# End-to-End Framework for Continuous Space-Time Super-Resolution on Remote Sensing

Cristian Gutiérrez Gómez

**Resum–** Al camp del Remote Sensing, s'han dedicat molts esforços al camp de la Super-Resolution per superar les limitacions físiques dels sensors, el Deep Learning ha superat àmpliament els mètodes basats en Interpolació i Reconstrucció. Mètodes Espacials i Multi-Espectrals són predominants en el camp i, motivats per les històries d'èxit recents del modelatge espacial 3D amb Representació Neuronal implícita, estan apareixent nous mètodes de modelització continua aplicats sobre imatges. En el present treball, aprofitem les tècniques espacials i espectrals ja existents i la representació contínua d'imatges amb Local Implicit Image Function (LIIF) afegint la dimensió temporal al problema, resultant en un model d'interpolació contínua d'espai i temps com a una primer aproximació a la modelització total. Codi disponible a https://github.com/ggcr/Super-Temporal-LIIF.

**Paraules clau–** Remote Sensing, Space-Time Continuous Super-Resolution, Temporal Interpolation, Implicit Neural Representation, Local Implicit Image Function (LIIF).

**Abstract–** In Remote Sensing, much effort has been dedicated to the Super-Resolution field to overcome physical sensors limitations, and Deep Learning has vastly surpassed Interpolation and Reconstruction based methods. Spatial and Multi-Spectral based methods are commonly pre-dominant in the field, and, motivated by the recent success stories of 3D spatial modeling with Implicit Neural Representation, new continuous image modeling methods are appearing. In this present work, we take advantage of already existing Spatial and Spectral techniques and Learning Continuous Image Representation with Local Implicit Image Function (LIIF) by adding the Temporal dimension into the problem, leaning towards a continuous interpolation model of space and time as a first approximation to the total modelization. Code available at https://github.com/ggcr/Super-Temporal-LIIF.

**Keywords–** Remote Sensing, Space-Time Continuous Super-Resolution, Temporal Interpolation, Implicit Neural Representation, Local Implicit Image Function (LIIF).

## 1 Introduction

Remote Sensing collects data on the Earth's surface from a distance, usually using satellites, aircraft, or drones. The data collected can provide valuable information about the environment, such as climate change, vegetation health, and weather patterns.

The spatial, spectral, and temporal resolutions of the data refer to the level of detail (spatial resolution), wavelength range (spectral resolution), and frequency of data collection (temporal resolution), respectively. Different satellites may use different types of sensors and have different resolutions. Therefore, the data collected by one satellite may differ from that of another, even if they are monitoring the same area.

The resolution of satellite sensors has improved over the years, but even the most advanced sensors have limitations. For example, let us consider the distance that a pixel of the image represents over the ground, also known as Ground Sampling Distance (GSD). We see that some sensors can have a spatial resolution of 20 m/pixel of GDS (where 1

• E-mail de contacte: Cristian.GutierrezG@autonoma.cat
• Menció realitzada: Computació
• Treball tutoritzat per: Felipe Lumbreras Ruiz (Ciències de la Computació)
• Curs 2022/23

pixel represents 20 meters), while others can have a resolution of 40 m/pixel of GDS (where 1 pixel represents 30 meters). Moreover, in many cases, this level of resolution is insufficient for certain applications.

Multispectral satellites, such as the European Space Agency's (ESA) Sentinel S2, capture information from a wide range of areas, commonly called Areas Of Interest (AOIs), by measuring the electromagnetic radiation generated by the Earth by specific ranges of wavelengths and encapsulate them in different bands.

Moreover, the Sentinel S2 satellite can be divided into two principal sources, the L1C includes the raw data captured as it is, and the L2A uses the aforementioned raw data and applies some pre-processing techniques, resulting in a much more usable data with artifacts and weird noise removed. Because of this, for our project, we will focus mainly on the Sentinel S2 L2A.

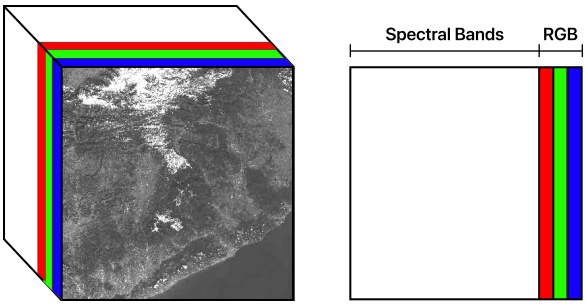



Fig. 1: Representation of Spatial and Spectral resolutions in a Sentinel S2 L2A sample.

This representation is composed of a two-dimensional plane corresponding to the mapping between coordinates and pixels of a sample, together with a third dimension that represents the different channels of the image. Within these channels, we distinguish between the Spectral bands and the Red, Green, and Blue bands, given that the latter are those within the electromagnetic spectrum visible to the human eye and are the most familiar representation model.

The Temporal resolution will be an added dimension to our problem. The Sentinel mission orchestrated by ESA has two identical S2 satellites, S2A and S2B respectively, which reduces the temporal resolution in half; five days, in which ten days is the time it takes a satellite to go around the planet.

An interpolation tool in the spatial, temporal, and spectral spaces under our representation could allow an effective modeling of the planet Earth and a significant advance in Remote Sensing. Ideally, we would stop having a discrete sample space to have a continuous space, where we could know the value of a point $p = f(x, y, z, t)$ for any pair of coordinates $(x, y)$, spectral band $z$ and time-stamp $t$.

## 1.1 Objectives

Due to the complexity of the field of work where this project is located, with so many conditions and fields of study, the need arises for a systematic way to organize and break down the project into smaller and more manageable components, different Phases will be established with different sub-objectives to be achieved, these phases will be incremental and sequential.

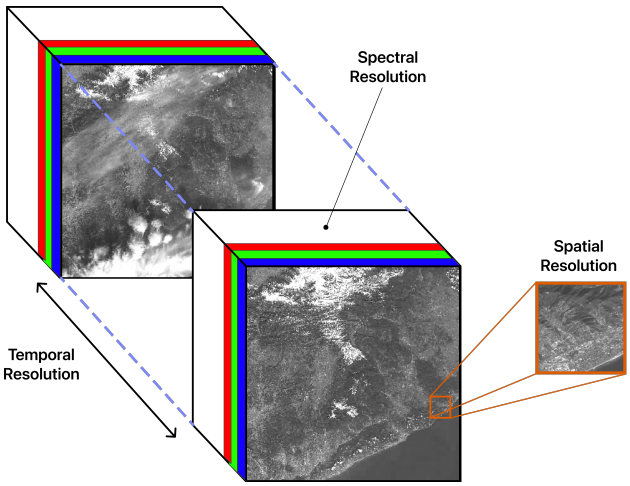



Fig. 2: Representation of Spatial, Temporal and Spectral resolutions in two Sentinel S2 L2A samples.

**Phase 1: Obtaining Data** Develop a Web service to obtain data from different providers and different formats where with universal queries we can communicate with different REST APIs. These universal queries will allow us to apply filters on a large set of data to obtain a subset of them. We will have to filter spatially in the form of coordinates, temporally in the form of date ranges, finally we will have to offer different bands from the suppliers separately.

1. Search for current suppliers.
2. Characteristics of each supplier.
3. Obtaining data from different APIs.
4. Support for other data (DSM, DEM) and popular Datasets.

**Phase 2: Data Processing Platform** Develop a platform for the processing and treatment of different types of data. Using ETL processes (Extract, transform, load) we will read a set of data, apply a calculation on them and save them in memory. Simple operations will need to be performed on large data sets in order to put them into context and will serve as input for more complex operations.

1. Read and View Bands, Masks, Metadata.
2. Square cropping operation.
3. Support for Shapefiles and Operation SHP cropping.
4. Support for Digital Elevation Models.

**Phase 3: Data Processing Techniques** During this phase, data processing techniques will be implemented in which the ground truth of the spatial, temporal and spectral resolutions will be altered. A research on the applied methods will have to be carried out and implemented on the data processing platform.

1. Overview of the State of the Art.
2. Spatial, Temporal and Spectral SR techniques.
3. Implementation of state of the art techniques in the application.

**Phase 4: Modeling** Finally, through everything built so far, the State of the Art technique will be studied in which they have been successful in performing a modeling of interest for our project as Implicit Learning, where experiments such as NeRF or LIIF have had good results. And, by combining the techniques of the previous phase, an approximation will be made to the total modeling of the problem.

1. Simple Interpolation Model:
   Baseline of the project. 1
2. Approach to a total Modeling:
   Local Implicit Image Function (LIIF).
3. Analysis and explanation of results.

## 2 State of the Art

To overcome the physical limitations of sensors, new techniques have been developed in Remote Sensing. Some of which can be leveraged for our use case on satellite imagery data.

### 2.1 Spatial Interpolation: Super-resolution Techniques

In 1991, this limitation gave rise to a whole new field of study within Computer Vision called Super-resolution, based on the problem of Single Image Super Resolution (SISR) [5] . This problem attempts to recover the high-frequency information, inherent in a High Resolution (HR) image, under the premise of treating a Low Resolution (LR) image as a High Resolution (HR) image that has been altered with a function degradation:

$$\mathrm{LR} = f(\mathrm{HR})\,. \tag{1}$$

The problem, by definition, is a problem with infinite solutions, which is why it is commonly called the "ill-poisoned problem".

**Interpolation** In the mid 1990s, the problem was tackled using Interpolation techniques, these are based on nearby pixels to perform the transformation from LR to HR. It is inevitable to lose high frequency information, very opposite but close colors will be smoothed. These techniques are usually fast since they can be implemented with simple operations.

**Example-based** In the decade of the 2000s, techniques based on Reconstruction emerged, in which we add prior knowledge of the reality of a certain area as a form of heuristic to guide our model to make better approximations to reality, we need, therefore , a source of truthful information as Ground Truth to be able to carry out the transformation of the LRs.

**Deep Learning** During the decade of the 2010s, the current of knowledge-based techniques was established, which use a large dataset of LR and HR data, contrary to the techniques based on Reconstruction, it does not need prior knowledge of the reality of a certain area, but learns to perform the mapping between LR and HR to predict the missing high-frequency information from any LR image.

More specifically, 2014 is the turning point in which Deep Learning is positioned as a new current trend to solve the SISR problem. The Super-Resolution Convolutional Neural Network (SRCNN) [1] outperforms bicubic interpolation with only a few training iterations and outperforms Reconstruction-based methods with relatively moderate training. Later in 2016, Liebel and Körner [4] applied the same SRCNN on Sentinel satellite images with promising results.


Fig. 3: Sample of Sentinel S2 from the UAB campus applied on ESRGAN to upsample x4 the spatial resolution.

From here Deep Learning is established as the standard and variants of the SRCNN emerge with new architectures, such as the Enhanced Super-Resolution Generative Adversarial Networks (ESRGAN) [7] which adds a generative approach to CNNs, where through an unsupervised learning technique in which it grants the power to automatically discover and learn regularities or patterns in the input data so that the model can be used to generate or produce new examples that plausibly they could have taken from the original data set.

### 2.2 Continuous Representation

In recent years efforts have been devoted to finding a continuous image representation, an example of this is Learning Continuous Image Representation with Local Implicit Image Function (LIIF) [8] , where each image is represented as a two-dimensional feature map and the same decoding function for an entire image. Given any coordinate, based on the nearest neighboring features it will provide a new RGB value.

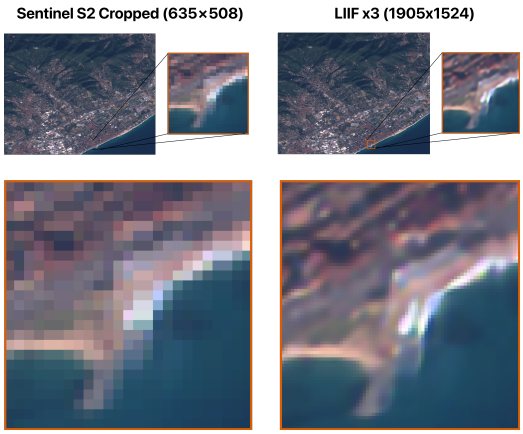



Fig. 4: Sample of Premià de Mar's Sentinel S2 applied to the LIIF model for spatial super-resolution in the form of a continuous representation of an RGB image.

This model allows for a theoretically infinite spatial inter-

polation given its passage through a continuous LIIF representation in which it expresses the coordinates of an image with functions. An extra method is therefore necessary to carry out a possible own extrapolation towards the temporal horizon, or a temporal interpolation by adding an image of the same characteristics with a different temporal resolution.

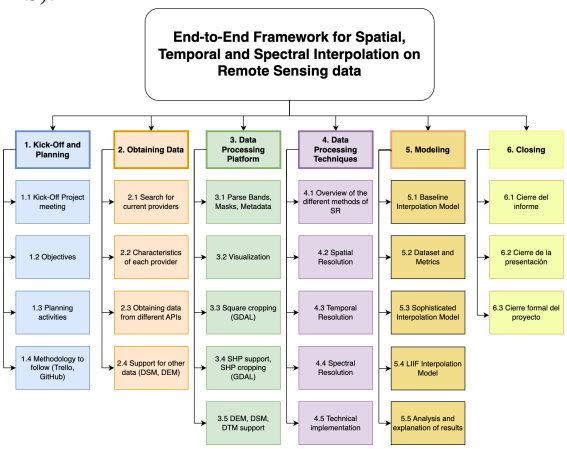
### 2.3 Temporal Interpolation

Temporal interpolation has been a much-attacked problem in the field of Video Motion, where models such as FILM: Frame Interpolation for Large Motion [2] from Google, try to temporally interpolate two images spaced in a range of a few seconds. In a satellite image these magnitudes of seconds become days since it is difficult to demonstrate how a change in a period of time affects a specific area.


Fig. 5: FILM inter-frame interpolation over a Sentinel S2 sample of the Delta, an area prone to altering its spatial resolution with new artifacts.

## 3 Methodology

The present is a project within a large area that directly and indirectly involves a multitude of challenges and fields of study, to avoid falling into a rabbit hole one must be very careful when define the goals and always keep them in mind. This is why, as a direct consequence, the objectives have been defined following the Work breakdown structure (WBS):




In order to be able to carry out comprehensive control of these goals, provisional start and end dates have been added to them, and they have been added to a Trello board to be able to follow the iterations on a Kanban methodology. This allows you to see at a glance the overall status of the project in each iteration.

This allows us to represent a Gantt Chart identifying our critical tasks. In parallel with the tasks, the writing and delivery of the various reports that are carried out during the realization of the TFG has also been specified.

For the development of the project, a version control system (VCS) is used, such as Git, hosted remotely on GitHub. There will be a private central repository where development of the services will take place. And, in addition, the services will have a mirror repository where they will be open to the public, in Open Source.

## 4 Experiments and Results

The experimental work-load will assemble the construction of three models. Similar to the incremental methodology followed by the spatial superresolution during the decades from the 1990s to the 2010s, we will first develop a "**Baseline Interpolation model**" that will serve as the baseline for all the experiments. Later on, we will develop a new model **based on the LIIF technique**, that ideally, will make a continuous interpolation and thus, infinite by definition.

### 4.1 WorldStrat Dataset Pre-processing

In order to be able to compare models we will make usage of a set of established metrics in the Computer Vision realm, and, to be able to obtain metrics we need a source of Ground Truth. In the case of Remote Sensing, there will never be such source, instead we try to set as the Ground Truth sources of information with a low Ground Distance Sample (GDS), for example Drones (Satellogic) or other Satellite Missions (ESA's SPOT 6/7).

During the 36th Conference on Neural Information Processing Systems (NeurIPS 2022) Track on Datasets and Benchmarks, the European Space Agency (ESA) Phi-Lab as part of the ESA-funded QueryPlanet project, presented the WorldStrat Dataset [3], a dataset of nearly 10000 $km^2$ of free high-resolution and matched low-resolution satellite imagery of unique locations. Each high-resolution image (GDS of 1.5 m/pixel) comes with multiple temporally-matched low-resolution images from the freely accessible lower-resolution Sentinel-2 satellites (GDS of 10 m/pixel).

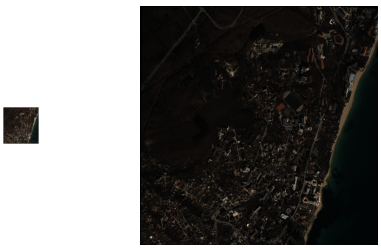


Fig. 6: A WorldStrat dataset sample of Varna, Bulgaria. The Sentinel S2 L2A low resolution (left) and a SPOT 6/7 high resolution (right) sample. Right after doing the common color conversion to RGB.

Therefore, if we consider the GDSs previously mentioned. In order to do the transformation of the low-resolution image to the high-resolution bounds we will need to upscale the low-resolution image by a coefficient of $\frac{10}{1.5} = 6.\widehat{66}$ m/pixel, hence, we need to upscale the low resolution image by a factor of $6.\widehat{66}$ times its size in order to acquire the size of the high resolution image. The bits of

representation or color model representation are other factors to take into account.

#### 4.1.1 Common Color model

First off, we need a common color model representation for both of the samples. In the low resolution Sentinel S2 L2A image, we can compose a True Color Image with the mapping suggested by its own documentation: (RGB) = (B04, B03, B02). The SPOT 6/7 high resolution image is RGBN by default, with an added extra band known as Near-Infrared band (NIR) which has been used to improve the high-resolution sample by Pan-sharpening. In this case, to compose the True Color Image for the High-Resolution sample, we will crop the extra NIR band. As a result of this we can display both images in the RGB color model (Figure 6).

#### 4.1.2 Adjustments

From Figure 6 we can also note out that the images are very dark. This is because the orders of magnitude used in the bits of representation. In Remote Sensing, it is common to use many bits to represent information to ensure precision and minimal loss of information. From the histogram of the SPOT 6/7 high resolution image of Varna, Bulgaria we can notice that a lot of the high-end spectrum of the picture is not even used:

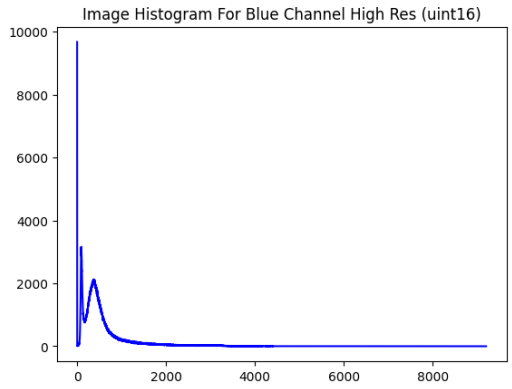



Fig. 7: The histogram of the Blue Channel for the SPOT 6/7 high resolution image of Varna, Bulgaria.

More formally, there is a notorious difference between the Maximum and the Mean for all the pixels of each channel of the image (Table 1). Note that this problem is worse than one could think, having dark spots in an image will make the results of the metrics miss-leading, and we will perform better than we should when doing the interpolation, causing a lot of False-Positives (see Appendix A.1).

| | Blue channel | Red channel | Green channel |
|---|---|---|---|
| Max | 8071 | 7211 | 6778 |
| Mean | 571.11 | 519.47 | 446.06 |

Table 1: The high resolution SPOT 6/7 is represented in uint16, 16 bits of representation. Therefore the Max and Mean values shown in the table are of type uint16.

In order to solve this problem we will try to reduce it to an already established problem. First of all, we will take advantage that the Sentinel S2 L2A sensor already does a pre-processing step adjusting the contrast without losing information (as mentioned in Section 1: Introduction). Even though, other techniques, such as Histogram Equalization, had been used and delivered great results at a first glance, we cannot take the grant to modify the ground truth and we shall stay rigorous.

Therefore, a low-resolution Sentinel S2 L2A sample will be used as a reference to adjust the brightness of the remaining images of the set. By applying Histogram Matching (hist_match) we will balance the brightness of all the images of the set in a linear fashion, without interfering directly into the color. Note that this technique will be applied only to the V channel of the HSV (Hue, Saturation, Value) color model representation which is widely used in color theory and color transfer.

$$\text{RGB} \rightarrow \text{HSV} \rightarrow \text{hist\_match(V)} \rightarrow \text{HSV*} \rightarrow \text{RGB*}$$

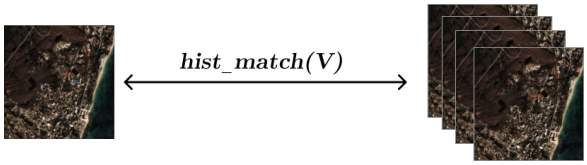



Fig. 8: Histogram matching (1-N) between the selected LR reference image and all other images.

As a result of this pre-processing step (see Figure 9) we adjusted and balanced the brightness for all the images of the set with respect to each other without interfering directly into the color, therefore, our metrics should be more precise and reliable along the way.

Fig. 9: A pre-processed WorldStrat dataset sample of Varna, Bulgaria. The Sentinel S2 L2A low resolution (left) has been adjusted and used as reference for the SPOT 6/7 high resolution (right) sample. Compare it with the initial sample in Figure 6.

### 4.2 Metrics

If we take a look at the NTIRE challenge, the largest benchmark of current state of the art Superresolution, we can see that the predominant metric that every model takes into account is Structural similarity (SSIM), which is used to measure the similarity between two images (the upscaled one and the ground truth). This metric is biased towards factors of human perception, such as texture, when evaluating [6], and, may not be suitable for Remote Sensing, where we

are interested high precision of pixel values rather than the overall result.

#### 4.2.1 Mean Squared Error (MSE)

Mean Squared Error is the foundational metric used to determine the difference between two images. Other metrics use MSE underneath. For each pixel value it will calculate the squared difference, so-called error, to avoid negative values, and finally it will perform the mean over the pixels of the image. This is a very numerical metric of telling whether the interpolated image is different from the ground truth or not. From this, a more natural way of deducing the quality of our interpolation is required.

Given a High Resolution image HR and its Interpolation image $\tilde{\mathrm{I}}$, both of size m $\times$ n:

$$\mathrm{MSE} = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} \big(\mathrm{HR}(i,j) - \tilde{\mathrm{I}}(i,j)\big)^2 . \qquad (2)$$

#### 4.2.2 Peak Signal Noise Ratio (PSNR)

Peak Signal Noise Ratio (PSNR) is a metric widely used in compression codecs but we can apply it to our scope of work with Interpolation on mind. It measures a signal, defined by the ground truth image, and a noise, defined by the difference between the ground truth image and the corresponding interpolation. It then calculates the ratio between the maximum possible amount of power, the maximum pixel value (255), and the needed power of noise to corrupt the given signal. This provides a quantitative measure of the quality of the overall interpolated image.

It is expressed in decibels (dB). The higher it is, the better our low-resolution image has been reconstructed. A PSNR of 0 is the worst reconstruction scenario, where the noise has infected all our signal, and a PSNR of 255 is the best reconstruction with a Mean Squared Error (Equation 2) of zero.

Given a High Resolution image HR and its Interpolation image $\tilde{\mathrm{I}}$, both of size mxn, and the MSE then we define:

$$\mathrm{PSNR} = 20 \log \frac{255}{\mathrm{MSE}} . \qquad (3)$$

Other publications on remote sensing also use a variant of PSNR, cPSNR [6], because it takes into account the necessary shifting to cover the average geolocation accuracy of about 60m (±50m standard deviation). However, because we will be working with the dataset, as we don't have the ground truth of a random sample at a given location and time, these dataset images have been shifted, if necessary, to minimise the error.

### 4.3 Baseline Interpolation Model

Our first approximation to Spatial, Temporal, and Spectral Interpolation will be with the so-called **Baseline Interpolation Model**, where given four coefficients $(x, y, z, t)$ for each resolution respectively, we will generate the corresponding Interpolation.

#### 4.3.1 Spatial and Spectral Resolutions

Given a high-resolution image HR and a low-resolution image LR, the Spatial and Spectral scale coefficients $(x, y, z)$ should be able to transform from the LR dimensions to the HR dimensions so that we can compare this interpolation with the corresponding ground truth.

#### 4.3.2 Adding the Temporal Resolution

However, with temporal resolution, we add an extra dimension to the problem. The $t$ coefficient represents the number of inter-frames generated between each pair of ground truth frames. Figure 5 has a temporal coefficient $t$ of 1 because it generates one new inter-frame between each pair of original frames.

In order to be able to perform temporal interpolation, we will use the previous model with the additional feature of being able to define time ranges. Each ground-truth frame from the dataset is a time unit, and we interpolate a new frame for any value between each time unit. This flexibility allows the model to generate an image at any time between a range of times. It is, therefore, capable of generating infinitely many possible images:

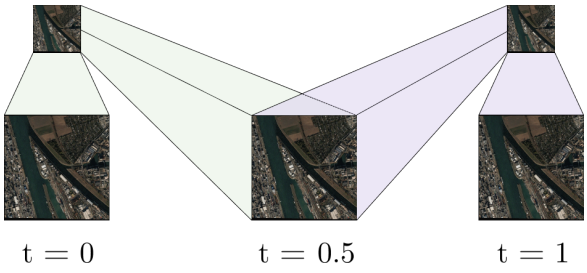



Fig. 10: Given two ground truth samples, a newly image is interpolated at $t = 0.5$.

To evaluate the newly added temporal resolution, we can take advantage of the temporal distribution of the data set. Each sample has metadata with the date of acquisition, following a similar methodology as shown in Fig. 12, N samples of LR samples will be chosen so that the temporal interpolation of all of them generates an interpolation at time $t$, where $t$ will be the exact date of acquisition of our HR ground truth sample.

Thus, given a high-resolution sample of the date of acquisition of $t_{\mathrm{HR}}$ and $n$ LR samples of date of acquisition $t_1$, $t_2$, …, $t_n$ respectively, and as a pre-requisite we sort them by date, then, we can say that it must satisfy the following Equation 4 to be able to generate an adequate temporal interpolation that will be later compared with the HR ground truth image.

$$\underbrace{t_1 + \sum_{i=1}^{n-1} \frac{\sqrt{(t_i - t_{i+1})^2}}{2}}_{t \text{ value of the generated interpolation}} = t_{\mathrm{HR}} . \qquad (4)$$

Note that the pre-requisite of having a sorted array of $n$ LR samples as input is to simplify the check, since this check could be implemented in $O(n \log n)$ time complexity.

#### 4.3.3 Temporal Dataset

A subset of 31 images has been selected from the dataset to ensure that the model is robust and applicable to real world data (and within our computing resources). This subset will evaluate the results and we make sure to include a variety of locations and labels, such as the International Panel on Climate Change (IPCC) classification or the LCCS and SMOD classifications that include vegetation typology and urban density. It should also be noted that these samples may come from different sources, such as Amnesty International, whose intention is to identify destroyed villages in areas of conflict.

| **Sample** | **Location** | **Label** (LCCS, SMOD) |
|---|---|---|
| Amnesty POI-6-3-3 | Pyongyang, North Korea | Grassland, Sub-Urban |
| Amnesty POI-8-1-1 | Ji-Paraná, Brazil | Forest, Low density |
| Amnesty POI-13-1-1 | N'Djamena, Chad | Grassland, Urban |
| Landcover-8732 | Nakhon Ratchasima, Thailand | Agriculture, Very low density |
| Landcover-14956 | Rivne, Ukraine | Agriculture, Very low density |
| Landcover-31746 | Fada, Chad | Agriculture, Very low density |
| Landcover-461460 | Jasper, United States | Grassland, Semi-dense |
| Landcover-769356 | Tianjin, China | Settlement, Low density |
| Landcover-769486 | Catania, Italy | Rural, Very low density |
| Landcover-770156 | Burgas, Bulgaria | Settlement, Urban, Water |
| Landcover-770516 | Mannheim, Germany | Settlement, Urban Centre |
| Landcover-771435 | Tekirdağ, Turkey | Settlement, Sub-Urban |
| Landcover-771536 | Beijing, China | Settlement, Urban, Dense |
| Landcover-771547 | Tabriz, Iran | Rural, Low density |
| Landcover-772421 | Luhansk, Ukraine | Settlement, Very low density |
| Landcover-772423 | Lakeland, United States | Settlement, Urban, Water |
| Landcover-777404 | Imperia, Italy | Settlement, Urban, Dense |
| Landcover-777410 | Chelyabinsk, Russia | Settlement, Urban Centre |
| Landcover-1864782 | Novokuznetsk, Russia | Forest, Very low density |
| Landcover-1865521 | Tyumen, Russia | Forest, Very low density |
| UNHCR-7977 | Latakia, Syria | Settlement, Urban Centre |
| UNHCR-AFGs000003 | Herat, Afghanistan | Settlement, Urban Centre |
| UNHCR-AFGs003914 | Kabul, Afghanistan | Agriculture, Very low density |
| UNHCR-CODs026828 | Mpondwe-Lhubiriha, Uganda | Agriculture, Very low density |
| UNHCR-CODs026832 | Sabha, Libya | Forest, Tree-cover |
| UNHCR-NERs009694 | Ngaoundéré, Cameroon | Agriculture, Very low density |
| UNHCR-NERs009697 | Yaoundé, Cameroon | Settlement, Urban Centre |
| UNHCR-NGAs036019 | Yola, Nigeria | Water, Very low density |
| UNHCR-NGAs036021 | Maiduguri, Nigeria | Agriculture, Very low density |
| UNHCR-TGOs003383 | Phitsanulok, Thailand | Agriculture, Very low density |
| UNHCR-THAs001547 | Kanchanaburi, Thailand | Agriculture, Very low density |

Table 2: The selected subset of images with the associated corresponding location and labels.

Now that we have defined our own subset of images, in order to provide temporal precision, for each sample we will perform the aforementioned Temporal Check (Section 4.3.2) which will glance the 5 low-resolution samples whose single interpolation is closest to the high-resolution date of acquisition (see Appendix A.2).

Then, we will perform a Train-test split, having 21 images in the train set and 10 images on the test set, that will persist over the current work. Hence, all the displayed results will be of the test set.

Finally, we perform a Train-Test split, which will randomly split our Temporal Dataset into a Train set of 21 samples, and a Test set of 10 samples. From now on, all the results will be computed over the Test set.

#### 4.3.4 Results

When evaluating the interpolations, the temporal resolution precision (Section 4.3.2) does not affect the results, and, other factors are more important such as the quality of the Sentinel S2 L2A sample or the presence of clouds and artifacts. The average MSE and PSNR scores are logged for each of the 5 LR temporal samples. We used the **Bicubic interpolation method** which is considered the standard when doing comparisons in the Super-resolution realm.

| Test-set samples | x2 | | x4 | | x6 | |
|---|---|---|---|---|---|---|
| | MSE | PSNR | MSE | PSNR | MSE | PSNR |
| Amnesty POI-13-1-1 | 381.23 | 22.36 | 461.16 | 21.52 | 490.31 | 21.25 |
| Landcover-31746 | 208.19 | 24.97 | 258.37 | 24.02 | 274.22 | 23.76 |
| Landcover-769356 | 393.17 | 22.24 | 441.27 | 21.73 | 456.63 | 21.58 |
| Landcover-769486 | 231.16 | 24.6 | 281.46 | 23.71 | 300.29 | 23.42 |
| Landcover-771536 | 404.31 | 22.2 | 458.0 | 21.63 | 476.51 | 21.45 |
| Landcover-1865521 | 307.11 | 23.56 | 352.51 | 22.89 | 370.24 | 22.65 |
| UNHCR-7977 | 399.99 | 22.12 | 465.38 | 21.46 | 487.06 | 21.26 |
| UNHCR-NERs009694 | 298.76 | 23.47 | 347.19 | 22.79 | 365.94 | 22.56 |
| UNHCR-NGAs036019 | 352.65 | 23.07 | 380.54 | 22.68 | 391.58 | 22.54 |
| UNHCR-THAs001547 | 528.38 | 21.72 | 559.54 | 21.36 | 572.88 | 21.21 |
| | **448.92** | **23.09** | **464.11** | **22.87** | **464.37** | **22.88** |

Table 3: Results of the selected subset of images indicating the average MSE↓ and PSNR↑ (expressed in dB) scores obtained from all its possible interpolations.

Finally, we can evaluate this Baseline model with an average **PSNR score of 22.53 dB** (see Table 3).

### 4.4 LIIF Interpolation Model

Local Implicit Image Function (LIIF) [8] changes the way we think about images. Traditionally we represented images with a two-dimensional array of pixels in a discrete manner, but LIIF is built from the promise that each pixel of an image can be described as a continuous function of its coordinates and its neighbour features. The main advantage of this is that with our new continuous representation we are no longer constrained by resolution, and we can generate **arbitrary resolutions** for any image, even for upsample scales that the model wasn't even trained.

The **LIIF Continuous Representation** of an image will consist of a series of latent codes, a space in which items resembling each other are positioned closer to one another, distributed in spatial dimensions. Therefore we will have a set of encoding-decoding functions that will do the transformation between discrete image representation in Spatial Space and LIIF Continuous representation in Latent Space mapping coordinates to RGB values.

#### 4.4.1 Encoding Function (EDSR)

The encoder is the responsible for generating a feature map representing the input image in a continuous fashion through latent codes, it will be parameterized through an Enhanced Deep Residual Network (EDSR) omitting the upsampler module (that will be our LIIF decoder). The first layer of this Network is named Head and will be the one to perform the transformation from the input to a set of features, then will go through a deeper and wider architecture of 16 ResBlocks and 256 channels to improve performance.

In order to add the temporal dimension, we will perform three dimensional convolutions during the whole EDSR pipeline. This will make our model parameters and the features generated explode by a temporal factor, depending on how many temporal samples we use as the input.

As a result of the training we will have our encoding function with the guessed weights that will make the transformation between the discrete image and the LIIF continuous representation in Latent space.

#### 4.4.2 Decoding Function (MLP)

The decoding function takes a coordinate as an input, queries for local latent codes and then predicts the RGB value as the output for that coordinate. For this purpose, a Multi-Layer Perceptron with 5-layers, ReLU activation and hidden dimensions of 256, will be used.

**Feature Unfolding** When querying for local latent codes, the resulting latent code will be a result of applying a concatenation of the $3 \times 3 \times t$ neighboring latent codes.

**Ensemble for Local Predictions** In order to achieve smooth transitions between different independent, but close, predictions, we will apply 3 predictions from each of the spatial, temporal or spectral dimensions, thus increasing our features by a factor of 9, and merge them together so that in the end-result, no rough transitions are made by the model.

Once we have a set of ensembled unfolded latent codes, the Multilayer Perceptron will make the transformation between LIIF continuous representation in Latent space and the resulting discrete image.

#### 4.4.3 Experiments

One of the main hypothesis to test is if adding another dimension (such as time) will achieve better results in LIIF. To do this we have to compare two models within the same conditions. We will first develop a "**Default LIIF**" model that will essentially be the default LIIF model but applied into our temporal dataset, it won't consider time and will be focused in spatial super-resolution, therefore we say that this model supports **2 dimensions**. Then, we will develop the "**Temporal LIIF**" model in which we will introduce all the necesary changes to add another dimension, it will be focused in spatial and temporal super-resolution, therefore we say that this model supports **3 dimensions**.

The only change on LIIF that will affect both the models is to instead of following the default methodology of training, which is to take any sample in the train set and keep that sample as high-resolution and generate the low-resolution sample by down-scaling it. In our case, because we already have the high-resolution and low-resolution pre-established from the dataset, we will modify the LIIF PyTorch dataloader to include both the 5 temporal samples that will act as the LR input and the Ground Truth HR.

Both models will have 120 epochs of training. With the same train-set and test-set, and a repetition of 20 on the train-set, that is, each sample will be repeated 20 times, and thus our train-set of originally 21 samples, will now have 420 samples.

We will then do some experiments to determine **the size of the input** on training, and **temporal strategies** to achieve an arbitrary temporal scale, just like with the spatial resolution.

##### 4.4.3.1 Default LIIF (2 dimensions)

In order to see more precisely how time affects the model, we trained a LIIF without taking time into account with the default configuration that the authors provide in the paper which is called Baseline EDSR configuration. To be able to compare precisely, when either training or testing, we will receive the 5 temporal LR samples as the input, keeping with the middle one, that in most of the cases is the one temporally closer to the HR's date of acquisition, and ditch the 4 temporal LR samples left (see Figure 12).

Because this model is trained only with a single LR temporal sample, at first sight it has an obvious drawback that it does not know time.

##### 4.4.3.2 Temporal LIIF (3 dimensions)

During training, we obtain the high-resolution Ground Truth sample and the corresponding 5 temporal LR samples, then down-scale the Ground Truth to LR dimensions and apply a random crop for a random region of the image, this way the model will focus on a random region of the image thus avoiding artifacts or be biased by any remarkable features not desired for the network's overall knowledge, we will see how applying random crops for $48 \times 48$, $72 \times 72$ and $128 \times 128$ dimensions affects the overall result.

Also some experimental runs has been done regarding on whether, during training, which temporal samples to take into account. A complete view of the temporal set will make more generic samples with common parts of the image, but if we train with randomly changing dynamic groups we could have better results because the model will learn the variations between each temporal samples depending on the situation. We think that by randomly selecting different temporal images as input, incentives the model to learn the mappings between each temporal dimension.

**Complete** This model takes into account all the temporal samples at all times. Thus, our temporal set will consist of one unique set of all the 5 samples.

$$T_{complete} = \Big\{\{1,2,3,4,5\}\Big\}$$

**Random Adjacent Groups** This model takes into account a random range between 1 and 5 temporal samples, this range will therefore form an Adjacent Group of length between 1 and 5.

$$T_{groups} = \Big\{\{1\},\{1,2\},\{1,2,3\},\dots\Big\} \supseteq T_{complete}$$

**Random Adjacent Pairs** This model takes into account random adjacent pairs of the temporal samples. We could sum this up as the previous Random Adjacent Groups for Adjacent Groups of length 2.

$$T_{pairs} = \Big\{\{1,2\},\{2,3\},\{3,4\},\dots\Big\}$$

**Random Single** This model takes into account a single random temporal sample without the requirement of them being adjacent but respecting its original ordering.

$$T_{random} = \Big\{\{1\},\{2\},\{3\},\{4\},\{5\}\Big\}$$

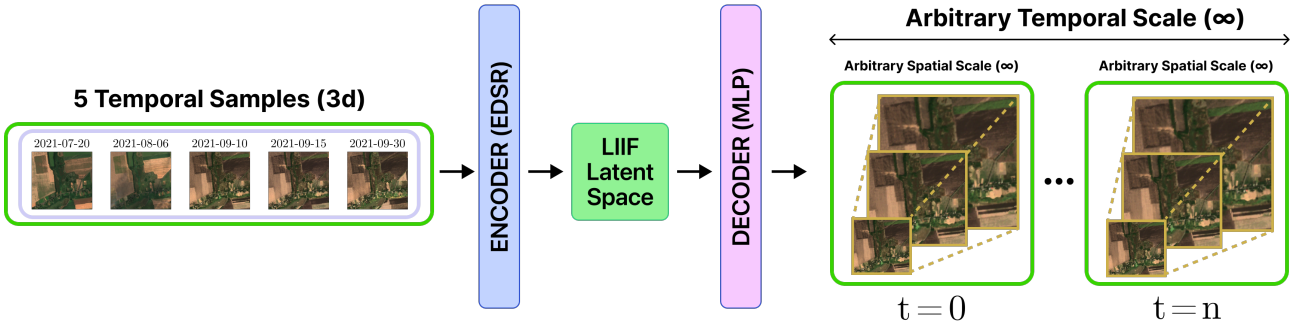



Fig. 11: A general overview of Temporal LIIF, which consists on a modified version of the original LIIF implementation with an added dimension. In our case, we use that added extra dimension for Time, resulting in a model capable of interpolating any scale and temporal factor, hence, an infinite interpolation model of space and time for Remote Sensing. We can specify any arbitrary resolution and value of time $n$.


Fig. 12: In blue, the 5 temporal samples. In green, the selected "middle" sample. The architecture of Default LIIF supports 2 dimensions and therefore is not capable of accepting inputs of 3 dimensions, this is why we only end up selecting one sample as the input of the neural network.

#### 4.4.4 Results

Each of the models shown in Table 4 had been trained with 120 epochs of training. Each of the results correspond to the average PSNR score for all the samples on the test-set (similar to Table 3) and the chosen scales $\times 2$, $\times 4$ and $\times 6$. We say that the later is "Out-of-distribution" because we did not include such scale during training.

| Models | In-distribution | | Out-of-distribution |
|---|---|---|---|
| | $\times 2$ | $\times 4$ | $\times 6$ |
| Bicubic | 23.09 | 22.87 | 22.88 |
| Default LIIF | 23.70 | 22.92 | 22.67 |
| Temporal LIIF[1] | | | |
| 48×48 | 24.79 | 23.85 | 23.55 |
| 72×72 | 24.91 | 23.95 | 23.64 |
| 128×128 | 24.89 | 23.95 | 23.65 |
| Temporal LIIF[2] | | | |
| Complete | 24.89 | 23.95 | 23.65 |
| Groups | 24.96 | 23.99 | 23.68 |
| Pairs | 21.32 | 20.80 | 20.62 |
| Single | 22.20 | 21.60 | 21.40 |

[1] Trained with a Complete temporal strategy.
[2] Trained with a random crop region dimensions of 128 × 128.

Table 4: Quantitative ablation study for fine-tuning the parameters of the different models presented through this present work. The results are expressed in **PSNR↑ (dB)**.

Bicubic 4.3 is outperformed by Default LIIF 4.4.3.1 on $\times 2$ and $\times 4$ scales, however for Out-of-distribution scales $\times 6$ the model is not capable of outperforming the Bicubic.

In order to determine which random region crop dimensions to choose we decided to go for the model that is more consistent through the overall training, we obtain an average PSNR score for all 120 epochs of training of 23.57 dB, 23.57 dB and 23.79 dB, for the corresponding dimensions of 48×48, 72×72 and 128×128. Therefore, **a random region of 128×128 is more stable through the process and it will be the dimensions chosen for the next experiments**.

Finally, from the experiment done to determine the temporal strategy we see that the **Random Adjacent Groups temporal strategy** slightly outperforms the Complete temporal strategy. Thus, we can see that we were indeed correct in the hypothesis made during Section 4.4.3.2 that randomly selecting different temporal images as input, incentives the model to learn the mappings between each temporal dimension. However, the Random Adjacent Pairs and the Random Single temporal strategies achieve worse results.

From this present section we conclude that the best model is the **Temporal LIIF with a random region crop of 128×128 and a Random Adjacent Groups temporal strategy**, in Table 5 we can see the final overall results.

| Models | In-distribution | | Out-of-distribution |
|---|---|---|---|
| | $\times 2$ | $\times 4$ | $\times 6$ |
| Bicubic | 23.09 | 22.87 | <u>22.88</u> |
| Default LIIF | <u>23.70</u> | <u>22.92</u> | 22.67 |
| Temporal LIIF | **24.96** | **23.99** | **23.68** |

Table 5: Quantitative ablation study on the different models presented through this present work. The results are expressed in **PSNR↑ (dB)**. The best performance is shown in **bold** and the second best <u>underlined</u>.

## 5 Conclusions

In this present work we achieved the building, from the ground up, of an **End-to-End Framework for Continous Space-Time Super-Resolution on Remote Sensing data**. We built an effective web service that interacts with multiple APIs to provide us from data, as well as a processing cross-platform app capable of processing that same data. We rigorously pre-processed the data without losing precision by taking advantage of already existing data from other

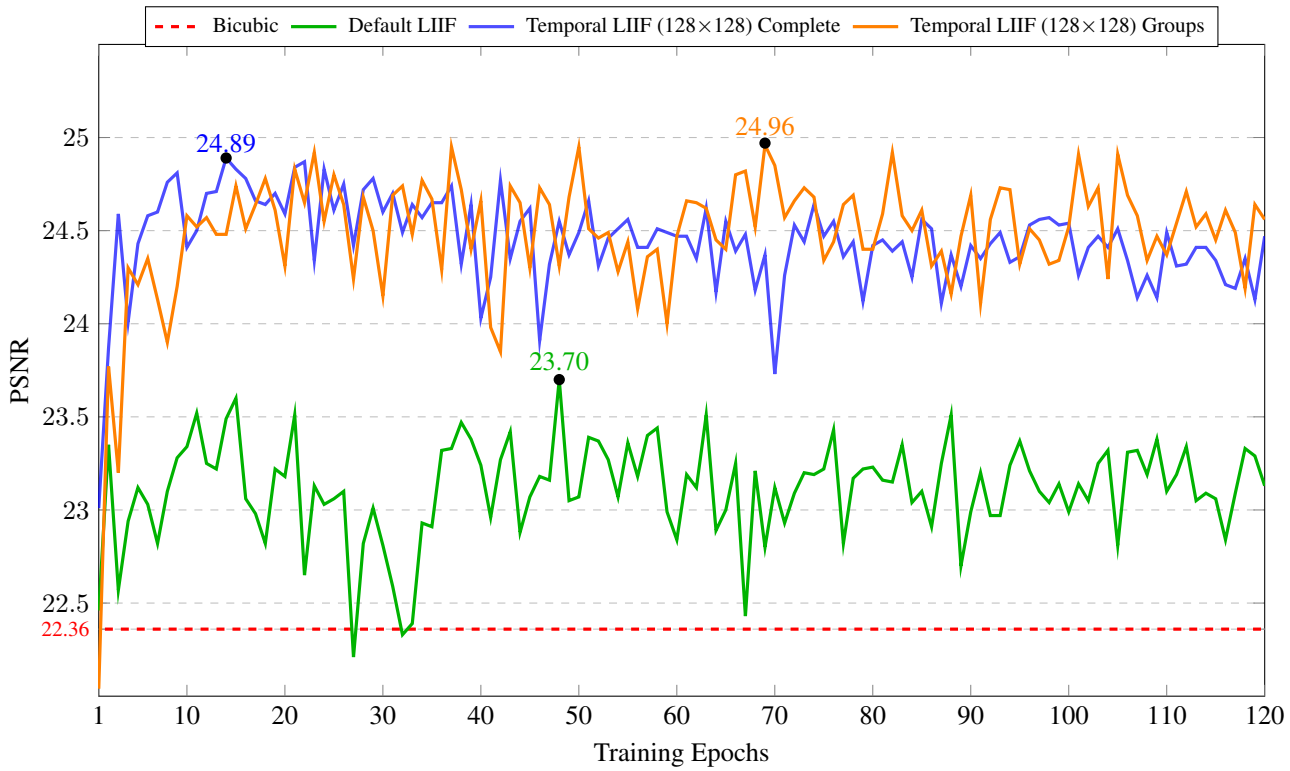



Fig. 13: A comparison for a Spatial scale of $\times 2$ between Bicubic (4.3), Default LIIF (4.4.3.1) and the Temporal LIIF (4.4.3.2) for a random range region of $128 \times 128$ and a temporal strategy of Complete and Random Adjacent Groups. The PSNR score obtained on the test-set for each training epoch is shown.

sources and built a brand new sub-set of data that is temporally precise. We then have shown how applying machine learning into the process represents a great qualitative leap in the results, and, going further, taking time into account by adding another dimension in Local Implicit Image Function (LIIF) [8] is indeed effective and leads to even better results. Finally we have seen that by randomly selecting different temporal images as input, incentives the model to learn the mappings between each temporal dimension. This resulted in a model capable of **interpolating any scale and temporal factor, hence, an infinite interpolation model of space and time for Remote Sensing**.

## 6 Acknowledgment

I would like to express my gratitude towards the tutor Felipe Lumbreras for encouraging me to push myself and providing guidance throughout the process, and the Computer Vision Center (CVC) since this work has been carried out jointly with a stay as a Research Intern in the latter. This work has been partially supported by the Spanish Ministry of Science and Innovation under the BEYOND project.

## References

[1] C. Dong, C. C. Loy, K. He and X. Tang, "Image Super-Resolution Using Deep Convolutional Networks," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 38, no. 2, pp. 295-307 (2016) 3

[2] F. Reda, J. Kontkanen, E. Tabellion, D. Sun, C. Pantofaru, B. Curless, "Film: Frame interpolation for large motion," in European Conference on Computer Vision (ECCV), pp. 250–266 (2022) 4

[3] J. Cornebise, I. Oršolic, F. Kalaitzis, "High-Resolution Satellite Imagery: The WorldStrat Dataset – With Application to Super-Resolution," in Conference on Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track (2022) 4

[4] L. Liebel, M. Körner, "Single-image super resolution for multispectral remote sensing data using convolutional neural networks," in ISPRS on Remote Sensing and Spatial Information Sciences, vol. XLI-B3, pp. 883-890 (2016) 3

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

[6] M. Märtens, D. Izzo, A. Krzic, D. Cox, "Super-Resolution of PROBA-V Images Using Convolutional Neural Networks," in Astrodyn 3 pp. 387–402 (2019) 5

[7] X. Wang, K. Yu, S. Wu, J. Gu, Y. Liu, "ESRGAN: Enhanced Super-Resolution Generative Adversarial Networks," in: L. Leal-Taixé, S. Roth (Eds.), L. Leal-Taixé, S. Roth, Computer Vision – ECCV 2018, Springer, Cham, pp. 63–79 (2019) 3

[8] Y. Chen, S. Liu, and X. Wang, "Learning continuous image representation with local implicit image function," in IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 8628–8638 (2021) 3, 7, 10

# APPENDIX

## A.1 Evaluating pre-processing

In Section 4.1: *WorldStrat Dataset Pre-processing* we perform an Histogram Matching over a reference image with the other images of the set in order to adjust our images.

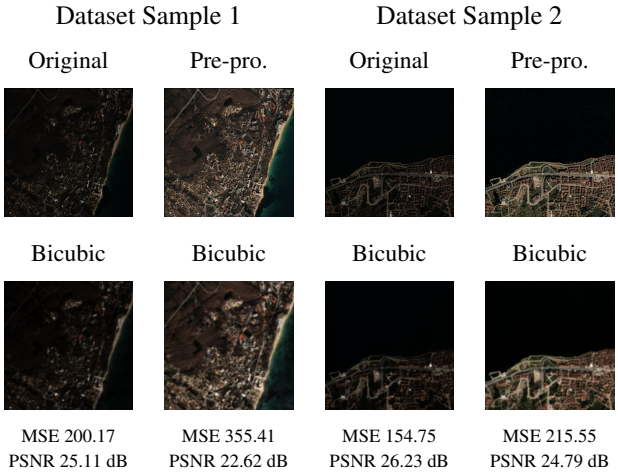



Fig. 14: A metric comparison of interpolation with the original data and the pre-processed data.

We can do the same interpolation for both the Original sample and the Pre-processed sample and we confirm the hypothesis said during Section 4.1 that, indeed, we perform worse once we have pre-processed the data. This is good because we avoid having False Positives and thus, having inaccurate metrics.

## A.2 Dataset Temporal adapted

As stated in Section 4.3.2, we need a way to temporally evaluate the results. By applying Equation 4 onto all the possible combination of 5 images of any given sample we can find the Interpolation that approaches the most at the Ground Truth's date of acquisition. Table 6 shows the result of applying this onto our subset, there are some perfect interpolations that should yield the exact date of acquisition, such as Amnesty POI-8-1-1, and there are others that are some days apart, such as UNHCR-AFGs000003. There is a clear outlier in the data-set which is UNHCR-7977 in which the Ground Truth $t$ value is two and a half years apart, this is an obvious error on the dataset. By any means, not taking the outlier into account we consider the subset temporally correct we consider the subset temporally correct as they approximate the Ground Truth date of acquisition by an average of 2.9 days without considering the outlier, and by an average of 32.03 days considering the outlier.

| Sample | Selected Images (in this order) | Interpolation $t$ value | Ground Truth $t$ value | Diff (days) |
|---|---|---|---|---|
| Amnesty POI-6-3-3 | 4, 2, 1, 8, 7 | 2017-10-11 | 2017-10-07 | 4 |
| Amnesty POI-8-1-1 | 4, 1, 2, 3, 5 | 2018-07-01 | 2018-07-01 | 0 |
| Amnesty POI-13-1-1 | 6, 3, 2, 5, 7 | 2020-11-12 | 2020-11-11 | 1 |
| Landcover-8732 | 4, 1, 6, 3, 5 | 2021-02-05 | 2021-02-04 | 1 |
| Landcover-14956 | 4, 7, 6, 1, 8 | 2019-10-18 | 2019-10-20 | 2 |
| Landcover-31746 | 6, 3, 2, 4, 5 | 2019-03-01 | 2019-03-02 | 1 |
| Landcover-461460 | 5, 8, 1, 6, 7 | 2018-07-20 | 2018-07-14 | 6 |
| Landcover-769356 | 2, 4, 1, 3, 5 | 2018-03-26 | 2018-03-24 | 2 |
| Landcover-769486 | 4, 7, 1, 3, 2 | 2020-03-13 | 2020-03-12 | 1 |
| Landcover-770156 | 12, 6, 1, 7, 4 | 2020-03-13 | 2020-03-12 | 1 |
| Landcover-770516 | 11, 6, 1, 7, 15 | 2018-11-16 | 2018-11-18 | 2 |
| Landcover-771435 | 12, 10, 2, 3, 8 | 2019-03-26 | 2019-03-26 | 0 |
| Landcover-771536 | 6, 1, 5, 4, 7 | 2020-04-30 | 2020-04-29 | 1 |
| Landcover-771547 | 4, 8, 1, 7, 5 | 2017-07-31 | 2017-08-02 | 2 |
| Landcover-772421 | 6, 5, 2, 3, 8 | 2021-08-25 | 2021-08-25 | 0 |
| Landcover-772423 | 3, 2, 8, 5, 4 | 2018-02-08 | 2018-02-06 | 2 |
| Landcover-777404 | 4, 3, 2, 8, 5 | 2019-10-04 | 2019-10-03 | 1 |
| Landcover-777410 | 8, 2, 3, 4, 5 | 2019-05-05 | 2019-05-07 | 2 |
| Landcover-1864782 | 7, 2, 4, 3, 8 | 2020-06-15 | 2020-06-17 | 2 |
| Landcover-1865521 | 4, 5, 1, 7, 8 | 2021-05-29 | 2021-05-30 | 1 |
| UNHCR-7977 | 5, 7, 1, 2, 3 | 2017-03-28 | 2014-10-04 | 906 |
| UNHCR-AFGs000003 | 8, 1, 2, 3, 4 | 2018-05-31 | 2018-05-20 | 11 |
| UNHCR-AFGs003914 | 7, 3, 1, 8, 2 | 2018-04-12 | 2018-04-01 | 11 |
| UNHCR-CODs026828 | 3, 5, 1, 2, 4 | 2017-12-29 | 2017-12-27 | 2 |
| UNHCR-CODs026832 | 4, 2, 1, 7, 5 | 2018-11-17 | 2018-11-09 | 8 |
| UNHCR-NERs009694 | 6, 1, 4, 5, 7 | 2019-12-19 | 2019-12-18 | 1 |
| UNHCR-NERs009697 | 3, 5, 8, 1, 2 | 2018-11-02 | 2018-10-22 | 11 |
| UNHCR-NGAs036019 | 5, 1, 8, 7, 4 | 2018-12-11 | 2018-12-10 | 1 |
| UNHCR-NGAs036021 | 3, 1, 6, 7, 4 | 2020-12-08 | 2020-12-06 | 2 |
| UNHCR-TGOs003383 | 5, 2, 8, 4, 1 | 2018-04-27 | 2018-04-27 | 0 |
| UNHCR-THAs001547 | 5, 4, 8, 1, 2 | 2020-11-28 | 2020-12-06 | 8 |
| **Average days difference without considering the outlier** | | | | **2.9** |
| **Average days difference considering the outlier** | | | | **32.03** |

Table 6: The subset of 5 images that approaches the most to the given date of acquisition of our Ground Truth samples.