract and collect the Sentinel-2 data I used an EarthConsole[3] Virtual Machine.

HR Image extraction Again it was necessary to make a script for data extraction. The procedure, and so the script, to extract HR Image from Sentinel-2 data was very similar to the one described previously for the Proba-V Dataset, with two main differences:

[3]https://earthconsole.eu

<MISSION>	Mission ID (S2A, S2B)
<DATE>	Start date of acquisition in YYYYMMDD format
<TIME>	Start time of acqusition in hhmmss format
<GRIDID>	ID of the tile in UTM projection
<CONTENT>	Content of the file (i.e. TOC, TOC-B02, TOC-B07..)
<RESOLUTION>	Spatial resolution of the product
<VESRSION>	Workflow's version (last is v200)

Table 4.2: Sentinel-2 products naming convention

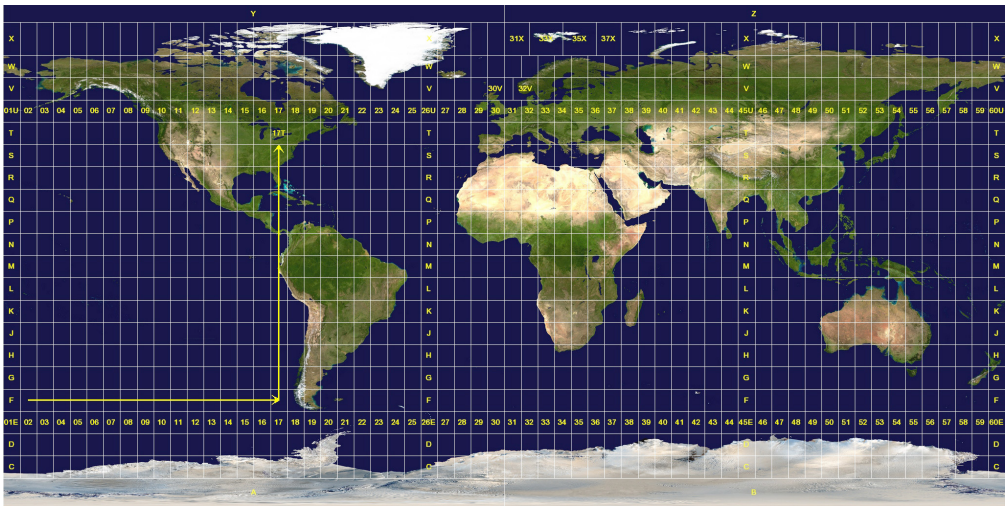


Figure 4.5: UTM Coordinate System
[**Credit: Wikipedia**]

- **Latitude/Longitude-UTM Conversion**: Since the Sentinel-2 data are projected accordingly to the UTM system I could not directly use latitude and longitude to locate the searched point; rather, it was necessary to convert the coordinates given in latitude and longitude to UTM so that I could then search for the target area. To perform this conversion I used the utm[4] python library.

- **Mosaicking**: As mentioned earlier one of the main problems was due to the tile size of Sentinel-2, in fact it often happened that an image to be extracted was located on multiple tiles (Fig. 4.6). Obviously in this case it was necessary to perform a merge between the various tiles that 'contained' the area of the

[4]https://github.com/Turbo87/utm

HR image to be extracted. To do this I included in the Sentinel-2 generation script some additional methods that had the purpose of: identifying if the area to be extracted was entirely on one tile, if it was not on one tile finding the adjacent tiles needed to observe the whole area, once found it was necessary to merge them and finally to crop only the area of our interest to generate the HR image.

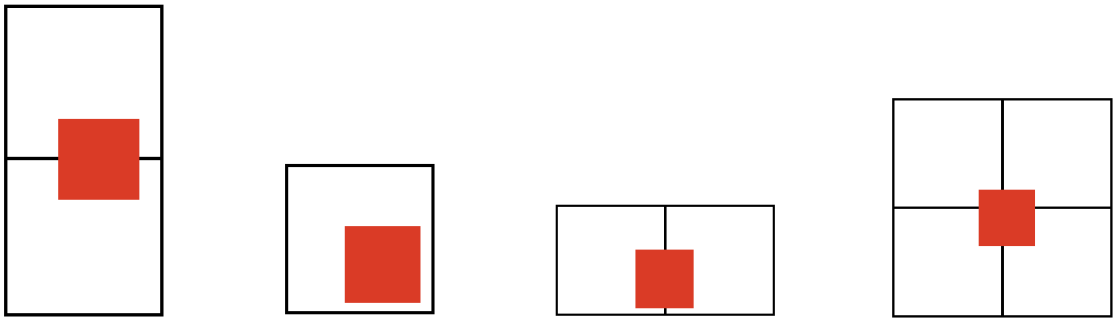


Figure 4.6: Visual representation of the mosaicking problem. The red square represents the area to be extracted (HR Image) while the squares with the black perimeter represent the Sentinel-2 tiles.*(a).* HR Image is located within two tiles; *(b).* HR Image is entirely inside just one tile; *(c).* HR Image is located within two tiles; *(d).* HR Image is located within four different tiles.

Quality Check and generation of QMs In order to perform quality checks with the sentinel 2 data, I acted slightly differently from what I did with the Proba-V dataset, in fact the Level 1C products of Sentinel-2 are a bit rougher than the Level 2A products of Proba-V. Specifically, to create the quality maps, I used all available files that contained quality information (*CLOUDS.shp* for clouds, *SATURATED.shp* for saturated pixels, and so on). In this way an early version of the quality map was defined, which was later improved by also using Sentinel-2's Level 2A quality files (*when available on the EarthConsole VM*), which contained more accurate information, and the scene classification maps. Using the quality data from Level 2A was made necessary in some images because the Sentinel-2 level 1C data are poorly refined and in some cases there was no quality information present (i.e., the quality files are empty), with the only exception for the cloud information (contained in the file CLOUDS.shp) that is always present even from level 1C, although not very accurate. Obviously the Level 2A data were used only for quality control, the extraction of the images was only done at Level 1C where the TOA reflectance is measured. Finally, in order to generate the quality map for a Sentinel-2 image I followed the following rules:

- All the saturated, 'dark-currents' and undefined pixels were identified as *dirty* in the corresponding quality map.
- All the pixels representing clouds and shadows were identified as *dirty* in the corresponding quality map.
- The remaining pixels that did not fall into the previous two categories were identified as *clear*.

The images, compared to Proba-V dataset, have on average a higher cleanliness; in fact, in the script I inserted a constraint such that to select and generate an image 92% of its pixels must be identified as *clear* (previously it was 85% in Proba-V). This choice is due to the fact that by then applying co-registration (more details in the next section) to the Sentinel-2 images the resulting image contained black side stripes, effectively decreasing the percentage of clean pixels, consequently to maintain a high quality of the Ground Truth images I decided to make this condition more stringent.

Downsampling Since the HR images of Proba-V originally used in PIUnet had spatial resolution of 100 meters (RED and NIR bands have 100m spatial resolutions), while the HR images of Sentinel-2 have spatial resolution of 10 meters, it was necessary to apply a *downsample filter* to the Sentinel-2 imagery so as to bring their spatial resolution to 100 meters. This is necessary for two reasons:

- To keep the problem specifications as close as possible to the original ones.
- PIUnet decreases the spatial resolution by a factor of 3 (from 300m to 100m), if one were to leave the HR images at 10m size the spatial resolution would have to decrease by a factor of 10 which is very difficult both because the improvement is really high, it is also difficult because of a computational aspect related to the number of model parameters and the limitations of the GPUs used for training.

To perform the downsampling operation, the *Image.resize()* method of the PIL python library[5] was used. This method, as we can guess from the name, returns a resized copy of the image; one of the optional parameter we can pass to the latter is *resample*: this parameter allows us to choose an optional resampling filter. Since the operation we wanted to perform is a downsampling, I chose the *LANCZOS* low-pass filter since it has the highest downscaling quality of all the filters available, at the expense of performance.

[5]https://pillow.readthedocs.io/en/stable/reference/Image.html

4.3 Final Dataset

The final dataset was created by merging the data from the previous two datasets, using Proba-V images as LR images and Sentinel-2 images as Ground Truth. Some operations were necessary to make the dataset correct, in particular, a process of reprojection and finally coregistration were carried out.

4.3.1 Reprojection

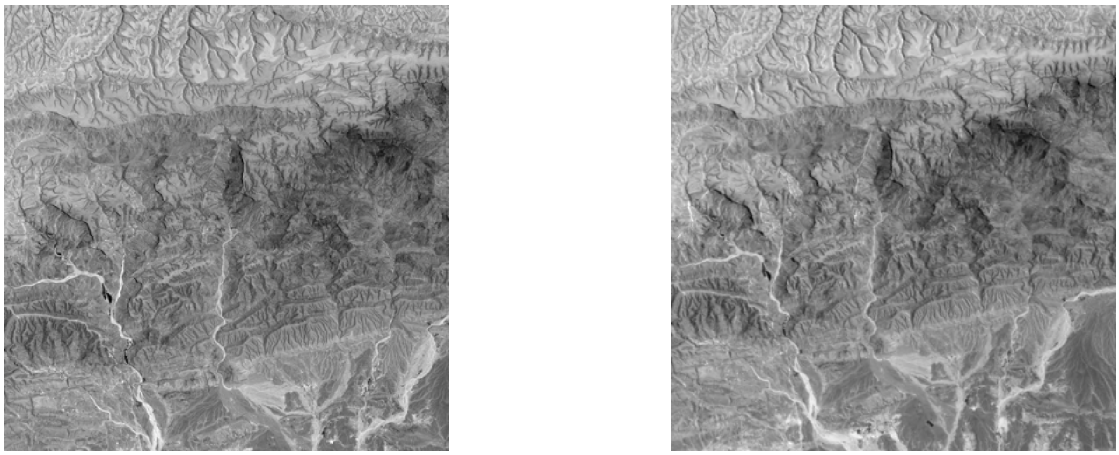
We saw in Chapter 2 that the two missions, Proba-V and Sentinel-2, have different specifications and characteristics; one of the differences between the two missions is the *projection system*, in fact in Proba-V the data are projected using the Plate-Carré projection system while in Sentinel-2 the UTM system is used. Before talking about reprojection, however, let's take a step back and understand what and why the projection system is so important.

Projection A projection is a mathematical transformation that takes spherical coordinates, usually latitude and longitude, and transforms them into an *(x, y)* coordinate system. This makes possible to create a map that accurately shows distances, areas, or directions. With this information, it is possible to work accurately on the data to calculate areas and distances and measure directions.

When working with satellite data, it is important that all images are projected using the same projection system, in fact, using different projection systems leads to displaying the data differently (even if the target area is the same), which is inconsistent, and therefore incorrect. One can visualize this problem in Figure 4.7 where the same area (the images are taken by using as reference the same coordinates on the upper-left angle and by extracting 384 pixels per size) of the earth is shown but projected using two different projection systems. From the latter it can be seen that although the upper-left corner has the same coordinates and both images have the same size of 384×384 pixels, one of the two images, the one projected in Plate-Carré, 'contains' a larger area than the other. Clearly, this needs to be corrected since the images that PIUnet will need to use must necessarily be the same; otherwise, the model will fail to perform the Super-Resolution task correctly.

Finally, to solve this problem, I decided to reproject, in the Sentinel-2 image extraction script, the Sentinel-2 images by changing the projection system from UTM to Plate-Carré. To do this it was necessary to use python's rasterio[6] library.

[6]https://rasterio.readthedocs.io/en/latest/index.html



(a) UTM Projection **(b)** Plate-Carré Projection

Figure 4.7: Image extracted from the RED data band at the coordinates *(latitude, longitude)* of *15.2321419, 49.4107126* (upper-left corner). *(a).* Sentinel-2 Image in UTM Projection; *(b).* Proba-V Image in Plate-Carré Projection. Note that the upper left coordinates are the same for both images and also have the same dimensions (384x384 pixels), but the Plate-Carré projection image *b* represents a larger area, this is due to the different projection system.

4.3.2 Co-registration

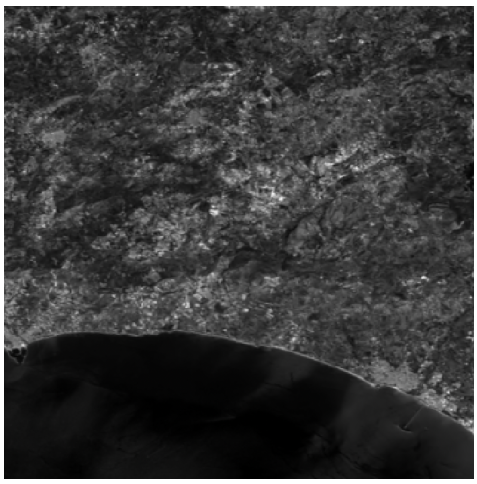
I have already introduced the concept of co-registration in Chapter 2. For the creation of the final dataset, it was necessary to apply co-registration between Sentinel-2 and Proba-V images so as to obtain images that are as geometrically aligned as possible and whose pixel shifts are as minimal as possible.

To perform the co-registration task I used arosics[7]: a python library that is specialized in performing automatic subpixel co-registration of two satellite image datasets. Unfortunately, because of the optical and geometric differences between the two satellites, I was not able to co-register all images using the latter's functions, but only those that differed by a few pixel shifts; therefore, it was necessary to create a special script for co-registering images to be used in all other cases (large shifts between pixels). In the end, the goal defined in the dataset specification was achieved and all images, after co-registration, did not differ by more than a couple of pixels. The only problem that arose as a result of co-registration was that in order to geometrically align the images it happened that the co-registered image often, not always, contained black side stripes actually decreasing the quality of

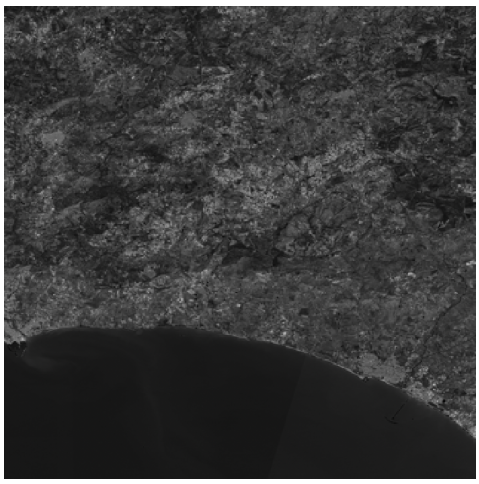
[7]https://github.com/GFZ/arosics

the image, which is why as mentioned before I inserted a constraint to extract as Ground Truth only images with *clearance* $\geq 92\%$.

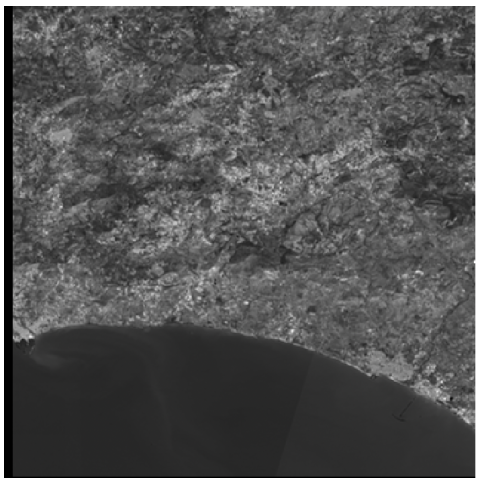
What is described in this section is shown in Figure 4.8.



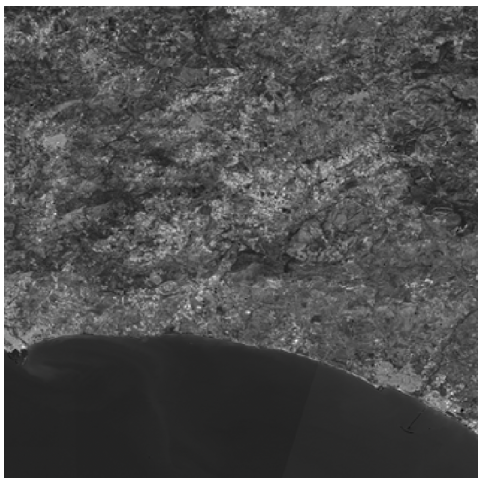
(a) Proba-V HR Image



(b) Sentinel-2 HR reprojected Image



(c) Co-registered image (arosics)



(d) Co-registered image (my script)

Figure 4.8: Co-registration process. In this specific scene the coregistration performed by the arosics (c) package was not enough reliable so it was necessary to use my co-registration script (d) to obtain a correctly co-registred image.

Chapter 5

Methods and Trainings

This chapter presents the modifications made on the architecture of PIUnet, in addition, experiments and trainings performed on both the standard version of PIUnet and the modified version are described.

5.1 Modified PIUnet

Doing Super-Resolution tasks by taking images from two different satellites, i.e., sensors with different characteristics is not easy for a number of reasons. One of the main ones, which will be further discussed in detail in later sections and chapters of this thesis, is that the model trains using low-resolution images from a certain satellite, in this case Proba-V, while the loss function is minimized by decreasing the error by using Sentinel-2 imagery as reference; this leads the model to create images that do not preserve the radiometry of either satellite, standing in the middle between the features of the latter two. For the former reason, taking a cue from the work done by M.T. Razzak et al. in [21], some changes were applied to the original architecture of PIUnet.

5.1.1 Consistency Loss

In the original implementation of PIUnet, the only loss function used is the *L1 Loss* (Eq. 3.6) shown earlier in Chapter 3; the former's main purpose is to generate output that is as similar as possible to Ground Truth images, then to Sentinel-2 imagery, but strong consideration must be given to the fact that the LR images used to train PIUnet are Proba-V images, thus with different radiometric and spatial characteristics. This same point was made by M.T. Razzak et al. in [21], who investigated in their work how different characteristics related to the optical sensors from which images are captured can lead to inconsistent results from a radiometric

point of view. Therefore, using an approach similar to that proposed in [21] an additional loss, called *consistency loss*, was added to the standard architecture of PIUnet. The idea was to obtain a Super-Resolved image that is radiometrically-consistent and does not have features that are somewhere between the LR images from Proba-V and the HR image from Sentinel-2.

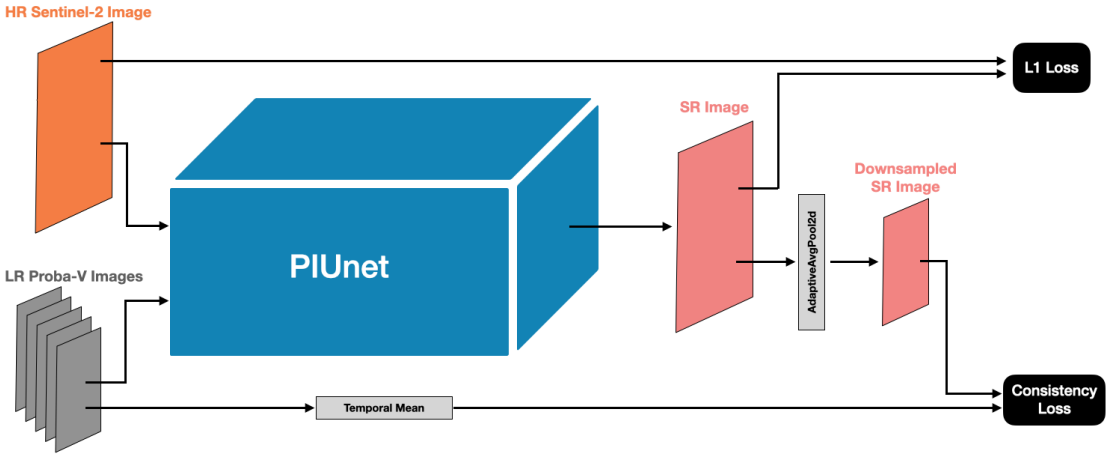


Figure 5.1: Modified version of PIUnet with the additional consistency loss.

In the modified PIUnet architecture, the SR image produced as output (together with the uncertainty map) is not only used to compute the L1 loss (upper branch of Figure 5.1), but it is then downsampled in such a way that its size is decreased and made the same as that of the LR input images, thus 128×128 pixels; the downsample is performed using a *2D adaptive average pooling* layer (more information on *AdaptiveAvgPool2d* are available PyTorch documentation[1]). At the same time, a temporal mean of the LR images used as input for the model is performed (lower branch of Figure 5.1); finally, the downsampled SR image and the result of the temporal mean of the LR images are used for the calculation of consistency loss.

For what concern the calculation of the original PIUnet loss, it remains unchanged from what was described by E. Magli and D. Valsesia in [14] (see section 3.5.3), while, about the consistency loss, it was implemented as a **corrected L1 loss** computed using as Ground Truth the result of the temporal mean of the LR input images and as predictions the downsampled version of the SR image, and which by similar reasoning to that used in equation 3.6, would take into account a bias factor given by the difference in brightness between the images; specifically, before computing the classical L1 loss formula (see equation 3.1), the SR image was corrected by adding to each pixel of the latter a value (b) that compensates the

[1]https://pytorch.org/docs/stable/generated/torch.nn.AdaptiveAvgPool2d.html

bias due to difference in brightness between the two inputs:

$$b = \frac{1}{128^2}(LR\_temporal\_mean - SR)$$

Furthermore, before calculating the consistency loss, normalization of the data is performed so as to bring them into the range [0, 1]; the normalization performed on the data is the L_2 *normalization*, already implemented in the PyTorch library[2], illustrated in the following formula:

$$t = \frac{t}{\max(||t||_2, \epsilon)}$$

Where t is the tensor input and ϵ is a small value (i.e., e^{-12}) needed to avoid division by 0.

Finally, to combine two (or more), losses into a total loss there are several alternatives: they can be summed, they can be averaged, they can be weighted with appropriate weights, just to mention some of the most popular methods used in the literature. In this case, after some experiments made to figure out the best way to combine the latter two, a simple subtraction was chosen as the operation to combine the two losses mentioned above. So the total loss resulted in:

$$total\_loss = PIUnet\_original\_loss - consistency\_loss \tag{5.1}$$

5.1.2 Regularization

In order to avoid overfitting problems, a regularization technique was implemented in the modified version of PIUnet; more specifically, the *weight decay* technique was chosen. *Weight decay* is a popular technique that is used to regularize the weights of the parameters of a certain model, and it works by penalizing the weights accordingly to their L2 norm. In our specific case, applying the weight decay to the new PIUnet architecture, the final loss can be described as:

$$loss = total\_loss + \frac{\lambda}{2n}\sum_{i=1}^{n} {w_i}^2 \tag{5.2}$$

Where λ is a manually chosen regularization hyperparameter.

[2]https://pytorch.org/docs/stable/generated/torch.nn.functional.normalize.html

5.1.3 Versions

For the sake of clarity and completeness, it is necessary to say that multiple modified versions of PIUnet have been implemented. The one described above represents the architecture of PIUnet from *version 2* onward.

Specifically, *version 1*, the first modified version of PIUnet, was different from version 2 (and later) for the following reasons:

- The consistency loss was not implemented in the same way as the corrected L1 loss we have just seen, but it was implemented, taking a cue from [21], as a classical MSE loss calculated without taking into account bias due to brightness difference between. However, the inputs were the same: the temporal mean of the input LR images and the downsampled SR image.
- The two losses, the PIUnet original loss and the consistency loss, respectively, were not combined by subtraction as seen in equation 5.1, but were multiplied by 0.1 and 0.9, respectively, and then combined by a weighted average operation.
- No regularization techniques were present, unlike in later versions.

For what concern the other versions, from version 2 onward, they all maintain the same architecture, which is the one described in the previous paragraphs, the only difference is the different choice of the regularization parameter λ, which changes from version to version. In table 5.1 are summarized the various modified versions of PIUnet made during this thesis.

Version	**Consistency Loss implementation**	**λ value**
1	Mean Squared Error	0 (No regularization)
2	Corrected L1	0.01
3	Corrected L1	0.0001
4	Corrected L1	0.000001
5	Corrected L1	0.0000001
6	Corrected L1	0.00000001

Table 5.1: Modified versions of PIUnet

In the remainder of this thesis, *version 1* will no longer be discussed because after some preliminary testing, it was found to be ineffective in training, where the loss function was not minimized correctly, and for that reason the network was never able to learn anything useful, i.e., perform the super-resolution task correctly; for the former reason, *version 1* was not even tested. Conversely, the

most interesting results and experiments were those obtained from *version 2* and *version 3*, which is why more emphasis will be placed in the analysis and discussion of the previous two versions.

5.2 Pre-processing step

In the next sections experiments and trainings performed on the model are described in depth, so it is important to mention that before any training was performed, either of the standard PIUnet architecture or the modified one, the data in the dataset used for training were processed appropriately through the use of a notebook very similar to those used and presented in [18] and [14]. The main steps of the latter are:

- **Dataset loading**: in this step each image and the corresponding quality map of every imageset are converted in tensors and added to two arrays, one for the LR images and one for the quality map.
- **Dataset registration**: a first step of registration among the images had already been done in the dataset creation phase (section 4.3.2), in this step we are going to perform a second registration of all the images in each individual imageset. Specifically, among the many available LR images, the one with the best quality is taken, checking the *clearance* obtained from the quality mask, and used as a reference to co-register all the others. This operation is performed for each imageset of the train, validation and test sets.
- **Selection of the best *n* images**: in this step, for each imgset, only the best n images are kept. The value n corresponds to the number of LR images we want to use as input, and represents the number of the temporal dimension of the images (which we recall is usually 9 in MISR models but in PIUnet is a non-fixed value). To select the best images, first a threshold for the *clearance* is selected, each image that has a lower clearance than the threshold is discarded, then among the remaining images, only the n with the highest clearance are kept; the default threshold value is 85%. Since, as we saw while describing the creation of the new dataset, the LR images must have a *clearance* $> 70\%$, it may happen that one or more imagesets are discarded if they do not reach the minimum threshold of n LR images with clearance greater than 85%. Due to the manual selection of appropriate ROIs with low cloud presence, described in section 4.1.1, no imagesets were discarded during this pre-processing step for not meeting the previous condition (considering $n=9$).
- **Saving the dataset**: finally, after performing all the previous steps, also for performance reasons, the pre-processed data are saved as *.npy* files.

Finally, we add that, even before performing the dataset loading step, all the data in the dataset were normalized using the *z-score* normalization:

$$z = \frac{x - \mu}{\sigma}$$

Where z is the new normalized value, μ is the the mean value and σ is the standard deviation, both calculated on the whole training set.

5.3 Trainings and experiments

At the beginning of the thesis, the agreed upon goal was to first test how the standard PIUnet architecture worked after being trained with the new dataset created using both Proba-V and Sentinel-2 data, and if necessary, in case of uninteresting, incorrect or improvable results, continue testing by applying changes to the network structure. It is important before going on to recall the fact that PIUnet works on two spectral bands, the NIR band and the RED band as we discussed earlier, consequently each training and experiment must be done and evaluated separately on each of the two bands. From here on, when we talk about, for example, *'PIUnet training'*, if the spectral band is not specified, this means that (at least) two trainings or tests have been carried out, one per band.

A series of trainings and experiments were performed during this work; their detailed description follows in the subsequent paragraphs.

5.3.1 Experiment using Pre-trained PIUnet

An experiment was carried out using the images from the test set of the new dataset created for this work as test images, but using as a model the pre-trained PIUnet model, originally trained using the original ESA's Kelvin competition dataset [3]. The pre-trained PIUnet model, in both the NIR [4] and RED [5] bands, that was used to carry out this experiment is the one (i.e., has the same configuration) that led to the results shown by D. Valsesia and E.Magli in [14].

This experiment, while not particularly useful to accomplish the main goal of this thesis, allowed us to answer two interesting questions.

[3] https://kelvins.esa.int/proba-v-super-resolution/data/

[4] NIR band Pre-Trained PIUnet

[5] RED band Pre-Trained PIUnet

How is the super-resolution task performed by PIUnet if the training and the test set used come from two different datasets? As said before, the two datasets in question are the one from the original ESA's challenge (as well as used in [14]) and the new one whose creation was described in Chapter 4 of this thesis. Both datasets have low-resolution images extracted from Proba-V, but in the pre-trained model of PIUnet, training was performed using the high-resolution images of Proba-V as ground truth, instead of Sentinel-2 imagery; the fact that tests were done on an already extensively tested and studied model might make the question uninteresting, but this is not true since the the two datasets used to test the model are different and were created by selecting different Regions Of Interest (ROIs) consequently it might have been expected that in one of the two datasets there would be images extracted from areas with radiometric or spatial features poorly present or absent in the other dataset, thus leading to the generation of (some) incorrectly created SR images or at least not totally accurate and reliable.

How much does a super-resolved image from Proba-V resemble a high-resolution image from Sentinel-2? Since the pre-trained PIUnet generates super-resolved images with the characteristics of Proba-V, using the new dataset as a test set, it was possible to quantitatively evaluate, by calculation of cPSNR, how closely the images in SR generated from Proba-V data resembled those from Sentinel-2 imagery. Logically, one cannot expect a very good result from cPSNR calculation, nevertheless it is still interesting to make an assessment of it because although the optical sensors of the two satellites have different characteristics, it is still true that Proba-V data are sufficiently consistent with those of Sentinel-2 in terms of overpass time, radiometry and spectral coverage.

5.3.2 Experiment using Standard PIUnet

This experiment was carried out training the standard PIUnet architecture on the new dataset created during this work and aimed to answer the fundamental question underlying this thesis: *is it possible to effectively perform super-resolution tasks using data from two different satellites in combination?* And more specifically, *is PIUnet capable of doing so?*

In this experiment, the first two PIUnet trainings were performed on the new dataset, one on the NIR spectral band images and one on the RED spectral band images. We remark here the fact that, as described extensively in Chapter 3 when discussing about PIUnet, the network takes as input a variable number of LR images; although each imageset of the created dataset has at least 16 LR images available, as described in section 4.1.3, in order to keep the configuration as close as possible to the original one, the trainings in this experiment were still carried out by selecting 9 LR images from the many available in each imageset (i.e., the

number of the temporal dimension is 9; nine different LR acquisitions of the same area are used to generate the correspondent SR image).

The Table 5.2 summarizes the configuration of Standard PIUnet during the execution of the present experiment.

no. RED train scenes	396
no. RED validation scenes	140
no. RED test scenes	100
no. NIR train scenes	396
no. NIR validation scenes	140
no. NIR test scenes	100
learning rate	e^{-4}
batch size	24
no. epoch	400
no. temporal dimension (i.e., different acquisitions over time)	9

Table 5.2: Experimental setup used for the trainings of Standard PIUnet

The neural network, as seen in Table 5.2, was trained for 400 epochs, a number that experimentally proved sufficient to reach the learning plateau. To complete the 400 epochs needed to finish the training phase, for each band, the model needed approximately 80 hours; this means that to train the model to work in both the NIR and RED bands the time needed to complete the training is one week, more or less.

Each training was performed on a dedicated server provided by Politecnico di Torino, using an Nvidia Quadro P6000, and the parameters presented above required approximately 19GB of GPU memory for training.

The figure 5.2 shows the trend of the loss function during the training of the standard version of PIUnet.

5.3.3 Experiments using Modified PIUnet

As will be explained later in Chapter 6, the results obtained from experiments on the standard version of PIUnet were interesting in many aspects, but certainly improvable. That is why multiple experiments were also carried out using the modified versions of PIUnet; all the modified versions of PIUnet shown in Table 5.1 were used for at least one experiment, consequently at least one training was performed for each of them.

The same parameters shown in Table 5.2 were chosen to perform the training and experiments of the various modified versions of PIUnet. The same considerations as

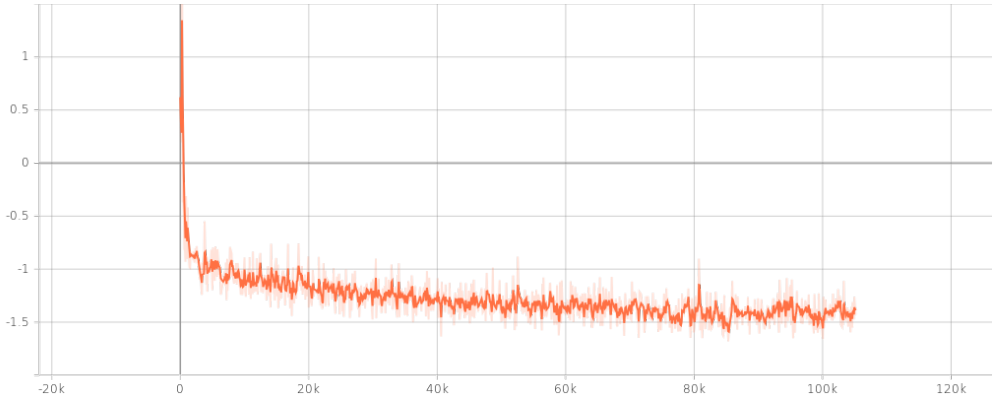


Figure 5.2: Trend of the loss function during the training of the standard PIUnet architecture in the NIR band.

before about the performance and execution time of a training also apply to these experiments: it took approximately 80 hours to complete a 400 epochs training. Since performing two trainings, one for each spectral band, took about a week of time, the following criterion was followed to perform the experiments as efficiently as possible: for each modified version of PIUnet a training was performed on the RED spectral band, subsequently, if the training led to quantifiable improvements with respect to the standard version of PIUnet then a training on the NIR band was also performed, conversely if the training worsened the performance or led to worse results than those obtained by the standard version of PIUnet then the second training in the NIR band was avoided. Finally, after analyzing in detail the cPSNR, the SR images and their histograms, performance (loss and cPSNR) during training, it turned out that among the various modified version of PIUnet tested, just the *version 2* and *version 3* led to interesting results.

Among the many experiments, these performed on the modified versions of PIUnet are, unquestionably, the most interesting as they allowed us to really understand the possibilities of working with SR models using images from two satellites in combination.

Figures 5.3 and 5.4, respectively, shows the trend of the loss function during the training of the *version 2* of modified PIUnet on the NIR and RED bands.

In Figure 5.5 is shown the trend of the loss function during the training of the *version 3* of modified PIUnet on the RED band.

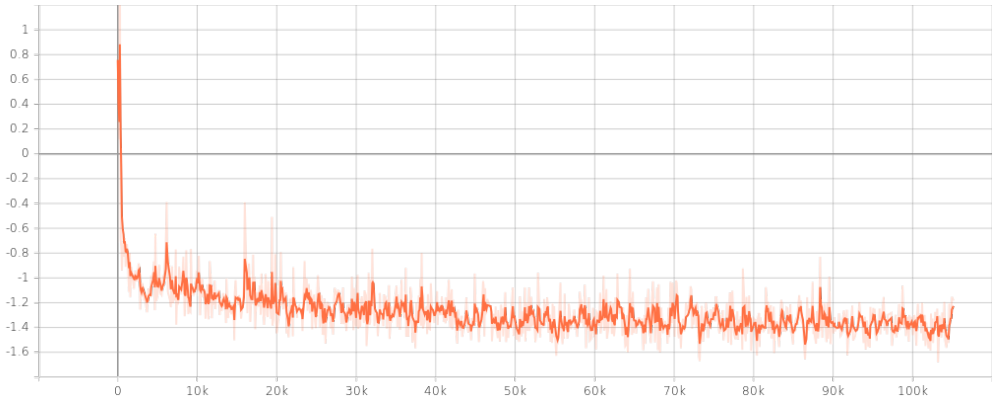


Figure 5.3: Trend of the loss function during the training of the modified PIUnet *version 2* in the NIR band.

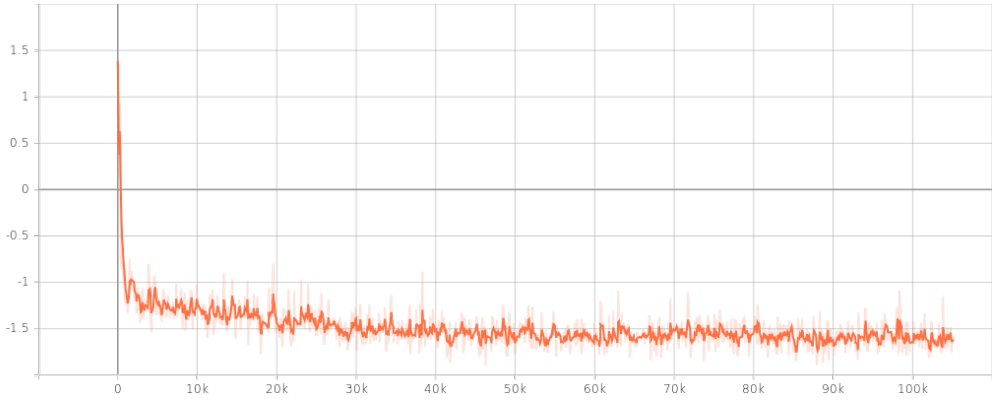


Figure 5.4: Trend of the loss function during the training of the modified PIUnet *version 2* in the RED band.

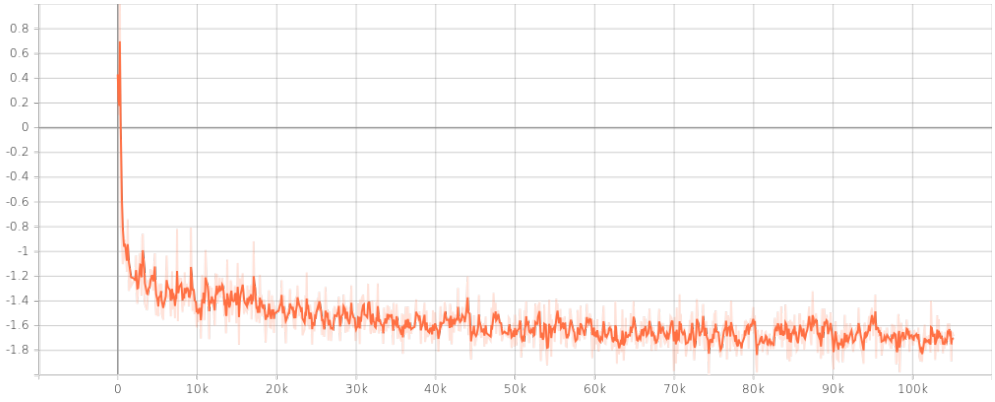


Figure 5.5: Trend of the loss function during the training of the modified PIUnet *version 3* in the RED band.

Chapter 6

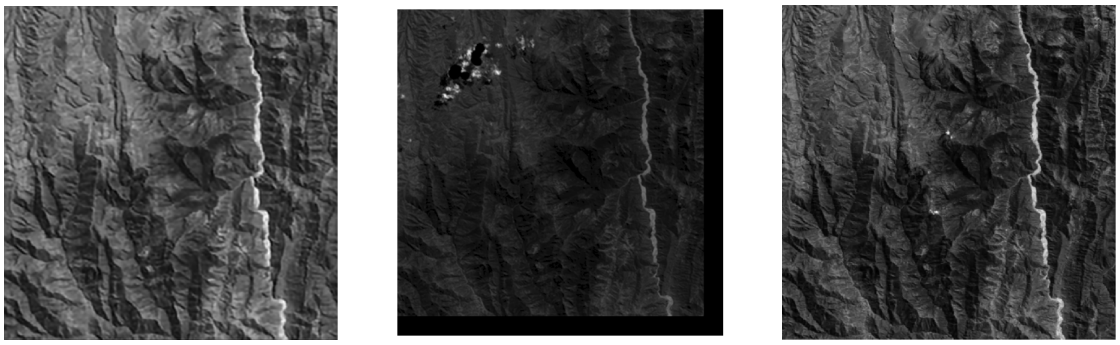
Results

In this chapter, the results of the previously mentioned experiments are presented and described. In addition, a comparison will be made between the results obtained from the two most interesting versions of PIUnet, namely *versions 2* and *version 3*, and the results of the standard architecture.

6.1 Results of Pre-trained PIUnet experiment

As discussed in the previous chapter, this experiment served to understand how pre-trained PIUnet worked with the Proba-V LR images contained in the new dataset, consequently the result of this experiment are super-resolved Proba-V images. We remark that since a pre-trained model was used, no training was carried out in this experiment, thus, the cPSNR could not be plotted during the training and validation phases; however, it was still possible to calculate cPSNR later using the test set.

In Figure 6.1 we can begin to observe, visually, the results obtained from this experiment. What is noticeable, which is what we expected, is that the SR image (Fig. 6.1a) obtained is extremely accurate with respect to the corresponding HR from Proba-V (Fig. 6.1c); this does not surprise us as the model that was used for this experiment is the one that was originally trained in [14] with a dataset that only contained data from Proba-V. Furthermore, what is observed visually in Figure 6.1 is also found quantitatively through the calculation of cPSNR: Table 6.1 shows the calculation of cPSNR evaluated using both Proba-V HR images and Sentinel-2 HR images as ground truth; by calculating cPSNR using Proba-V HR images as ground truth we can quantify how the super-resolved reconstruction of the model is similar to the original Proba-V image, similarly by calculating cPSNR using Sentinel-2 images as ground truth we understand how much an image with Proba-V features resembles one from Sentinel-2.



(a) SR Image **(b)** Sentinel-2 HR Image **(c)** Proba-V HR Image

Figure 6.1: A SR image *(a)* obtained by pre-trained PIUnet trained on the RED band. Image *(b)* shows the corresponding Sentinel-2 HR image while the Image (*c*) represents the corresponding Proba-V HR Image.

	Sentinel-2 HR	**Proba-V HR**
RED band	47.0338	69.4320
NIR band	45.7173	69.6610

Table 6.1: Mean cPSNR calculated over the test set using the results obtained by Pre-trained PIUnet. Columns headers show the data used as ground truth, while in the rows are indicated the spectral band over which the average cPSNR was calculated.

Looking at the average cPSNR values shown in Table 6.1 we can point out two aspects:

- The high values of the cPSNR calculated using Proba-V images as Ground Truth show us that even while changing dataset, the original model continues to perform correctly, performing the Proba-V Super-Resolution task with high accuracy.
- The large difference (> 20) in value between the cPSNR calculated using Proba-V data as ground truth and Sentinel-2 data as ground truth underscores the fact that although the two missions have similar characteristics in some respects (such as overpass time, radiometry, and spectral coverage) it is not possible to use the pre-trained PIUnet model to obtain a result that also accurately reconstructs SR images with characteristics from another satellite other than Proba-V

Also, in Figure 6.2 we can see how the super-resolution task is carried out using the pre-trained model and what are the outputs obtained; specifically, we can see

how the spatial resolution was increased by starting from multiple LR images (Fig. 6.2a shows one of the many LR images available), thus resulting in the two PIUnet outputs: the SR image (Fig. 6.2b) and the corresponding uncertainty map (Fig. 6.2c).

(a) LR Image **(b)** SR Image **(c)** Uncertainty Map

Figure 6.2: Input (*a*) and output obtained (*b,c*) from the pre-trained PIUnet model. The image was extracted from the RED data band at the coordinates *(latitude, longitude)* of *-20.034934, -64.876301*.

6.2 Results of Standard PIUnet experiment

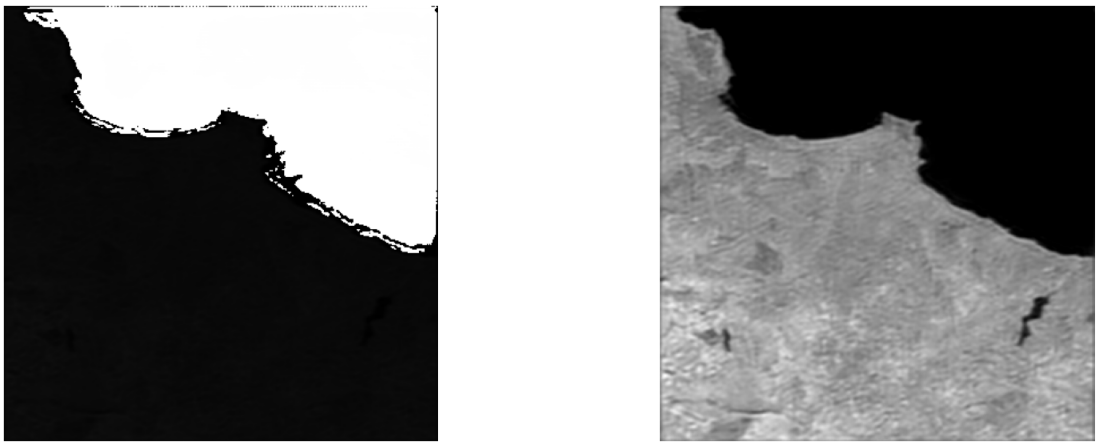
The results obtained from the experiments performed on the Standard PIUnet architecture, trained with the new dataset including images from both Proba-V (LR) and Sentinel-2 (HR), allowed us to answer a first fundamental question we asked: *is it possible to perform super-resolution using sentinel-2 and Proba-V data in combination?* To give us an answer, the results of this experiment are analyzed below.

In early experiments with the Standard version of PIUnet, two particular phenomena were detected in part of the images:

- In performing super-resolution of images with a massive presence of high values of reflectance (an example of surface that has high values of reflectance is the water) it happened that at these elements (pixels) were recreated in the SR image with extremely high values, greater than 60000, thus, absolutely unrealistic from a physical point of view.

- In the SR images that had the former phenomenon there was also often a blurring phenomenon, so as if the image had a lot of noise.

From a purely visual point of view, the first phenomenon represents an undesirable result as it does not allow a proper visualization of the SR image, in fact the very large difference between the maximum value (>60000, as mentioned) and the minimum value (a few thousand or hundreds usually) of TOA reflectance does not allow a clear visualization of the various shades of reflectance values present in the SR image. From an analysis point of view, on the other hand, surfaces with high reflectance such as water are of no particular interest for the purpose of analysis since increasing the spatial resolution serves mainly to perform better analyses on terrestrial surfaces such as vegetation, cities and so on; therefore, the solution was to mask all pixels in the SR image with a value greater than a certain threshold (chosen equal to 60000) by setting them to 0. The result of this post-processing process is shown in Figure 6.3 where the exact same SR image, before (Fig. 6.3a) and after (Fig. 6.3b) pixel masking, are compared; we can see that this pixel masking process done in post-processing actually made the SR image visible, at the same time, however, we notice that the blurring phenomenon is still noticeable even in the post-processed image.

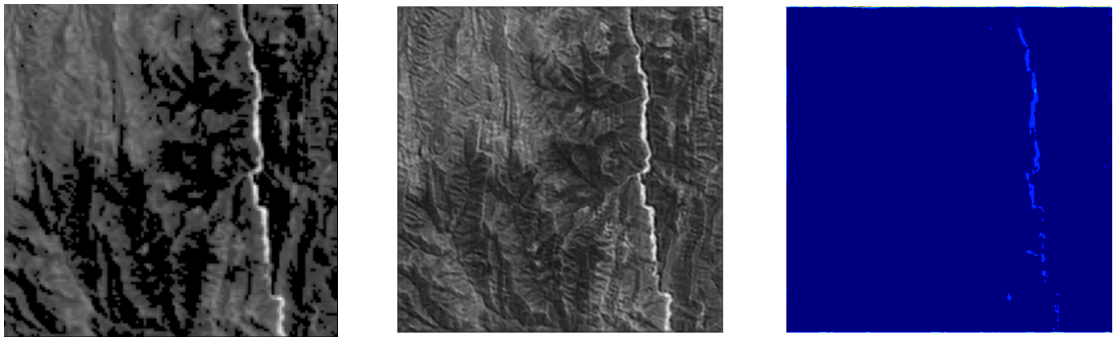


(a) SR Image before post-processing

(b) SR image after post-processing

Figure 6.3: Comparison between the SR Image before (a) and after(b) the masking operation applied on post-processing.

The two phenomena mentioned before, we stress again, were not found in all SR images but only in a part of them, all others, however, have no particular problems with blurring or unrealistic reflectance values. Figure 6.4, which shows an LR image and the outputs obtained from the model after being trained on the new dataset, visually demonstrates the considerations that have just been made: indeed, it can be seen that the SR image (Figure 6.4b) is displayed correctly and that the spatial resolution, compared to the LR image (Figure 6.4a) is increased considerably.



(a) LR Image **(b)** SR Image **(c)** Uncertainty Map

Figure 6.4: Input (a) and outputs obtained (b,c) after the training of the Standard PIUnet architecture on the new dataset containing data of both Proba-V and Sentinel-2 missions.

Again to get a quantitative idea based on a defined metric, of how accurate the obtained SR image is, we need to calculate the cPSNR. Table 6.2 shows the cPSNR values calculated on both NIR and RED bands and using both Sentinel-2 and Proba-V images as ground truth. Analyzing the data present in the latter we find that the cPSNR has decreased significantly especially in the calculation performed comparing the HR images of Proba-V, however, compared to the pre-trained version of the model, the cPSNR value calculated using the Sentinel-2 images as ground truth has increased by 3 per band, which means that after training the network on the new dataset and using Sentinel-2 images as ground truth the model became more proficient in creating Sentinel-2-like images rather than Proba-V-like images, while still using LR images of the latter satellite.

	Sentinel-2 HR	**Proba-V HR**
RED band	50.5796	47.6771
NIR band	48.1811	47.0348

Table 6.2: Mean cPSNR calculated over the test set using the results obtained after training the Standard PIUnet architecture. Columns headers show the data used as ground truth, while in the rows are indicated the spectral band over which the average cPSNR was calculated.

What is reported in Table 6.2 can be further verified in Figure 6.5 where the cPSNR was plotted during the model validation phase in the RED band. Indeed, it can be seen that at the end of the training, in the validation set, the calculated cPSNR is around the value of 50.4, which is in line with what can be seen in the

first cell of Table 6.2 (which, however, we recall represents the cPSNR calculated using the test set images).

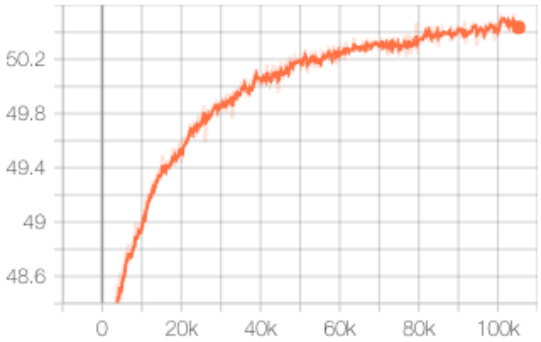
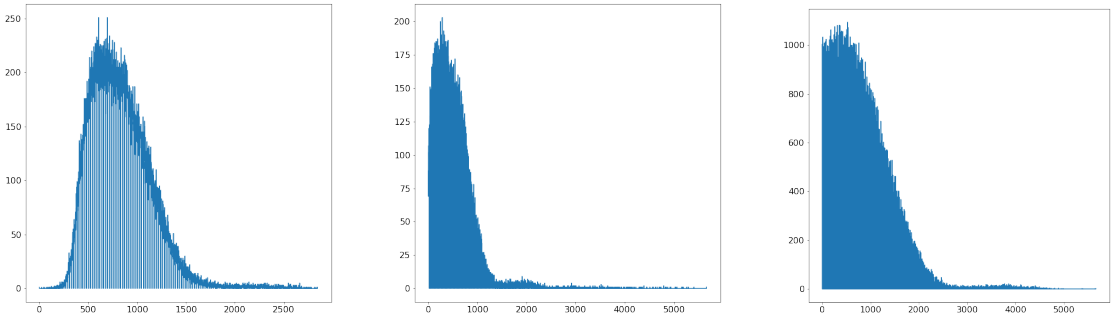


Figure 6.5: Trend of the cPSNR calculated on the imagesets of the validation set, during the training of Standard PIUnet architecture in the RED band. The peak value is 50.5 and was reached at step 101.500, about 3 days and 2 hours after the start of training.

We have visually analysed the images and calculated the cPSNR; at this point there is only one element missing to consider our analysis of the results complete: the histograms of the images, both the low-resolution and high-resolution images and the SR images, must be observed.

The *image histogram* is a graph used to plot the number of pixels (on the vertical axis) as a function of their intensity (on the horizontal axis), consequently it allows us to visualize the distribution of a continuous numeric variable (reflectance of a given image in our case); it is therefore important, for the purposes of our analysis, to compare several histograms with each other for the purpose of determining how much one SR image resembles another HR image. The histograms of some SR images trained on Standard PIUnet tend to better replicate the histograms of Sentinel-2 HR images than those of Proba-V images; the latter, however, is not a definite and valid pattern for all images: for many other images the pixel intensity values of the SR image, and the function plotted in the histogram, appear to lie somewhere between Proba-V and Sentinel-2 values, without resembling either histogram in particular but representing a middle ground between the two.

Figure 6.6 shows the histograms of the three images already shown in Figure 6.1. From the above figure we note that the histogram of the SR image (Fig. 6.6a) has an non-homogeneous distribution of values, this can be seen by the presence of white areas within the *bell* formed by the perimeter of the histogram (colored in blue); this particular pattern indicates the total absence of certain pixel intensity values in the super-resolved image in favor of a higher frequency of other reflectance values. This reason, combined with the fact that the radiometry of the SR image



(a) SR Image histogram

(b) Sentinel-2 HR Image histogram

(c) Proba-V HR Image histogram

Figure 6.6: From left to right: histogram of the SR image (a), histogram of the Sentinel-2 HR image (b) and histogram of the Proba-V HR image (c).

is not totally consistent with that of either Proba-V or Sentinel-2 imagery, led to the idea of developing a modified version of PIUnet.

6.3 Results of Modified PIUnet experiments

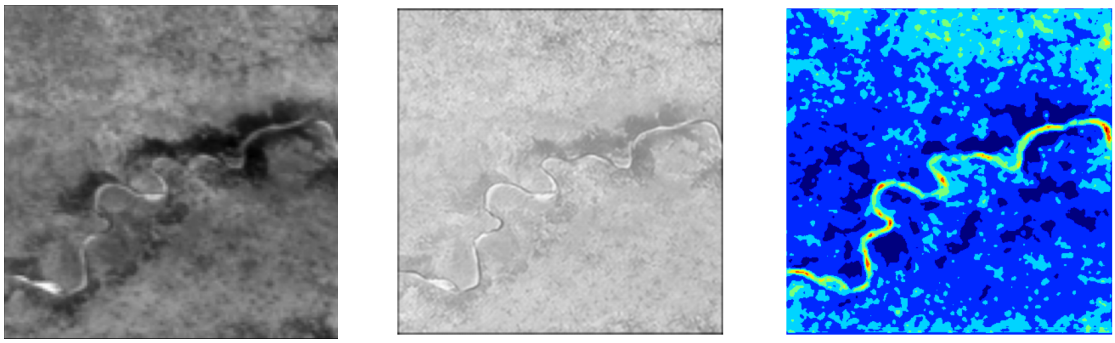
As mentioned earlier in Chapter 5, the modified versions of PIUnet from which we obtained the most interesting results were the *version 2* and the *version 3*. Therefore, in the following sections are described only the results obtained from the latter.

6.3.1 Modified PIUnet version 2

Let us begin the analysis of the results of modified PIUnet *version 2* by saying that the problems of image blurring and extremely high pixel values, both mentioned earlier, that occurred in the experiments performed on standard PIUnet did not occur by training and testing this version, consequently no pixel masking procedures were necessary in post-processing.

To get a visual idea of how the super-resolution task was performed by this version of the modified architecture, a low-resolution image and the corresponding SR image and uncertainty map are shown in Figure 6.7; from the latter we can observe that, from a visual point of view, the super-resolution task is solved correctly with an effective increase in spatial resolution and a result (Fig. 6.7b) that is pleasing to the eye, also, notice how the uncertainty map (Fig. 6.7c) realistically traces the features in the image indicating greater uncertainty values in all those areas where there are particular spatial features: notice how the river running

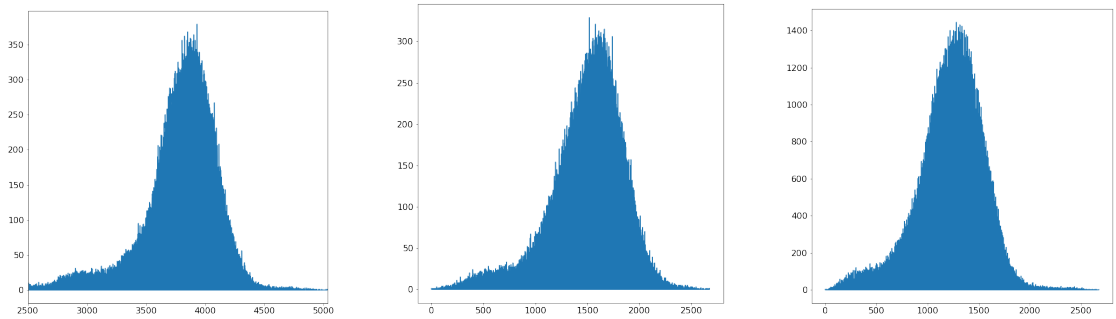
through the Figure 6.7 is colored in warmer colors in the UM, indicating less confidence in the reconstruction, which is a perfectly legitimate and realistic result considering the difficulty in increasing the resolution of spatially and radiometrically heterogeneous areas.



(a) LR Image **(b)** SR Image **(c)** Uncertainty Map

Figure 6.7: Input (*a*) and outputs obtained (*b*,*c*) after the training of the Modified PIUnet architecture *version 2* on the new dataset containing data of both Proba-V and Sentinel-2 missions.

A second consideration that can be made both by measuring the mean pixel value of the SR image (Fig. 6.7b) and by tracking the histograms (Fig. 6.8) of the images is that this modified version of PIUnet produces, on average, images that have a greater reflectance values.



(a) SR Image histogram **(b)** Sentinel-2 HR Image histogram **(c)** Proba-V HR Image histogram

Figure 6.8: From left to right: histogram of the SR image (*a*), histogram of the Sentinel-2 HR image (*b*) and histogram of the Proba-V HR image (*c*).

Notice looking at Figure 6.8 how the histogram of the SR image (Fig. 6.8a) has a homogeneous pixels distribution, like that of Sentinel-2 (Fig. 6.8b) and

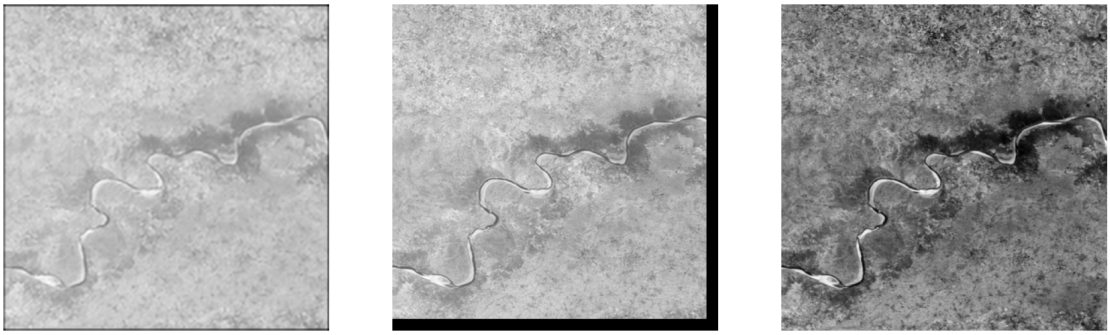
Proba-V (Fig. 6.8c). Note also, looking at the horizontal axis, where the pixel values of the images are shown (for the sake of better visualization the histogram in Fig. 6.8a shows the values on the horizontal axis starting from 2500), how the SR image has much higher intensity values than the other two HR images. Finally, the most interesting thing to note is that the histogram of the SR image almost perfectly follows that of Sentinel-2, from which we infer that the changes made in the modified PIUnet version 2 were effective: the image obtained from PIUnet is radiometrically consistent with Sentinel-2 imagery.

	Sentinel-2 HR	**Proba-V HR**
RED band	52.2491	48.4827
NIR band	50.5344	48.2647

Table 6.3: Mean cPSNR calculated over the test set using the results obtained after training the modified PIUnet architecture *version 2*. Columns headers show the data used as ground truth, while in the rows are indicated the spectral band over which the average cPSNR was calculated.

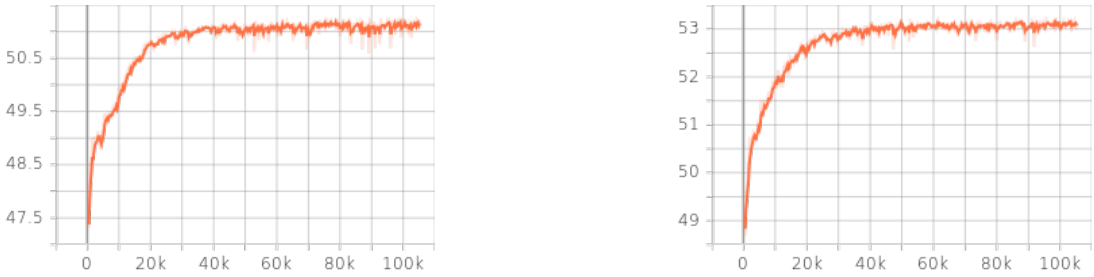
Table 6.3 shows the result of the cPSNR calculation, as usual calculated using both Sentinel-2 and Proba-V HR images as Ground Truth. Note how, compared with the data shown in Table 6.2, the cPSNR, especially calculated on Sentinel-2 data, has significantly increased; this also allows us to observe from a quantitative point of view how the modification implemented on the original PIUnet architecture has actually brought improvements in performing the super-resolution on Sentinel-2 imagery. A further very interesting point to make is that, as seen in the previous chapter, the consistency loss was calculated using the downsampled SR image as input data and as ground truth the LR images of Proba-V (which were averaged on the temporal axis), we would therefore have expected a more pronounced increase in cPSNR with respect to the Proba-V images, instead what we see from Table 6.3 is that the addition of this loss had more effect in improving the similarity to HR images (thus Sentinel-2) rather than LR images (thus Proba-V). It is also interesting to note the trend of cPSNR during trainings shown in Figure 6.10.

Finally, let us compare, in Figure 6.9 the SR image obtained from modified PIUnet *version 2* with the corresponding high-resolution images from Proba-V and Sentinel-2. This figure, especially, allows us to note the high pixel value of the SR image compared to both the Sentinel-2 and Proba-V images, in fact this is visually reflected in an image that looks significantly brighter than the other two. Clearly, we remark that the cPSNR is not adversely affected by this phenomenon since, as explained in section 3.5.4, it is insensitive to the absolute brightness of images.



(a) SR Image **(b)** Sentinel-2 HR Image **(c)** Proba-V HR Image

Figure 6.9: A SR image *(a)* obtained by modified PIUnet *version 2* trained on the RED band. Image *(b)* shows the corresponding Sentinel-2 HR image while the Image (*c*) represents the corresponding Proba-V HR Image.



(a) cPSNR of the validation set (NIR band)

(b) cPSNR of the validation set (RED band)

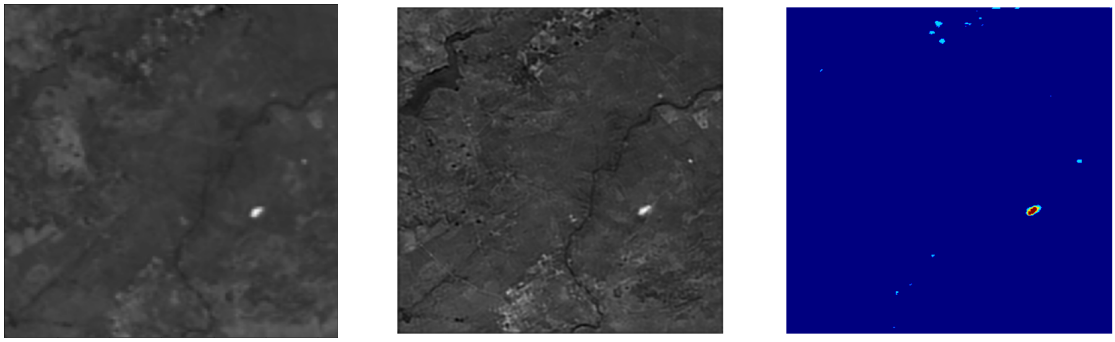
Figure 6.10: Trends of the cPSNR calculated on the imagesets of the validation set during the training of modified PIUnet *version 2* on both RED and NIR bands.

6.3.2 Modified PIUnet version 3

Also in the results of modified PIUnet *version 3*, as in the *version 2*, the problem of high pixel values did not occur, although some SR images turned out to be blurry as in the case of Standard PIUnet.

Figure 6.11 shows a low-resolution Proba-V image and the corresponding SR image and uncertainty map produced by *version 3*: analysing the figure we can observe that the super-resolution is correctly performed, with an effective improvement in spatial resolution of the SR image (Fig. 6.11b) compared with the LR image (Fig. 6.11a). Regarding the uncertainty map (Fig. 6.11c), note that the model has high confidence (i.e., low aleatoric uncertainty values) in all areas of the image; this pattern is present in, more or less, all uncertainty maps, but

this is not a particularly realistic and reliable result since it is expected that in areas of the image with particular spatial features there will be lower confidence in reconstructing the SR image, as was the case in *version 2*.



(a) LR Image **(b)** SR Image **(c)** Uncertainty Map

Figure 6.11: Input (a) and outputs obtained (b,c) after the training of the Modified PIUnet architecture *version 3* on the new dataset containing data of both Proba-V and Sentinel-2 missions.

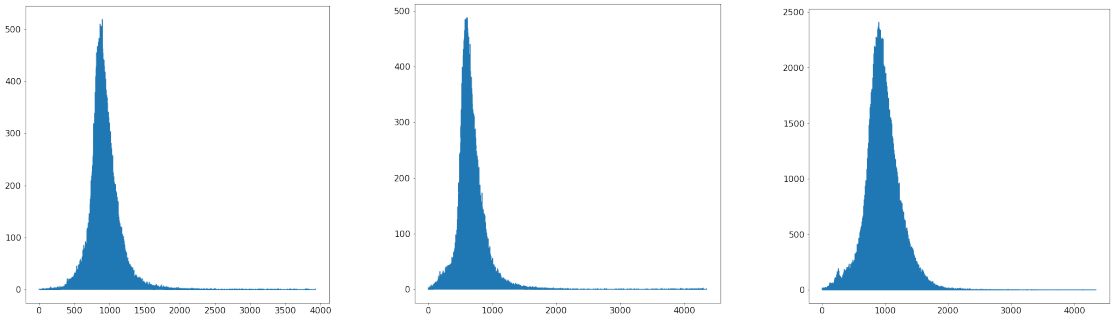
As for the cPSNR values, they are shown in Table 6.4: we note that compared with the standard version of PIUnet there was a general improvement in cPSNR calculated using Sentinel-2 images as ground truth, while for Proba-V images the improvement is minimal. Also in this modified version of PIUnet, as in *version 2*, by analyzing the cPSNR values, we notice that the model is more capable of recreating Sentinel-2-like images than Proba-V.

	Sentinel-2 HR	**Proba-V HR**
RED band	51.5208	47.7112
NIR band	49.4665	47.2381

Table 6.4: Mean cPSNR calculated over the test set using the results obtained after training the modified PIUnet architecture *version 3*. Columns headers show the data used as ground truth, while in the rows are indicated the spectral band over which the average cPSNR was calculated.

The most interesting results to analyze from this version, however, are those obtained by plotting the histograms of the images. As can be seen in Figure 6.12, the SR image created by this version recreates pretty well the histograms of Sentinel-2: this means that the SR image (Fig. 6.12a) is radiometrically consistent with the ground truth of Sentinel-2 (Fig. 6.12b), unfortunately, this is not the case for all images; in fact, some SR images have histograms that follow the shape

of that of Sentinel-2, but with an uneven distribution of pixels. But the most interesting consideration is that, unlike what we saw in the analysis of *version 2*, we notice how in this version of PIUnet the intensity of the pixel values of the SR image is perfectly in line with the values of the HR images of Proba-V and Sentinel-2, particularly of the latter. This last observation made, in addition to the histograms, also has visual confirmation by comparing three images: the SR image, the Proba-V HR image and the Sentinel-2 HR image; this comparison is presented in Figure 6.13.

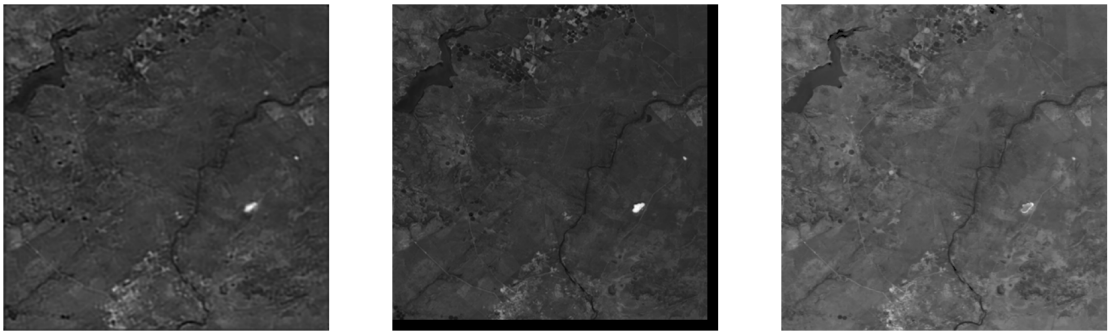


(a) SR Image histogram

(b) Sentinel-2 HR Image histogram

(c) Proba-V HR Image histogram

Figure 6.12: From left to right: histogram of the SR image (a), histogram of the Sentinel-2 HR image (b) and histogram of the Proba-V HR image (c).



(a) SR Image

(b) Sentinel-2 HR Image

(c) Proba-V HR Image

Figure 6.13: A SR image *(a)* obtained by modified PIUnet *version 3* trained on the RED band. Image *(b)* shows the corresponding Sentinel-2 HR image while the Image (c) represents the corresponding Proba-V HR Image.

6.4 Comparison of the results

This section aims to present a detailed comparison of the results obtained from the three main experiments: the one performed on the standard version of PIUnet and the other on the *version 2* and *3* of modified PIUnet. To make the analysis clearer and easier to understand, it will be divided into several points: super-resolved image, uncertainty map, cPSNR and histogram.

It is important to specify, before moving on to the comparison, that the considerations that are made below are argued by showing only one scene, significant for understanding, at a time; however, it is clear that these considerations are general and applicable to all, or almost all, of the imagsets obtained from the various versions.

6.4.1 Super-Resolved Image

A comparison of the three super-resolved images obtained from Standard PIUnet, Modified PIUnet *version 2* and Modified PIUnet *version 3*, respectively, is shown in Figure 6.14; from this figure, it can be seen that all three models perform the super-resolution task correctly, decreasing the spatial resolution from 300 to 100 meters. A slight blurring effect can also be seen in the image obtained from Standard PIUnet; this effect is totally absent from *version 2* while in *version 3* results you can find slightly blurred images. Finally, we note that the image obtained from *version 2* has significantly higher intensity values than the other two models, which can be seen from the high brightness of the image.

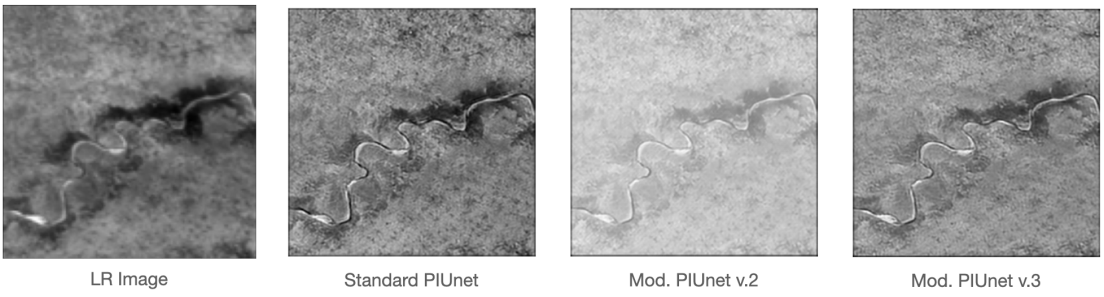


Figure 6.14: From left to right: LR image, SR image obtained from Standard PIUnet, SR image obtained from Modified PIUnet *version 2*, and SR image obtained from Modified PIUnet *version 3*.

We also remark that the image obtained from Standard PIUnet is correctly visualized only after the pixel masking operation performed in post-processing: this operation is not applied to the other two models since neither of them has problems with extremely high values in the images.

6.4.2 Uncertainty Map

Regarding uncertainty maps, the only model that always generates the most reliable and accurate maps is Modified PIUnet *version 2*, the other two models, however, tend to generate uncertainty maps with most of the pixels indicating a very low aleatoric uncertainty value, which is quite improbable especially in those images that have peculiar and heterogeneous spatial and radiometric features.

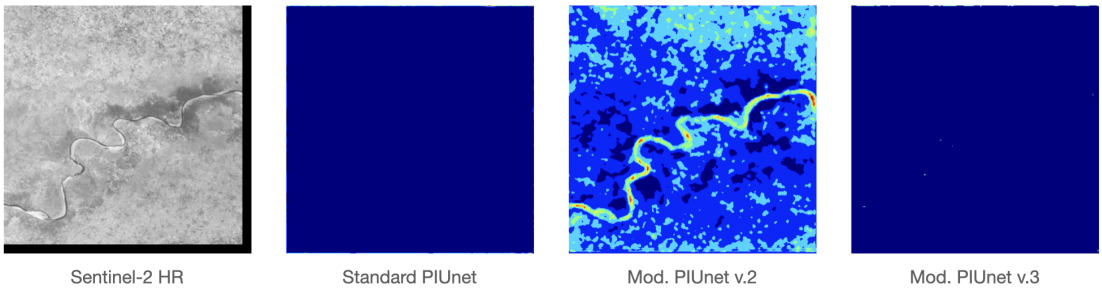


Figure 6.15: From left to right: Sentinel-2 HR image, UM obtained from Standard PIUnet, UM obtained from from Modified PIUnet *version 2*, and UM obtained from Modified PIUnet *version 3*.

Figure 6.15 shows an example of an image for which only Modified PIUnet *version 2* generated a realistic uncertainty map, while Figure 6.16 shows an image where the other two models also succeed in creating reliable, pseudo-realisitic uncertainty maps.

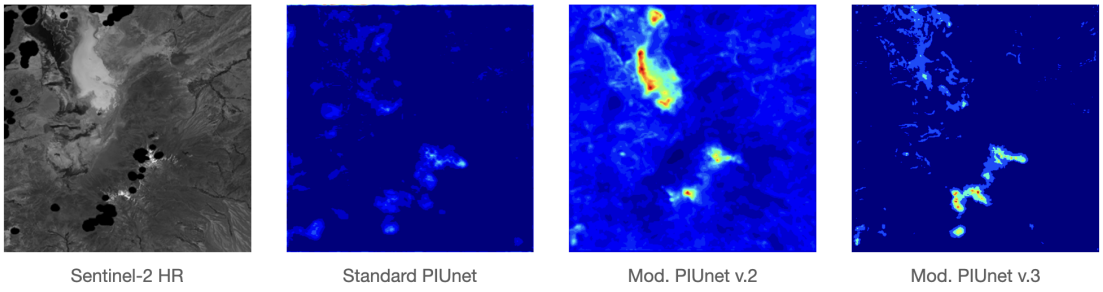


Figure 6.16: From left to right: Sentinel-2 HR image, UM obtained from Standard PIUnet, UM obtained from from Modified PIUnet *version 2*, and UM obtained from Modified PIUnet *version 3*.

6.4.3 cPSNR

The table in Figure 6.17 shows all cPSNR values calculated on the results obtained from the three different models.

	Standard PIUnet		Modified PIUnet v.2		Modified PIUnet v.3	
	Sentinel-2 HR	Proba-V HR	Sentinel-2 HR	Proba-V HR	Sentinel-2 HR	Proba-V HR
RED band	50.5796	47.6771	52.2491	48.4827	51.5208	47.7112
NIR band	48.1811	47.0348	50.5344	48.2647	49.4665	47.2381

Figure 6.17: Comparison of cPSNR values calculated on Standard PIUnet, Modified PIUnet *version 2* and Modified PIUnet *version 3*

From the above table we can make the following observations:

- The introduction of the new dataset, i.e., changing the ground truth during training from Proba-V images to Sentinel-2 images, has made the model more capable of creating Sentinel-2-like SR images; just observe how the cPSNR is always higher when calculated using Sentinel-2 as the ground truth. This allows us to say that it is indeed possible to perform super-resolution tasks using data from two different missions in combination.
- Modified PIUnet *version 2* is the model that gave us the highest improvement in terms of cPSNR, with a very good improvement of about 2 points for both the RED and NIR band (with Sentinel-2 data as ground truth); however, Modified PIUnet *version* 3 also showed a noticeable, albeit minor, improvement in cPSNR values. We can therefore say that the best improvement was obtained from *version 2*, while *version 3* still represents an improvement, but somewhere in between Standard PIUnet and Modified PIUnet *version 2*.

6.4.4 Histogram

As far as the histograms are concerned, what can be observed is that:

- The histograms obtained from Standard PIUnet are not radiometrically consistent with either Sentinel-2 or Proba-V: this is demonstrated by the fact that according to the image being analysed, the histogram of the super-resolved

image sometimes resembles that of the Sentinel-2 HR image, sometimes that of the Proba-V HR image, and sometimes represents a middle ground between the two previous cases, which is the worst case. In terms of pixel intensity, however, the SR images of Standard PIUnet have values in line with those of the two satellites.

- The histograms obtained from Modifed PIUnet *version 2* are radiometrically consistent with the Sentinel-2 HR images: they are almost equal to the Sentinel-2 histograms and have a homogeneous pixel distribution. The problem with this version is the high intensity of the pixel values in the SR image.
- The histograms obtained by Modifed PIUnet *version 3* are often, but not always, radiometrically consistent with the HR images of Sentinel-2; in some images, histograms are not particularly similar to those of Sentinel-2 or have a different distribution of pixel values than those of Sentinel-2 and Proba-V. The greatest strength of this version is the pixel intensity value, which follows Sentinel-2's intensity values almost perfectly.

Figure 6.18 shows the histograms (of imageset 12 of the RED band) of each of the previously mentioned architectures, compared with those of the high-resolution images of Proba-V and Sentinel-2; from this figure, the previous considerations can be verified.

6.4.5 Conclusions

Finally, we can sum up our analysis with the following considerations:

- All models tested perform the super-resolution task visually correctly, with a real improvement in the spatial resolution of the images. Standard PIUnet, however, suffers from a problem of blurring and extremely high values for pixels representing water.
- *Version 2* is the one that yields the highest cPSNR values, which means that the resulting image is the one with the best quality compared to the original high-resolution image; furthermore, the histograms of this version follow those of the Sentinel-2 images particularly reliably and the uncertainty maps are reliable and accurate. The negative aspect of this version is the intensity of the pixel values, which on average is higher than that of the original images.
- *Version 3* represents a middle ground: it has a lower cPSNR than *version 2*, but higher than the Standard PIUnet version. The images are often recreated in a radiometrically accurate manner, and this can be verified by comparing the histograms of this version with those of the original images; despite this,

some specific images have a non-uniform pixel distribution, moreover, the uncertainty maps generated by this version are not always reliable. The positive aspect of this version is the intensity of the pixel, that is reliable with Sentinel-2 and Proba-V data.

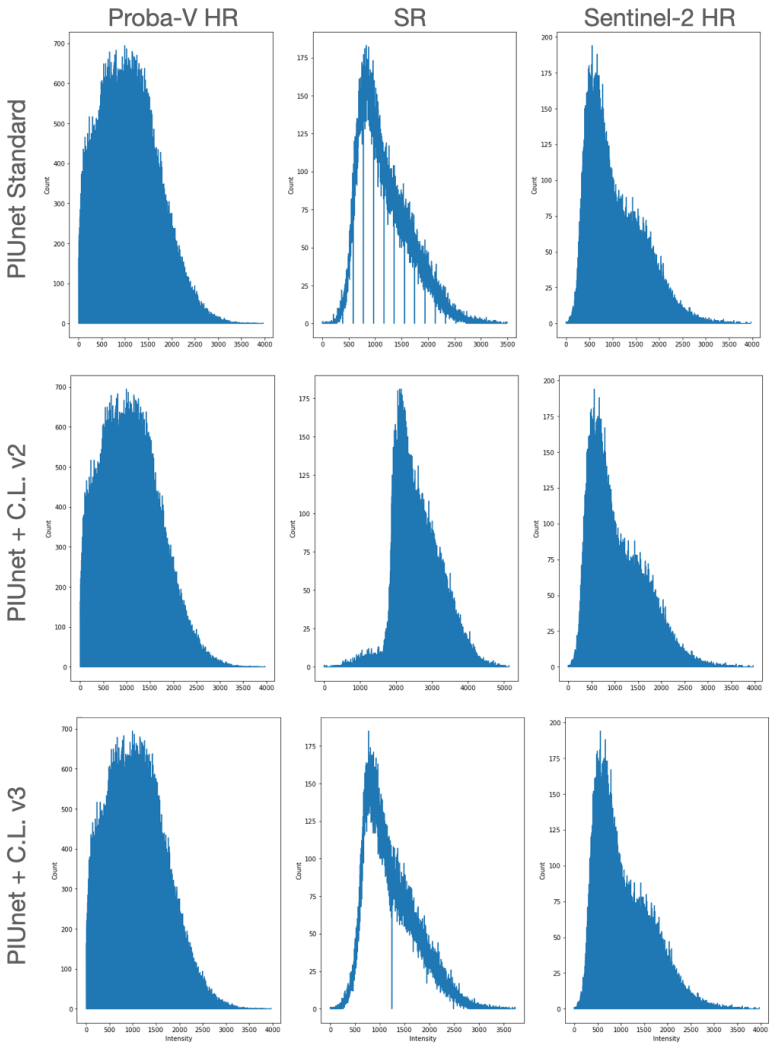


Figure 6.18: Comparison of histograms of Standard PIUnet, Modified PIUnet *version 2* and Modified PIUnet *version 3*.

In conclusion, the most visually and radiometrically interesting result is represented by *version 2*, which has the only flaw of having excessively high pixel intensity values; unfortunately, since these images are mainly used to perform analysis, having high pixel values means that proper analysis cannot be performed, which is why version 2 is (at present) probably the perfect middle ground.

Chapter 7

Conclusion

The goal of this thesis was initially to understand whether it was possible to combine images obtained from two satellites, Proba-V and Sentinel-2 in our case, to do super-resolution tasks using an existing model: PIUnet.

The results obtained, first from the training of the standard PIUnet architecture on the new dataset, allowed us to understand that it is possible to perform super-resolution tasks of Proba-V data in combination with Sentinel-2 data; but the most interesting results came with the modification of the architecture of PIUnet and the addition of a new loss function: the consistency loss.

As a result of the modifications, we obtained even more interesting results, achieving radiometrically consistent SR images with Sentinel-2 and increasing the value of cPSNR, the main quality evaluation metric used in super-resolution tasks. Although the results obtained are interesting, there is still room for improvement, for example, with the creation of a novel neural network or further improvement of PIUnet with the aim of obtaining a final version that allows the generation of super-resolved Sentinel-2-like images, with the same intensity (like in *version 3*), radiometrically consistent and with reliable uncertainty maps (like in *version 2*). Probably, at present, the best result to be obtained from this work would be to be able to decrease the pixel intensity of Modified PIUnet *version 2* (the only defect of this version),for example by applying processes in post-processing to decrease the intensity by a certain value calculated image by image, but keeping the excellent results obtained by the latter in terms of cPSNR, reliability of uncertainty maps and radiometric consistency.

Furthermore, building on this work, one can also try, in the future, to carry out super-resolution tasks by combining other satellites than Proba-V and Sentinel-2: a meaningful example would be the use of **cubesat** satellites. The latter are miniaturized satellites that are particularly interesting since they have very low spatial resolution, on the order of a few meters, and acquire data in the visible and near-infrared spectral bands. Although these satellites allow us to have very

high resolution both temporally and spatially, they lack in radiometric quality, so starting from an approach similar to the one presented in this paper we can think of combining cubesat data with data obtained from other satellites (with high radiometric quality) so as to obtain SR images with very low spatial resolution and at the same time high radiometric quality.

Appendix A

Dataset's coordinates

The following table contains the first 30 coordinates chosen for the scenes of the dataset used in this work and mentioned in Chapter 4.

Table A.1: List of coordinates.

List of coordinates			
scene no.	**Latitude**	**Longitude**	**Time period**
1	38.211305890764514	13.312500272478376	May-June 2020
2	15.232141585577104	49.410712650844026	May-June 2020
3	37.7797600882394	13.270832879202706	May-June 2020
4	37.37202744256883	13.928571530750819	May-June 2020
5	37.812498183477494	14.767856688726514	May-June 2020
6	24.122021266392302	52.49702217465355	May-June 2020
7	10.907734462193083	48.42261977422805	May-June 2020
8	24.520832970028835	52.497022356305806	May-June 2020
9	23.8124996366955	52.497022356305806	May-June 2020
10	23.3124996366955	52.497022356305806	May-June 2020
11	22.8124996366955	52.497022356305806	May-June 2020
12	24.122021266392302	52.99702217465355	May-June 2020
13	24.122021266392302	53.49702217465355	May-June 2020
14	24.122021266392302	53.99702217465355	May-June 2020
15	24.122021266392302	54.49702217465355	May-June 2020
16	24.520832970028835	52.997022356305806	May-June 2020
17	24.520832970028835	53.497022356305806	May-June 2020
18	24.520832970028835	54.497022356305806	May-June 2020
19	23.8124996366955	52.997022356305806	May-June 2020
20	23.8124996366955	53.497022356305806	May-June 2020
21	23.8124996366955	53.997022356305806	May-June 2020

Continuation of Table A.1			
scene no.	**Latitude**	**Longitude**	**Time period**
22	23.8124996366955	54.497022356305806	May-June 2020
23	23.3124996366955	52.997022356305806	May-June 2020
24	23.3124996366955	53.497022356305806	May-June 2020
25	23.3124996366955	53.997022356305806	May-June 2020
26	23.3124996366955	54.497022356305806	May-June 2020
27	22.8124996366955	52.997022356305806	May-June 2020
28	22.8124996366955	53.497022356305806	May-June 2020
29	22.8124996366955	53.997022356305806	May-June 2020
30	22.8124996366955	54.497022356305806	May-June 2020
End of Table			

Due to a matter of space it is not possible to include all 636 coordinates (in the *latitude, longitude* format) in the previous table, I summarize below the rest of the information related to the coordinates chosen for the dataset:

- 8 scenes selected in the zone ranging from 35.172615, 64.169644 to 35.172615, 65.169644 in April-May 2020.
- 48 scenes selected in the zone ranging from 35.172615, 65.669644 to 32.672615, 68.669644 in May-June 2020.
- 89 scenes selected in the zone ranging from 35.0, -113.0 to 32.0, -107.0 in April-May 2020.
- 48 scenes selected in the zone ranging from -18.035713, -66.875003 to -20.535713, -63.375003 in April-May 2020.
- 78 scenes selected in the zone ranging from -29.5, 138.0 to -33.0, 142.5 in February-March 2020.
- 74 scenes selected in the zone ranging from 31.0, 28.0 to 28.0, 33.0 in April-May 2020.
- 70 scenes selected in the zone ranging from -28.0, 24.0 to -31.0, 28.5 in May-June 2020.
- 99 scenes selected in the zone ranging from 28.0, 76.0 to 24.0, 81.0 in May-June 2020.
- 45 scenes selected in the zone ranging from -25.5, 116.0 to -28.0, 120.0 in May-June 2020.

- 47 scenes selected in the zone ranging from 41.0, -6.0 to 38.5, -2.5 in May-June 2020.

Bibliography

[1] Abdulsalam Ghalib Alkholidi and Khaleel Saeed Altowij. «Free Space Optical Communications — Theory and Practices». In: (2014). URL: https://www.semanticscholar.org/paper/Free-Space-Optical-Communications-%E2%80%94-Theory-and-Alkholidi-Altowij/c798c9f917c270f25c9e7e20e7a92edf3eeaeb90 (cit. on p. 6).

[2] Siegmund. «Eduspace». In: (2005). URL: https://www.esa.int/SPECIALS/Eduspace_Disasters_IT/SEMJR5460QH_0.html#subhead5 (cit. on p. 6).

[3] Gastellu-Etchegorry Jean-Philippe. «Physics of Remote Sensing». In: (Aug. 2016). URL: https://earth.esa.int/web/eo-summer-school/documents/973910/2642313/JG1to3.pdf (cit. on p. 7).

[4] Richard Müller. «Calibration and Verification of Remote Sensing Instruments and Observations». In: (June 2014) (cit. on p. 10).

[5] Erwin Wolters, Wouter Dierckx, Marian-Daniel Iordache, and Else Swinnen. «PROBA-V Products User Manual v3.01». In: (Mar. 2018) (cit. on pp. 13–17).

[6] Wouter Dierckx, Sindy Sterckx, Iskander Benhadj, Stefan Livens, Geert Duhoux, Tanja Van Achteren, Michael Francois, Karim Mellab, and Gilbert Saint. «PROBA-V mission for global vegetation monitoring: standard products and image quality». In: (Mar. 2014) (cit. on p. 14).

[7] M. Drusch et al. «Sentinel-2: ESA's Optical High-Resolution Mission for GMES Operational Services». In: (Feb. 2012) (cit. on p. 18).

[8] William A. Giovinazzo. «Overfitting/Underfitting – How Well Does Your Model Fit?» In: (May 2017). URL: https://meditationsonbianddatascience.com/2017/05/11/overfitting-underfitting-how-well-does-your-model-fit/ (cit. on p. 24).

[9] R. Mohammad, F. Thabtah, and L. McCluskey. «Tutorial and critical analysis of phishing websites methods». In: *Computer Science Review Journal* (2015) (cit. on p. 25).

[10] Aurélien Géron. «Hands-on Machine Learning with Scikit-Learn, Keras and TensorFlow». In: (2019) (cit. on p. 25).

[11] Sabina Pokhrel. «Beginners Guide to Convolutional Neural Networks». In: (2019). URL: https://towardsdatascience.com/beginners-guide-to-understanding-convolutional-neural-networks-ae9ed58bb17d (cit. on p. 26).

[12] Vishal Rajput. «Pooling layers in Neural nets and their variants». In: (Jan. 2022). URL: https://medium.com/aiguys/pooling-layers-in-neural-nets-and-their-variants-f6129fc4628b (cit. on p. 27).

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. «Deep Residual Learning for Image Recognition». In: (Dec. 2015) (cit. on pp. 27, 28).

[14] Diego Valsesia and Enrico Magli. «Permutation invariance and uncertainty in multitemporal image super-resolution». In: (May 2021) (cit. on pp. 29–32, 34, 35, 37, 51, 54–56, 60).

[15] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. «Attention Is All You Need». In: (2017) (cit. on p. 31).

[16] Y. Zhang, K. Li, K. Li, L. Wang, B. Zhong, and Y. Fu. «Image super-resolution using very deep residual channel attention networks». In: (2018) (cit. on p. 32).

[17] Andrea Bordone Molini, Diego Valsesia, Giulia Fracastoro, and Enrico Magli. «DeepSUM: Deep neural network for Super-resolution of Unregistered Multitemporal images». In: (Jan. 2020) (cit. on pp. 32, 34).

[18] F. Salvetti, V. Mazzia, A. Khaliq, and M. Chiaberge. «Multi-Image Super Resolution of Remotely Sensed Images Using Residual Attention Deep Neural Networks». In: (July 2020) (cit. on pp. 34, 54).

[19] Marcus Märten, Dario Izzo, Andrej Krzic, and Danie l Cox. «Super-Resolution of PROBA-V Images Using Convolutional Neural Networks». In: (July 2019) (cit. on pp. 38, 39).

[20] SUHET. «Sentinel-2 User Handbook». In: (July 2015) (cit. on p. 43).

[21] Muhammed T. Razzak, Gonzalo Mateo-Garcia, Luis Gómez-Chova, Yarin Gal, and Freddie Kalaitzis. «Multi-Spectral Multi-Image Super-Resolution of Sentinel-2 with Radiometric Consistency Losses and Its Effect on Building Delineation». In: (Nov. 2021) (cit. on pp. 50, 51, 53).