POLITECNICO DI TORINO
MSc in Data Science for Egineering

Master's Thesis

EagleAI: Estimation of Attitude Geo-localizing Landmarks on Earth

AI-enabled attitude estimation for the nanoSatellite OPS-SAT

Supervisor
prof. Paolo Garza

Candidate
Nelly Gaillard
matricola: 282613

Internship Tutor
ESA - European Space Agency
Dr.ssa Evridiki Ntagiou

Academic Year 2022-2023

Volo d'aquila
Tra Terra e Spazio. Fiera,

Seguii le stelle.

Abstract

This Master's Thesis presents a visual method based on machine learning to estimate, from a picture of the Earth, the attitude of a satellite at the moment it captured that image. Attitude estimation is a crucial task in satellite operations as it determines the orientation and position of the satellite with respect to its surroundings. Conventional methods for estimating satellite attitude require multiple sensors and complex algorithms, making them prone to errors and limitations, especially in the case of small and low-cost satellites as CubeSats. In this work, a machine learning-based approach is presented, to be deployed on-ground, which leverages image data collected by cameras onboard the satellite to geographically localize the landmarks captured, and provide an estimation of the spacecraft attitude. In the proposed method we can identify three main steps: (1) first, the retrieval of a dataset of reference geolocalized pictures; (2) then, the selection of the best candidate pictures for the matching by means of a convolutional Siamese neural network, trained on a large dataset of Sentinel images synthetically modified; (3) finally, a pixel-level keypoint matching that enables the overlap of the input images and the geo-localization of the query. Results from the experiments demonstrate the feasibility of the proposed method and an in-depth study of the literature allows to point-out possible further developments to enhance its accuracy and robustness.

Acknowledgements

This thesis is the conclusion of a challenging path of five years, that lies in my heart with happy and meaningful memories thanks to the presence of all the people I had beside me, without whom I could not have reached this point.

Before anyone else, I want to thank my mum and dad, that constantly believe in me and gift me with endless love, which is a push for every step I make. Thank you for teaching me the value of hard work, that every difficult moment brings something good, and for letting me be completely free to design my path. I thank also my aunts and uncle Elena, Enrica, Eva, Cele and André, and my cousins Giorgia, Luca, Marco, Gigi and Alessandro, that are ready to receive me with open arms at any moment.

I would like to thank my professor Paolo Garza, who has always been supportive and ready to help his students whenever possible; and my supervisor at ESA Evridiki Ntagiou, who gave me the chance to work on the most exciting project so far, and who is for me a source of inspiration as a great and successful woman.

From the bottom of my heart, a huge thanks goes to all my dear friends:

To Cece, you are a friend of the rarest kind, always ready to support me with your precious presence, that comes along with endless laugh, no matter what, whether in Darmstadt, on the phone, in the middle of the night, or with your influence on the Universe. To Ilaria and Sara, you are my oldest friends with whom my soul feels at home.

To Emma, you have been the driving force of my years in PoliTo, and the smartest twin sister I always wished for. I learnt more from you than from many professors, from your brightness and from the tireless dedication with which you face your life; I hope to have hundred more experiences ahead

to share with you, possibly decorated with fancy aperispritz. To Stefano, Gioele, Riccardo, Jacopo, Andrea and Dario, you were the only reason I wanted to be present at the university for every lesson.

To Bea and Berry, your friendship, born under weird and lucky circumstances, makes me forget the emptiness created by the pandemic and fills those years with memories of belly-laughs and holiday dreams, that I am sure one day will become true.

To Bea, Carlotta, Flavia, and Cece, more than just friends and more than just flatmates, you made me feel the luckiest out-of-door student ever. Living with all of you was like living in a musical: there was always the perfect beat to sing away every trouble.

To Gaia, Martina, Silvia, Ivan, Laura, Ale Collet e Sophie, since that October night at the *Panche*, you filled my free time with a climax of wonderful, crazy, and funny adventures I will never forget.

To the AGSA lab dream team, made of Isabella, Eliott, Lorenzo, Lukas, Nuno, and Ryan, and to all the amazing people I had the chance to meet at ESA. Thanks to you I consider the six months I spent in Darmstadt as the best ones of my life, for the constant enthusiasm you shared, and the warm and familiar atmosphere you created. Meeting all of you made me a better person.

Last but not least, I want to truly thank all the other people that I cannot explicitly mention here, but that supported me in many ways during these years, with a hug, a smile, some nice words or with inspiring stories.

Contents

Chapter 1

Introduction

CubeSats, also referred to as nanoSatellites, are miniature satellites that usually have a standard dimension of 10cm cubes (less than a shoe box), which gives them a fraction of the mass and cost of traditional satellites (with the size of a bus to give an idea). Initially designed as teaching aids, CubeSats are now being actively used in orbit for technological demonstration, research, and always more for commercial objectives too. Their reduced cost of production and launch, made space accessible not only to big companies but to private businesses as well, which in turn sparked an ever-growing number of space tech startups with a focus on Earth-related applications, such as satellite communications and imagery, Earth monitoring, and geospatial analytics. Deloitte's Spacetech Report [11] writes that these startups are more and more the object of investors, who are betting on their capabilities to lead the future innovation on Earth, from almost real-time pictures of climate change effects to the high-speed internet needs of any industry involved in the IoT. This trend is referred to as New Space Era, which is attracting more and more the investors attention and is characterised by rapid and dynamic innovation. In support of this, the Space Report 2022 [14] shows that the global space economy reached $469 billion worth in 2021, with a revenue increase of 6.4% since 2020, while the Spacetech one points out that it is estimated to surge to over $1tn by 2040.

The multiple sensors mounted on the satellites enable extensive gathering of data that can be analysed exploiting the power of deep learning algorithms. The possible applications in this sense are endless, but among these an important role is played by the fields of robotics and autonomous systems and

of the so-called "AI at the edge", which consists of the design of optimized models that can run onboard CubeSats and other small satellites, where the hardware resources are limited, especially in terms of storage and computational power. Numerous startups are indeed betting on the key role that AI will have in the Space sector. For instance, AIKO, a deep tech software company born in 2019 at the I3P of Politecnico di Torino, develops pioneering AI technology for the automation of space missions and their operation. The European Space Agency (ESA), as one of the largest space agencies in the world, is determined to keep its leading role in the design, deployment, and operation of missions. To do so, under ESAs Basic Activities, several studies have investigated the use of artificial intelligence for space applications and spacecraft operations. In particular, a strong focus is given to the development of software, concepts, and protocols to push to its limits ESAs CubeSat OPS-SAT.

OPS-SAT is the world's first satellite mission entirely dedicated to testing satellite control technology in orbit, meant to drive innovation and experimentation in this domain [1]. It is a CubeSat, hosting a large number of high-performance payloads, such as an Altera Cyclone V System-on-Chip processor, a high-resolution camera and a software-defined radio. Companies from all around Europe are now able to submit experiments involving OPS-SAT to the European Space Operations Centre (ESOC) in Germany, which will be executed at the Special Mission Infrastructure Lab Environment (SMILE). The project "EagleAI: Estimation of Attitude Geo-localizing Landmarks on Earth", the object of this thesis and registered as OPS-SAT Experiment, emerged in this context.

1.1 Thesis outline

This thesis is the result of a work of six months, carried out at ESOC, the European Space Operation Center of Darmstadt, to support with novel, AI-enabled approaches the operation of OPS-SAT CubeSat, with a view to the development of highly autonomous space missions. In particular, the project aimed to design and build a method to estimate the attitude, i.e. the orientation in space, of the spacecraft OPS-SAT, determining its pointing on Earth employing a visual-based system. Results from the experiments demonstrate the feasibility of the proposed method, showing how it can effectively provide an accurate estimation of the attitude starting from a picture, its timestamp and the satellites orbit.

In Chapter II, we introduce the discipline of attitude determination, with an overview of the traditional approaches and a summary of the mathematical background required for a basic understanding of this subject. In addition, we mention some examples of projects presenting visual-based and AI-enabled attitude estimation methods, followed by the description of the solution proposed in this thesis.

Being the solution scheme composed of a series of consecutive steps, a different chapter is devoted to each of them: starting with a formal definition of the task that is going to be addressed, as well as an outline of its specific challenges and characteristics related to the particular application considered. They then proceed with a literature review of the state-of-the-art and an in-depth description of the approach adopted in the proof of concept. These building blocks consist of the search for geo-localized images to use as a reference, the retrieval of the candidate images in the database most similar to the sensed picture, and the finer keypoint matching that enables the geo-localization of the query image. The former step is the subject of Chapter III, which includes an overview of Earth Observation (EO) satellites and data providers. Chapter IV focuses instead on the task of Image Retrieval, with a thorough definition of the Siamese network implemented and its training procedure. Finally, Chapter V explains the approach adopted to perform the overlapping of the selected reference image with the one sensed by OPS-SAT, commonly referred to as Image Registration.

The core of this work resides in the proof of concept, implemented as a pipeline of sequential modules, which is exposed in Chapter VI with a curated explanation of each step of the process. The main goal was the development of a consistent and functional offline attitude estimator, flexible enough to become a solid starting point for future research. Its modular design allows an easy modification and tuning of each element of the framework, as suggested in the conclusions.

Chapter 2

Autonomous Attitude Estimation

Attitude estimation is a crucial task in satellite operations as it determines the orientation and pointing of the satellite with respect to its surroundings. In order to enable novel autonomous operations of satellites, and in particular of CubeSats, the original idea was to develop a method for the autonomous determination of the attitude of the spacecraft starting from the images captured by the on-board camera. This would allow to obtain an estimation of the attitude alternative to the one provided by the cADCS (coarse Attitude Determination and Control System) by exploiting a cheaper high-resolution camera. After a preliminary examination of the problem, potential solutions, and hardware limits on the target machine, it has been decided to relax some constraints and concentrate on the development of a practical solution running on-ground, with more computing and storage capabilities. Therefore, the problem can be reformulated as follows: given a raw query image taken from OPS-SAT, the goal is to automatically determine the attitude of the spacecraft at the moment the picture has been taken.

2.1 Attitude determination

The discipline of spacecraft attitude determination has the objective of computing the space orientation of the satellite, based on sensors that can give measurements of known quantities in the form of 3D vectors.

2.1.1 Traditional approaches

Conventional techniques for estimating satellite attitude require the calibration, weighting and filtering of multiple sensor inputs and the bespoke tuning of complex estimation algorithms. The basic approach makes use of various sensors (such as sun sensor, gyroscope, geomagnetic sensor, magnetometers, horizon sensor, etc.) to record the 3D variation of a satellite rotation angle. As a second step, the recorded rotation angles are filtered by a Kalman filter (or variants), that uses a recursive algorithm to estimate the state variables of the system based on its mathematical model and on the sensors measurements.

Another method consists of the estimation of the spacecraft orientation based on preexisting simulation databases. In this case, attitude parameters can be solved by using optimal search algorithms.

Furthermore, it is possible to use the Start Tracker (SST), which is a very precise device (it can determine the attitude with an error smaller than 0.1 deg) but it is not always used for its size and weight. Start trackers are based on image processing; they acquire images of the star field observable at visible and near-infrared wavelengths and they identify geometric patterns of potential stars in the captured images. They then compare the geometric patterns identified with a pre-stored star catalogue and compute the satellite's attitude in relation to the stars actually observed.

For what concerns OPS-SAT, it is equipped with two Attitude Determination and Control Systems (ADCS):

- **Coarse ADCS**: it is provided as part of the bus, and it includes control algorithms implemented in the nanomind on-board computer. It relies on magnetotorquers as actuators and on sun sensors and magnetometers as sensors.
- **Fine-pointing ADCS**: also riferred to as integrated ADCS, it is part of the payload and it provides a set of high performance sensors and actuators (i.e. ST-200 Star Tracker and miniature reaction wheels) that allows to obtain an accuracy below 1 deg. Moreover, it enable both nadir pointing and target pointing, which are operations usually only available on larger spacecrafts.

While the coarse ADCS is always operational, the iACDS is only used when a precise information on the attitude is required and whenever there is the need

to point the camera. Therefore, depending on the timestamp, the telemetries might bring information about only one of the two systems.

2.1.2 Mathematical background

To ease the understanding of the software used to compute the attitude and of the problem itself, this section introduces some key notions about the mathematics behind 3D rotations in space.

Recalling the definition of attitude determination, it consists of the study of methods for estimating the proper orthogonal matrix that transforms vectors from a reference frame fixed in space to a frame fixed to the spacecraft body.

Reference coordinate frame

Especially in the context of satellite attitude estimation, where multiple coordinate frames can be used, it is very important to understand their difference and properly defined the one in use.

The reference frames used more often are:

- **Spacecraft body frame**: it is defined by an origin at a specified point in the spacecraft body and 3 Cartesian axes. According to this frame, the camera is pointing towards the -Z axis. It is used to describe the different magnitudes recorded by the satellite.
- **Earth Centered Inertial (ECI)**: the ECI's center coincides with the Earth's center, and it is fixed with respect to the stars. In fact, its x-axis points towards the first star of Aries, the z-axis is aligned with the Earth's rotation axis and the y-axis follows the right-hand rule. This reference frame can be considered inertial, thus it is used to express the orbital motions of objects in Space as well as to specify the direction of celestial objects (such as the Sun).
- **Earth-Centered / Earth-Fixed frame (ECEF)**: the ECEF's center coincides with the Earth's center, and its axes follow the Earth in its rotation movement. The x-axis is contained in the equatorial plane and points to Greenwich meridian, the z-axis is aligned with the Earth's rotation axis and the y-axis is defined by the right-hand rule. This reference frame is not inertial, it is accelerated and it rotates with respect to stars. It is used to express the motion of objects on the Earth's surface and to describe the Earth's magnetic field.

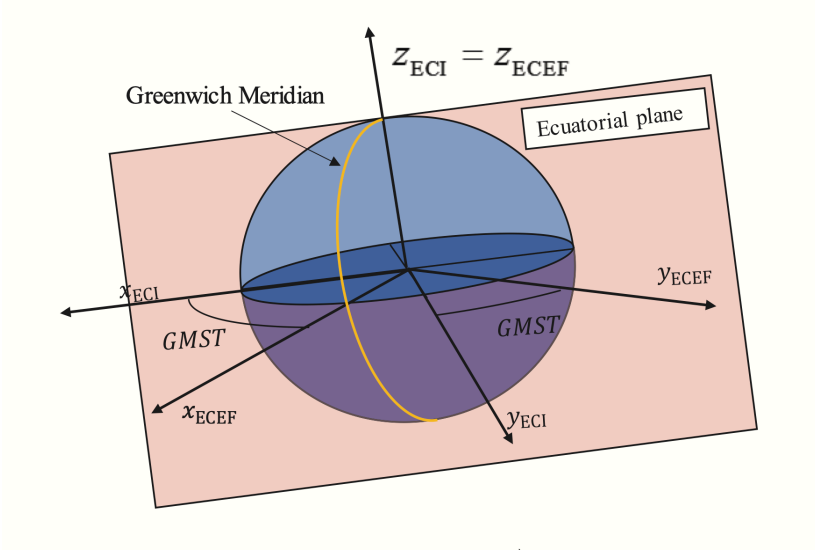


Figure 2.1: Relation between ECI and ECEF frames. [24]

3D rotations

The parametrization of a 3D rotation is not straightforward, as several different possibilities exists, and many of them can be used at the same time according to the information to convey. In the context of robotics as well as spacecraft attitude determination, the two most used representations are:

- **Axis/Angle Representation**: a rotation can be represented by a rotation axis $\hat{n}$ and an angle θ, as shown in fig 2.2. This transformation can be described by a rotation matrix written as a function of $\hat{n}$ and θ. It represents the orientation of a rigid body with respect to an inertial coordinate system describing 3 successive transformations around the body fixed axis. It is the best option in the case of small rotations, but in general it is not a unique representation, since it is always possible to add a multiple of 2π radians to θ and get the same rotation matrix.

- **Quaternion Representation**: also a quaternion is defined by a rotational axis $\hat{n}$ and a rotation angle θ, and it consists of a unit length 4-vector whose components can be written as $\mathbf{q} = (q_x, q_y, q_z, q_w)$. Unit quaternions live on the unit sphere $||\mathbf{q}|| = 1$ and *antipodal* quaternions, **q** and **-q**, represent the same rotation. Other than this duality, the unit quaternion representation of a rotation is unique. Moreover, this representation is *continuous*. The quaternion representation expresses

the attitude matrix as a homogeneous quadratic function of the quaternion's elements, requiring no trigonometric or trascendental function evaluation.

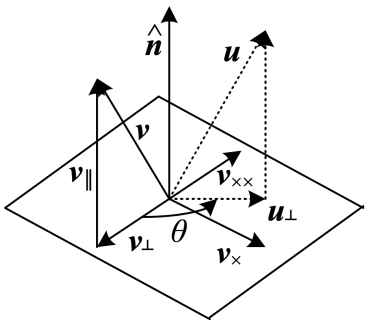


Figure 2.2: Axis/Angle representation of a rotation around an axis $\hat{n}$ by an angle θ. [30]

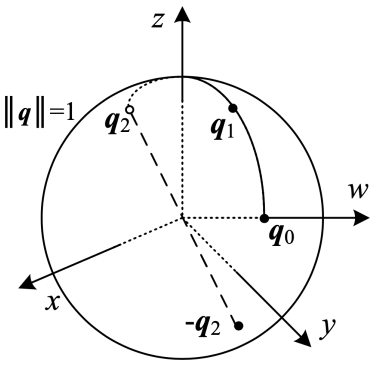


Figure 2.3: Unit quaternions live on the unit sphere $||\mathbf{q}|| = 1$. This figure shows a smooth trajectory through the three quaternions **q0**, **q1**, and **q2**. [30]

2.2 Related works

The geo-localization of images has been widely studied and several solutions can be found, exploiting diverse AI approaches. On the other hand, there are few evidences of its practical application to provide an autonomous estimation of a spacecraft attitude. In the following paragraph we are going

to present a couple of use cases that have already addressed this problem by means of visual-based Artificial Intelligence techniques.

Both examples deal with onboard applications. The first one, referred to as the DLAS project, makes use of a deep learning technique for semantic segmentation. It assumes that the satellite is equipped with a GPS sensor providing the precise location of the spacecraft, which is not available on OPS-SAT. The DAT project, instead, proposes a solution embedding one of the latest deep learning algorithms for edge devices, but its functioning is assessed only on a limited scenario.

2.2.1 DLAS project

Deep Learning Attitude Sensor (DLAS) [16] is a project that proposes a method to estimate the attitude of Small Satellites by means of machine learning techniques applied to image data. It represents one of the first examples of the implementation of an attitude sensor enabled by image-based machine learning algorithms.

It has been developed by the Tokyo Institute of Technology in 2018 and tested a couple of years later with the JAXA's program of innovative satellite technology demonstration. The aim of this work was to perform attitude estimation using color images taken with a low-cost COTS visible light camera. The algorithm was developed t be run on the onboard computer, equivalent to a RaspberryPi 3 Model B. The main idea was to compare the Earth surface patterns on the pictures with map data preloaded onboard, reducing the search space with the help of the GPS position. The approach used in this case to determine the 3-axis attitude of the spacecraft is based on the following steps:

1. *Semantic segmentation*: the input image is fed into a light-weight Multi Layer Perceptron network that categorizes each pixel into 10 classes, by means of a sliding-window approach.

2. *Reference database creation*: the system generates imaginary pictures with all possible angles around the nadir vector, with the help of a dataset previously stored onboard.

3. *Similarity search*: it then computes the similarity of the obtained segmented picture with the generated catalogue images by using a template matching technique.

4. Finally, it obtains the angle with the highest similarity values among all the generated catalog images.

The picture 2.4 provides a clear visualization of the approach just described.

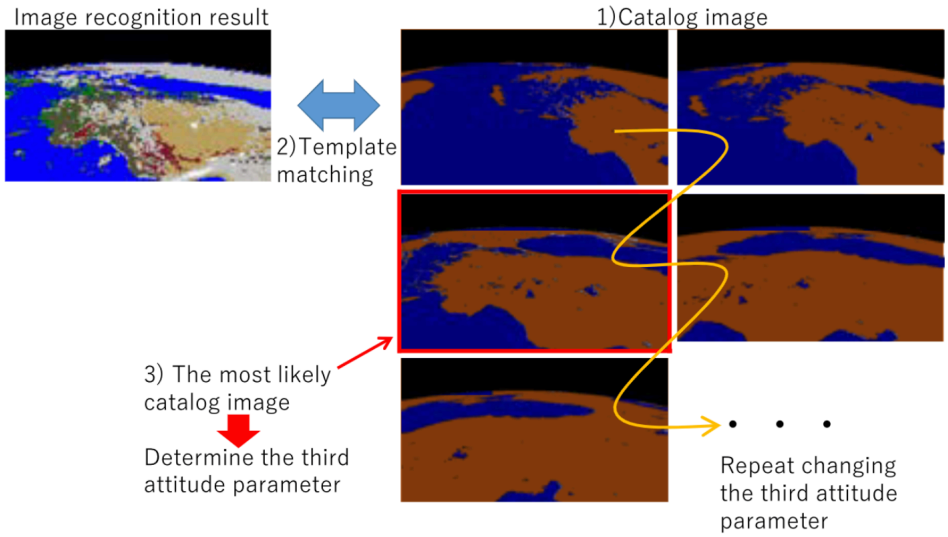


Figure 2.4: Deep Learning Attitude Sensor approach

The experimental results reported in 2021 show how the systems was able to detect the 3-axis attitude under certain conditions (as a low cloud coverage), and that the accuracy was comparable to the one of a coarse sun sensor ($\tilde{1}$ deg).

2.2.2 Deep Active Tracking project

The Deep Active Tracking (DAT) project is an AI-based system for an active tracking of Earth features from OPS-SAT pictures, developed by the company *Adatica Engineering* and submitted to OPS-SAT as an Experimenter. The DAT's goal is to demonstrate the operation of an AI-enabled attitude control system that can detect a landmark on Earth with the on-board optical camera and maintain it framed and focused on the target by controlling the reaction wheels to actively control the attitude of the probe.

The system consists of two main parts: initially, a computer vision algorithm process the pictures taken by OPS-SAT to detect the presence of a target

landmark and gives as output the corresponding coordinates; then, a second algorithm based on Deep Reinforcement Learning determines the best sequence of actions to position and maintain the target centered on the field of view of the OPS-SAT optical camera.

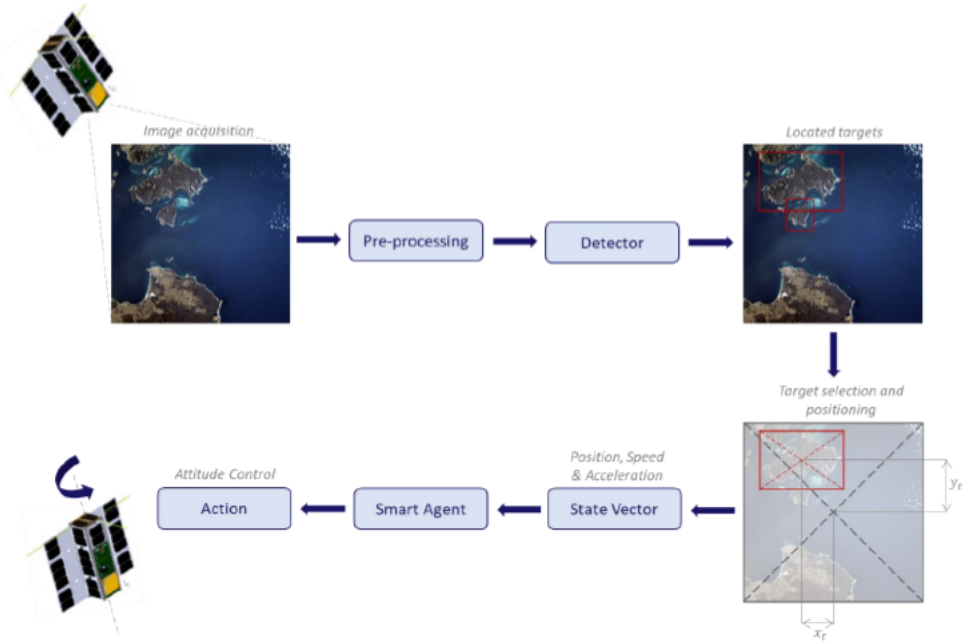


Figure 2.5: DAT scheme of target landmark detection and attitude control process

The first task is the most relevant for us, since it aims to solve the problem of geolocalizing landmarks on Earth. In this case, it is addressed by means of a Deep Neural Network trained for object detection with a supervised learning setup. Specifically, due to the constraints in terms of speed and accuracy in real-time applications, the selected neural network is the YOLO architecture. YOLO is a Convolutional Neural Network (CNN) that detects objects as a regression problem and provides, at the same time, class probabilities of the detected landmarks as well as their location by means of a bounding box set of coordinates.

The network has been designed to be trained with a supervised learning technique, which means that the training dataset must include both the target landmarks and the corresponding geographical annotations that are used as ground truth. For the purpose of this Proof of Concept the authors selected the islands as the target landmark. Limiting the application to

islands allows in fact to provide an efficient end-to-end system, while leaving it open to generalizations, i.e. allowing it to be trained to track any other natural or man-made feature. Since the number of images taken from OPS-SAT are in a limited number, as an alternative data source for training the algorithm they used the Sentinel-2 Open database.

2.3 Proposed solution

As previously mentioned, the problem addressed by this project is to autonomously retrieve an estimation of the spacecraft attitude starting only from the pictures captured by the on-board HD optical camera. Having considered the examples previously described, this work has been set up to provide an alternative solution, which is designed to be run on-ground to avoid the strict constraints in terms of data storage and processing power. The adaptation of this work for an onboard use is let as a future development.

Given the context of deployment, it is legitimate to assume the position of the spacecraft along the orbit to be known, since we can retrieve it from the intersection of the picture timestamp and the TLE[1] (Two-Line Element set) of the satellite. Therefore, the challenge translates in the detection of landmarks, i.e. geographical objects, in the image and in their geo-localization. Once we can find the coordinates of at least three pixels of the picture, the attitude can be retrieved by solving a geometrical problem. We can identify three main steps:

1. **Retrieval of reference images**: the first requirement for localizing the landmark is to find the corresponding reference image equipped with geographic information. Considering the whole World land as initial search space is unfeasible: the database of images would be too big to be stored, and the search would not be efficient. At the same time, restricting the research to a predefined subset of landmarks, as was done for the DAT project with the selection of islands, creates a big limitation in the usefulness of the application. For these reasons, we chose not to consider an on-board use, which would add restrictions in terms of memory and computational resources, but to relax the constraints and focus on the

[1]The TLE is a data format encoding a list of orbital elements of an Earth-orbiting object for a given point in time. In this case, it uniquely describes the state of the satellite at a given epoch.

development of a method running on-ground. Given these assumptions, it is possible to retrieve the reference pictures from open source satellite images providers. After considering and comparing several options, as discussed in the next chapter, the Copernicus Open Access Hub has been chosen to retrieve Sentinel-2 pictures. Since the spacecraft's location on the ground track at the time of the photo's capture is known, the idea is to draw an appropriate Area of Interest around that point which covers the satellite's field of view. Once the AoI is defined, it is possible to download lightweight images from the Copernicus Open Access Hub to perform a first selection of the products and avoid a massive processing of useless data.

2. **Coarse matching step**: the second step of the pipeline is to filter, among the downloaded images covering the AoI, those that actually represent the same scene as the query picture. Since usually the initial number of downloaded pictures is around 300-500, the goal is to select the 5 or 10 candidate reference images that most likely contain the landmark captured by OPS-SAT, to increase the efficiency of keypoint matching. The approach we adopted consists of a Siamese Neural Network, trained to automatically detect similarities between two images. It takes as input a pair of pictures, obtains their feature representations in the latent space and gives as output the Euclidean distance between the two feature vectors.

3. **Fine keypoint matching**: once good candidates for the matching are obtained, we apply a keypoint matching algorithm, in order to precisely link at a pixel level the landmarks in the two input pictures. In general, the main goal of a keypoint matching algorithm is to detect the points of an image that carry the most information and describe them through a feature vector. This enables the comparison of feature vectors between two different images and the matching of the points that are more similar to each other. Also in this case, several solutions have been studied and compared. The more straightforward method is SIFT, which is a traditional and widely used computer vision algorithm. Thanks to its robustness it has been chosen as the best candidate to provide an initial proof of concept for this project. On top of the keypoint matching algorithm, we use the RANSAC algorithm to refine the results, which divides the keypoints set in inliers and outliers and estimates the geometrical transformation that links the two images.

A graphical representation of the entire pipeline is reported in fig 2.6.

From the inliers set of keypoints we then estimate the geometrical transformation that allows to overlap the image we want to geolocalize with the reference one. The goodness of the image registration is evaluated by means of a visual check or by computing the Root Mean Squared Error of the transformed keypoints. In this way, we can easily obtain the geographical coordinates of each pixel. To extract the attitude, at this point it is enough to select three points of the picture and feed them to a python script already implemented by the OPS-SAT team, that performs the geometric computations to extract the attitude of the spacecraft.

For the validation of the results it is possible to compare the obtained attitude parameters with those of the fine iADCS (Attitude Determination and Control System) of OPS-SAT, which can be retrieved from the telemetry history. Another option is to visually test the transformation of the reference image such that it matches the query, i.e. the image we want to localize.

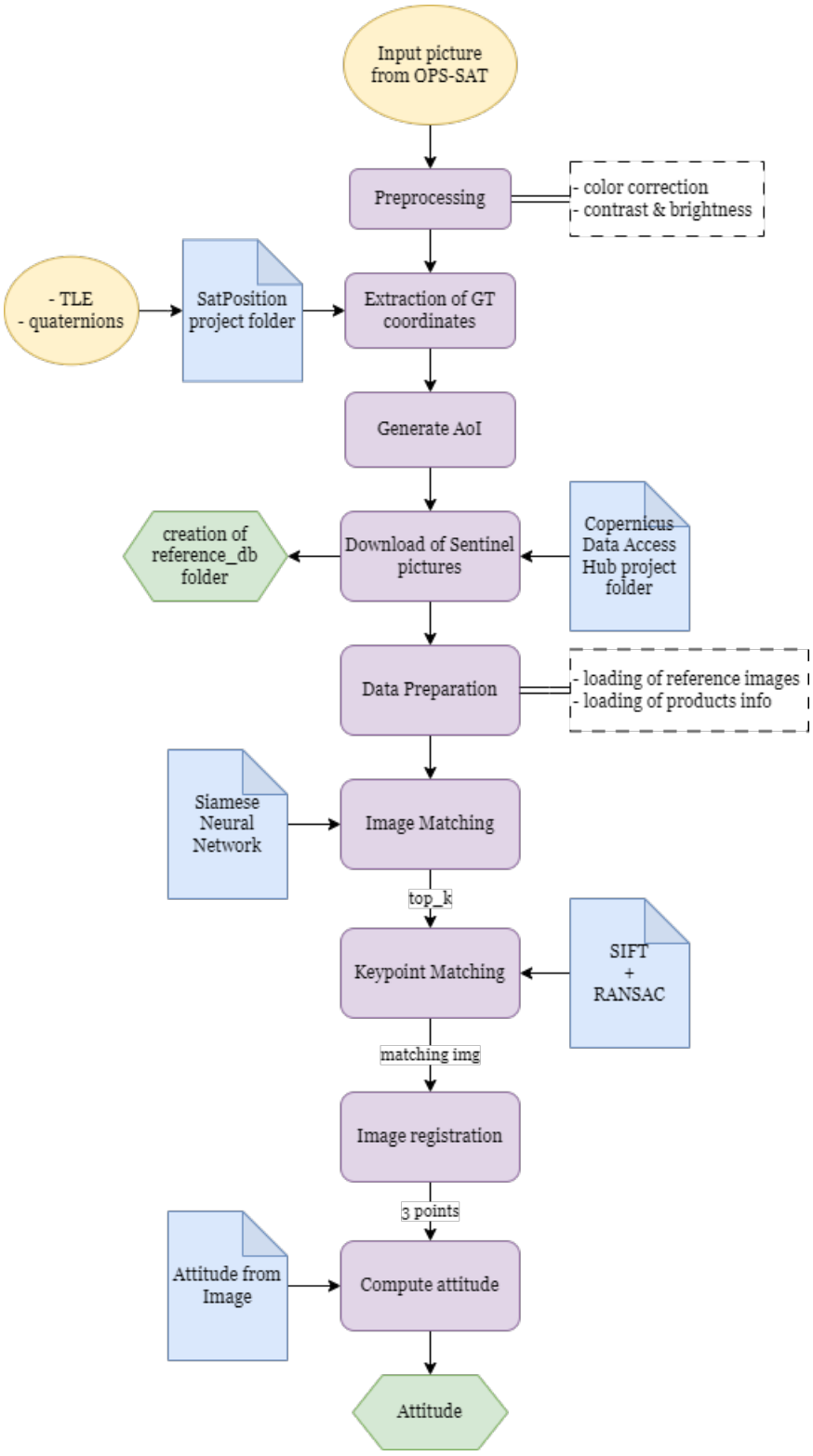


Figure 2.6: Diagram of the project architecture

Chapter 3

Geo-referenced images retrieval

The data collection is the most important and delicate step in the development of a machine learning application, since it will directly affect the outcome of the algorithm. Specifically, our use case requires two different datasets:

- One or more query images, that are the ones we are interested to localize
- A dataset of geo-referenced images that we use as basis for the registration

The query image is provided with a unix timestamp, which allows us to retrieve the position of OPS-SAT in the orbit from the historical telemetries database and its associated point on the ground. Starting with this information, in order to geo-localize the picture, we need a set of reference images covering the whole field of view of the satellite, which can have a diameter of several hundreds of kilometers, given that the spacecraft lies at 600km orbit height.

In the case of satellite imagery, there are many potential data sources available (such as Sentinel-2 of ESA, or Landsat-7 and Landsat-8 of NASA), since the number of satellites orbiting the Earth is increasing day by day, but each of them differs in resolution, level of processing, cost and licensing, making the choice of the right data not straightforward.

Therefore, the first step consists of the choice of the best source of geospatial

Table 3.1: Technical details of the OPS-SAT onboard camera

Parameter	**Value**	**Comment**
Spatial Resolution	53 m	@ 600km orbit height
Field of View	135x105 km	@ 600km orbit height
Channels	3	RGB via on sensor Bayer Pattern
Frame rate	7 / 15	2600x2000 pixels / 1300x1000 pixels

data and of the setup of a pipeline that, given the timestamp of query image, can retrieve the geographical position of OPS-SAT. Then, it downloads the reference pictures that cover the desired Area of Interest (corresponding to the satellite Field of View) around that point.

Since the target of our application is OPS-SAT satellite, or more in general small satellites, the reference dataset is required to have similar characteristics with respect to the query. From the specifics of the onboard camera reported in the table 3.1, we can notice that the resolution is lower with respect to larger Earth Observation satellites. However, the one mentioned is not the only parameter to consider when comparing pictures from different sources. Other important parameters that characterise geospatial data are:

- *Spatial resolution*: also referred to as ground sample distance GSD, it indicates the size of the ground tile covered by one pixel.
- *Temporal resolution*: frequency with which data is collected over the same region (also called 'revisit time')
- *Spectral resolution*: the number of spectral bands and width of each spectral band (e.g. multi-spectral sensors have 3-10 bands, OPS-SAT captures RGB bands).

In addition to the above mentioned parameters, for the creation of an efficient reference database, the images should be available real-time and they should be light-weight, allowing to reduce as much as possible the size of the downloaded data.

In this chapter, guided by the above mentioned requirements, we first present an overview of satellite imagery ecosystem, then we expose the research done on the possible data providers, analysing the different platforms and comparing the characteristics of the infrastructures.

3.1 EO Satellites and Data Access Platforms

Geo-referenced data consists of data structures, referred to as *raster*, combining the remote sensing information (from a camera or another kind of sensor) with its geographic location. They can be collected by satellites, aircraft, or drones.

In recent years, thanks to technological advancements, the amount of geospatial data collected by Earth Observation (EO) satellites has rapidly increased, allowing to assess the status of the natural and manmade environment as well as its changes in time. This kind of data enables the extraction of a wide variety of information useful for monitoring environmental changes, risk detection, and the analysis of urban occupation. Among all the data collected, petabyte-scale archives of remote sensing data have become freely available for society and researchers by open data rules set by governments and space agencies, with the intent to push the development of EO applications. Examples of these resources are the images from Sentinel satellites, maintained and operated by Copernicus, i.e. the Earth Observation component of the European Union space programme [13], the NASA Landsat satellites, and the Moderate Resolution Imaging Spectroradiometer (MODIS) imagery of the U.S. Geological Survey (USGS).

However, the large amount of daily collected data comes with a big challenge for what concerns the access and storage of satellite and aerial remote sensing data (just consider that, only in 2019, the amount of data collected by Landsat-7 and -8, Sentinel-1, -2 and -3 and MODIS is around 5PB). As a consequence, even if satellite images themselves are made freely available, the storage and processing of a large number of data for further analysis is anything but easy. To ease the access and management of spatial data, many companies and institutions developed advanced Spatial Data Infrastructures, i.e. platforms that serve as an interface between the user and the data system and that integrate different kinds of technologies, such as Application Programming Interfaces (API) and web services [9].

An example of raw spatio-temporal data is reported in fig 3.1[1], where it is possible to identify the multiple dimensions of the raster data (space, time and bands). As indicated, each image is provided with metadata containing information on the coordinates of each pixel, on the capture time, and in

[1]Image taken from *www.openeo.org*

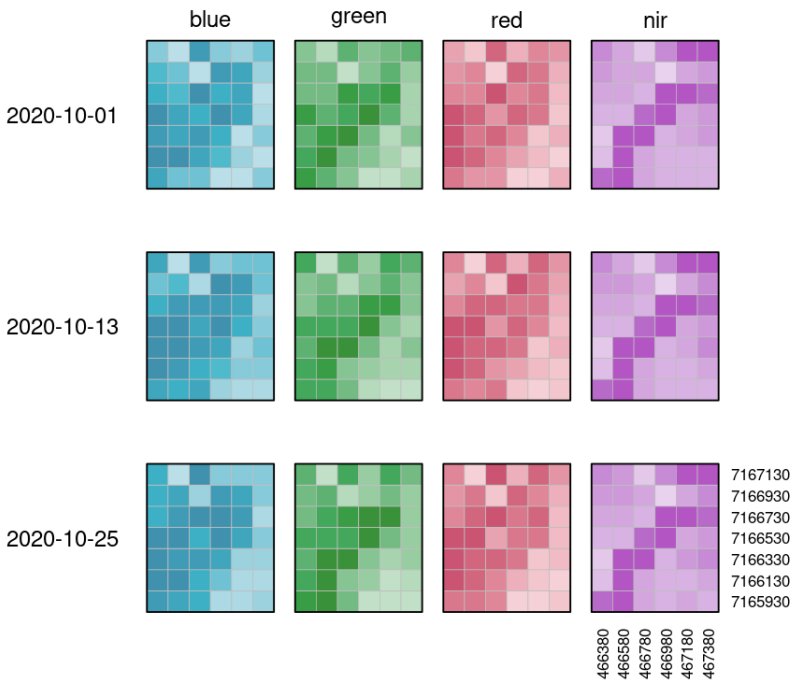


Figure 3.1: Example of datacube

some cases also a cloud mask; the final product therefore often takes up hundreds of MBytes.

The following paragraphs report some of the possible platforms providing satellite imagery.

3.1.1 Sentinel Hub

Sentinel Hub[2] (SH) is a private platform owned, developed, and operated by Sinergise that provide access and visualisation services for satellite data (from Sentinel, Landstat and other providers).

Based on the functionalities SH offers different payment plans, allowing to access for free only the visualization, selection and download of data from

[2] *https://www.sentinel-hub.com*

the EO Browser. Under subscription, many different functionalities are made available, such as higher resource access limits, a user-friendly *Dashboard* for the selection of data, and different APIs shaped upon the users needs (OGC API to directly integrate the data in a GIS application, a RESTful *Process API* that allows to easily process data inside a custom script, or a *Batch Processing API* for the access to larger amount of information).

3.1.2 Google Earth Engine

Google Earth Engine[3] (GEE) is a cloud-based computing platform where users can perform large-scale geospatial analysis and visualisation supported by Google's infrastructure.

GEE was created in 2010 by Google as a proprietary system, ad its services are made free for any non-commercial use and research projects with small and medium workloads. The data catalogue it offers is very wide, including raw satellite images from Sentinel, Landsat, MODIS and more, as well as geospatial datasets equipped with environmental variables, weather and climate forecasts and hindcasts, land cover, topographic and socio-economic data. To ease the access, it provides a JavaScript API, for which there is also a web Integrated Development Environment (IDE), and a Python API for data management and analysis.

The APIs implement a large variety of functions and operators (more than 800 considering both simple mathematical operations and complex machine learning and geostatistical algorithms), that are automatically executed by the parallel processing backed system, which subdivides and distributes computations with by means of a MapReduce approach.

As described in [10], Earth Engine is built on top of the Google's suite of big data technologies, and it enables their interaction with the final application through client libraries that send queries to the system by means of a RESTful API.

As a down side, the use of GEE for geospatial data analysis makes the application dependent from Google technologies and not easily exported. In fact, all computations must be expressed using the Earth Engine library, which means that existing algorithms and workflows have to be converted to utilize

[3] *https://earthengine.google.com*

the platform at all. This impedes the user of extending the functionalities of the library, such as using the processed images as input to custom machine learning models.

3.1.3 openEO

The limitation just mentioned, of platforms like EEO, often represents an entry barrier for EO scientists, that might fear of becoming dependent on the provider of the chosen system. In fact, all these backend providers have adopted their own customer-driven development paradigm, tailored to their specific data infrastructures, making it very hard to switch to other service providers and to compare the results and the performances of different back-ends [28].

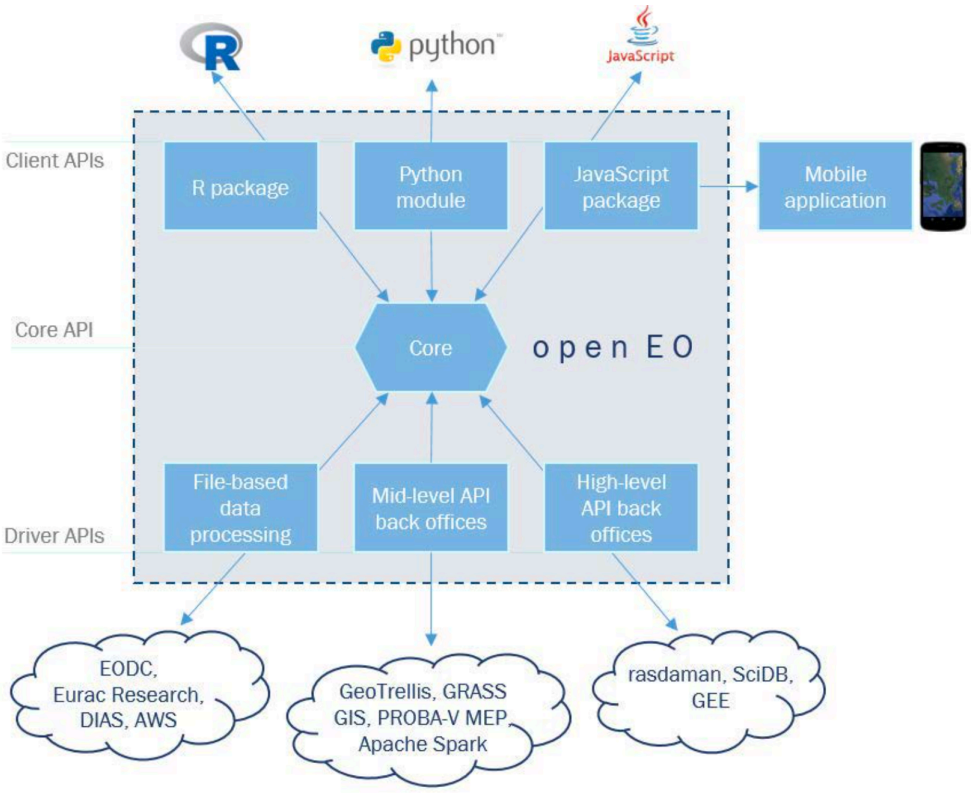


Figure 3.2: OpenEO platform architecture [9]

This issue has been addressed by the open source OpenEO project[4], which

[4] *https://openeo.org*

aims to provide a mechanism for researchers to develop their applications and analyses using a single standard, which hides the architectural differences between cloud platforms [9]. It consists of a three-year funded by the EU under the Horizon 2020 project, and developed by a consortium of companies (EODC (AT), VITO (BE), Sinergise (SI), EURAC (IT), EGI (NL) and University Muenster (DE)). The goal was to design an API composed of three layers: a **Core API** for finding, accessing and processing large datasets, a **Driver API** to connect to backends operated by European and worldwide industry, and a **Client API** for analyzing these datasets using R, Python or JavaScript. The structure of the OpenEO platform is reported in fig 3.2.

3.1.4 Copernicus Open Access Hub

The vast majority of data and information delivered by the Copernicus space infrastructure, which includes the Sentinel satellites fleet and in-situ data gathering as depicted in 3.3, is made freely available and accessible by the European Commission policies to any citizen and any organisation around the world.

The platform that makes this possible is the Copernicus Open Access Hub, a portal that provides complete and open access to Sentinel-1, Sentinel-2, Sentinel-3, and Sentinel-5P user products, with synchronous access to most recent data and asynchronous access to historic data stored in the Long Term Archive (LTA). It offers either a Graphical User Interface, that allows performing thorough queries to the database in a user-friendly way, either two dedicated programming APIs that can handle a large set of client tools and enable the creation of scripts for automatic search and download of data:

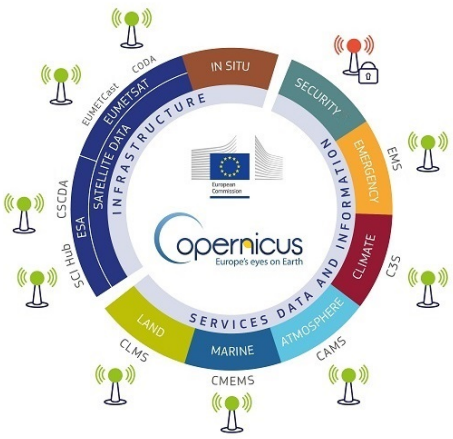


Figure 3.3: Copernicus system

- **OData** is a data access protocol based on fundamental protocols like HTTP and widely used techniques like REST. Therefore, it is easily handled by different client tools and applications, and it can be used to create input URIs to perform batch data download.

- **OpenSearch** is a set of RESTful technologies that can be used to quickly locate the desired resources, which can then be downloaded by using OData.

The ODaata an OpenSearch URIs can be combined to create complex queries to be executed in non-interactive scripts using programs like cURL and Wget.

Additionally, there exists Sentinelsat, a Python library that further simplify the process of searching, downloading and retrieving the metadata form the Copernicus Open Access Hub. Sentinelsat offers a user-friendly command line interface in addition to a python API, that makes it straightforward to integrate the data retrieval into more complex python projects.

The API provides many functionalities to access the Sentinels data products, in order to select among the large amount of data hierarchically organized, only those parts that are actually needed. To give an idea, each Sentinel 2 product refers to a directory folder that contains a collection of information as described in [12]. The main subfolders are:

1. Auxiliary Data Folder (AUX_DATA), that contains the set of auxiliary files that can be embedded in the product (e.g. International Earth Rotation & Reference System bulletin)
2. DATASTRIP, with datastrip level information
3. GRANULE, which includes image data (granules/tiles) in JPEG2000 format as well as quality indicators (quality masky masks, quality reports, etc.). The image data is contained several levels down within the folder. A granule is a 100x100km2 ortho-image in the UTM/WGS84 projection.
4. User product html data folder (HTML), containing a product presentation file allowing to display easily the main content of the product
5. A Representation Information Folder (rep_info) with an XSD schema that describes the product components
6. An INSPIRE.xml file, collecting metadata according to the INSPIRE Metadata regulation
7. Product Level-1C Metadata File in XML format that describes the physical organization and the content of the product
8. The Sentinel-2 Manifest file (MANIFEST.safe), which holds the general

product information in SAFE format, specifically designed as a standard to archive and convey data within ESA Earth Observation archiving facilities. The SAFE format wraps a folder containing image data in a binary data format and product metadata in XML. This flexibility allows the format to be scalable enough to represent all levels of SENTINEL products.

9. A preview image in JPEG2000 format.

A new Copernicus Data Space Environment with additional tools for data processing and visualization has been operational from January 24th 2023, which is going to replace the current services of the Open Access Hub. This new platforms is intended to ensure instant data availability to users, offering real-time availability of the full data archive acquired by the Copernicus Sentinel satellites. It will still provide OData and OpenSearch APIs and a newly designed web browser application.

3.2 Sentinel Satellites

Sentinel is a set of Earth Observation satellites controlled and maintained by ESA and the European Commission under their joint Copernicus initiative. It currently includes six main missions, each of which is equipped with different sensing instruments to provide high quality data that focus on different aspects of Earth Observation, such as Atmospheric, Oceanic, and Land monitoring.

For the aim of this project, Sentinel-2 products has been chosen as the best option for the construction of the reference databases, for the high quality of the data and for their availability through the Copernicus Open Access Hub. A brief description of each mission is provided below:

1. **Sentinel 1**: it is composed by two polar-orbiting satellites performing synthetic aperture Radar imaging and, that enables them to acquire imagery day and night, regardless of the cloud coverage or the illumination. They have a 12 days repeat cycle, which means that the mission is able to map the entire world every 6 days. Sentinel-1A launched 3 April 2014, Sentinel-1B launched 25 April 2016.

2. **Sentinel 2** it is a wide-swath, high-resolution, multi-spectral imaging mission, focused on land monitoring. It provides, as an example, imagery of vegetation, soil and water cover, since each satellite carries an optical

instrument payload that samples 13 spectral bands (with different spatial resolutions). Sentinel 2 has a 290km swath width and a revisit time of 5 days with 2 satellites. The acquired imagery is projected onto a UTM/WGS84 grid and made publicly available on 100x100 km2 tiles. The first Sentinel-2 satellite was launched in June 2015.

3. **Sentinel 3**: its main goal is marine observation, for which it will measure ocean and land surface temperature, ocean and land color, and sea-surface topography. The mission, which consists of three satellites, has a radar altimeter as its main instrument, although the polar-orbiting spacecraft will also carry other experiments, such as optical imagers.

4. **Sentinel 4**: designed specifically to monitor the air quality, Sentinel-4 is equipped with a UVN sensor, which is a spectrometer that allows to gather information about the composition of the Earth's atmosphere.

5. **Sentinel 5P and 5**: is a payload on a MetOp Second Generation satellite that will observe and monitor the atmosphere from a polar orbit. Sentinel 5P was the precursor, used to fill the gap and provide data continuity between the retirement of the Envisat satellite and NASA's Aura mission and the launch of Sentinel-5.

6. **Sentinel 6**: particularly designed for operational oceanography and climate studies, it contains a radar altimeter to to extend the legacy of sea-surface height measurements. The first Sentinel-6 was launched into orbit in November 2020 on SpaceX Falcon 9 rocket.

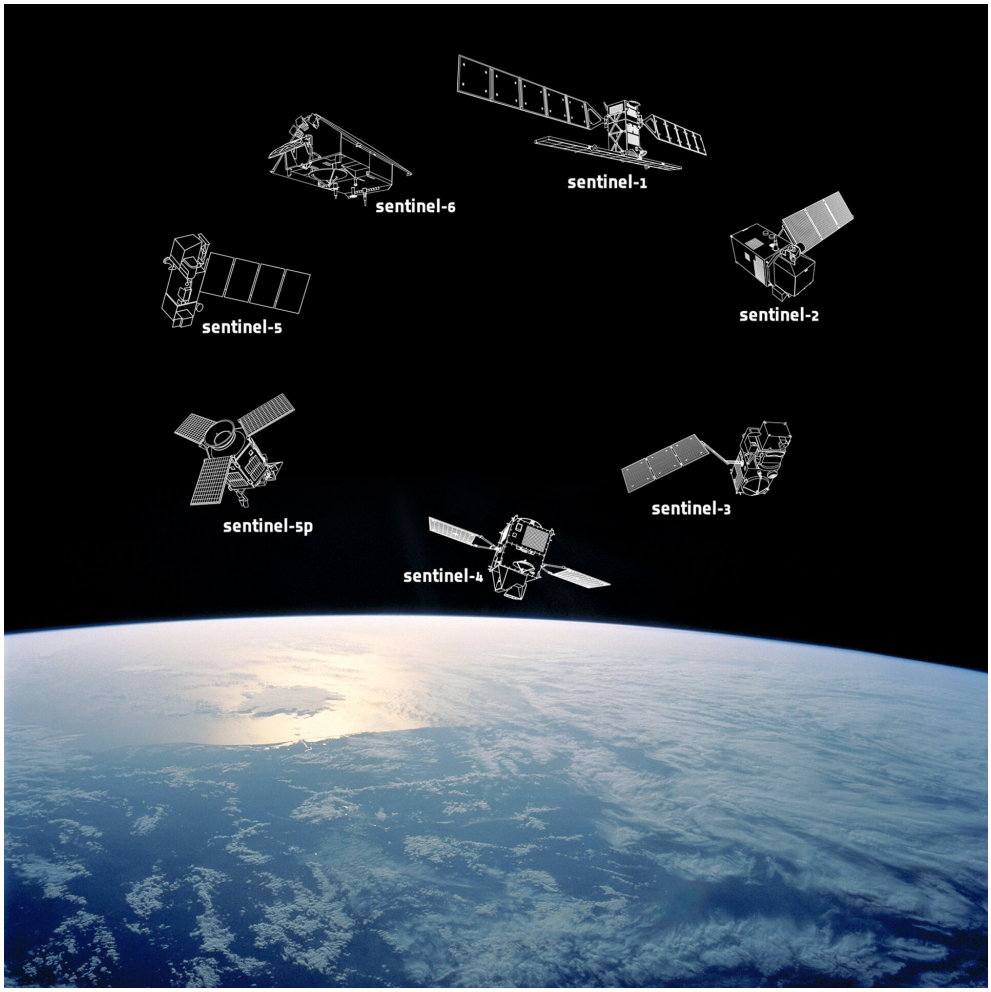


Figure 3.4: Sentinel family satellites

Chapter 4

Image Retrieval

Once we collected the images covering as much as possible the satellite field of view, we need to apply an algorithm that finds, among them, the pictures that contain the same landmarks represented in the query. This step has been created to ease the work of the following, more precise, keypoint matching, by filtering among all possible reference images only those that share at least one landmark. In the machine learning literature, this problem is classified as Image Retrieval: given a query image, the aim to find, among a large database, all the images that contain the same instance, often captured under different conditions, such as different viewing angles, seasons or illumination.

In the case of landmark detection from satellite imagery, the domain shift is a sensitive issue, and one of the main obstacles faced in the project development. First of all, while it is possible to require the absence of clouds when collecting Sentinel images, the query picture coming from OPS-SAT often includes a high percentage of clouds, which can be misleading for the algorithm and create occlusions of some important reference points. Moreover, another challenge is represented by the change of perspective concerning the landmark: Sentinel-2 always operates at the nadir, which means that the camera is pointed perpedicular to the Earth's surface, while OPS-SAT captures the scenes from different points of view, producing a very distorted reproduction of the object captured. Last but not least, the resolution of the image is another problem hard to address. In fact, not only the field of view of the query pictures changes according to the inclination angle of the camera, but, at the same time, the images from Sentinel-2 have higher accuracy than the CubeSat ones.

Therefore, the design of the algorithm needs to take into account the viewpoint invariance for the perspective transformation as well as the appearance invariance, mostly concerning the colors. A robust and solid image matching system should deal effectively with changing light conditions (OPS-SAT's orbit follows the shadowed region between the day and the night, which gives to the pictures a blue hue) and seasonal variations of the land appearance.

This chapter is devoted to the study and implementation of the Image Matching step: first of all it presents an overview of the literature and of the state-of-the-art approaches; in a second moment, it will focus on the adopted solution.

4.1 Related Work

Given that the topic of Image Matching very broad, to drive the study of the literature, at the beginning we define which is the task that most align with the problem at hand. In the research papers there are many similar designations used to describe nuances of the Image Retrieval task or some specific sub-tasks, which make it difficult to visualize a clear map of the available approaches. In general it is possible to place this case in the context of Content-Based Image Retrieval (CBIR), a long-established research area, consisting of the search of semantic matches or similar images in a large database, given a query image. Its essential stages are shown in Fig 4.1.

However, it is also possible to look at the problem at hand in the light of *Visual-Based Localization*, a research direction aiming to retrieve the orientation of the sensor that captured the visual information. In this case, the proposed methods mostly concern robotic applications, such as the Structure-from-Motion task, that takes into account the scene around the robot and tries to map the environment through 2D or 3D data for estimating the pose of the camera with 6 degrees-of-freedom. Since the input data is very different from satellite imagery, most Visual-Based Localization solutions cannot be applied in this case: the main difference is that the context captured from the spacecraft is too far to appreciate features that indicate the relative position of the camera. Another approach, that also applies in this context, is the one of *Landmark Retrieval*, which is a particular case of Instance Retrieval concerning only landmarks, i.e. relevant distinguishable objects. The strategies adopted usually consider supervised approaches, where the training dataset is composed of labeled samples, with a discrete number of

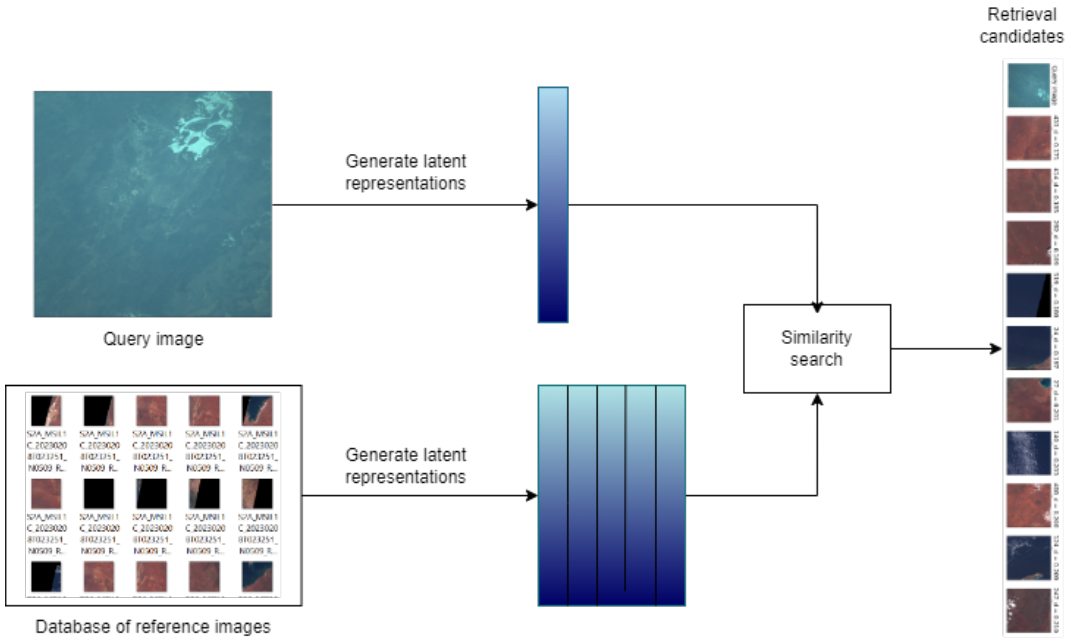


Figure 4.1: Architecture of an Image Matching system

distinct landmarks. As an example, this category of task is similar to the solution proposed for the DAT project: the network is trained on a labeled dataset, with the goal of recognizing a predefined set of reference objects. The downsides of this strategy are indeed the fact that the landmarks considered have to be selected a priori, and that there is often the need for a labeled dataset, the construction of which is time consuming, especially in the case of OPS-SAT where the availability of high-quality images is very low. Furthermore, the rise of available satellite imagery, that is continuously accumularing Peta/Zetta Bytes of data, created the need of a line of research specifically dedicated to remote sensing data, that largely differs from natural images in terms of modality, spectral and resolution [19]. In fact, due of its wide applications, earth observation data is exploited more and more, thus making it urgent to develop efficient image retrieval methods to select the images of interest from the massive RS image repositories.

Being aware of the existing nuances in the research, from now on we will consider the general field of CBIR, with a special attention on RS applications. CBIR systems usually include three core modules:

1. **Feature Detection**: it includes all the methods used to first localize the geometric structures in the image and then describe those features

by transforming the original local information around the interest point into a stable and discriminative form.

2. **Feature Representation**: this step is often included in the feature extraction algorithms, and it consists of the compact combination of the features extracted for each input, with the goal of reducing the dimensionality of the data while increasing their discriminative abilities. Embedding and Aggregation

3. **Similarity search**: it includes the algorithms that can perform a similarity search in the descriptors space between the feature vector of the query image and those of the reference database. Usually, the similarity is obtained by computing the Euclidean or cosine distance among pairs of features vectors. The most popular option is to use the k-Nearest Neighbours (kNN) algortihm.

Before the rise of deep learning, these steps were independent, while nowadays many methods integrate feature detection in the entire matching pipeline, by jointly training with feature description and matching.

In fact, one of the possible classification criteria of CBIR methods is to consider the methodology perspective: on one side there is the set of *traditional approaches*, performing feature engineering by means of hand-crafted descriptors and classic shallow machine learning algorithms; on the other side, more recent *data-drive deep learning* techniques are rapidly evolving since their first breakthrough in 2012 with AlexNet [17] winning the ImageNet competition.

Moreover, the information extracted from a picture can be divided in levels based on the spatial resolution of the represented feature. To each of these levels it is possible to associate a set of techniques, which in many taxonomies correspond to *local* or *global* descriptors.

4.1.1 Local descritptors

The methods falling in this category are those responsible for the extraction of *low-level features*, and typically correspond to traditional hand-crefted descriptors. Low-level features include for instance the texture, i.e. repeated structures in the image, and the presence of shapes, lines, or edges, that depict the outline of objects (geospatial landmarks in our case) but without information on their spatial relationships. One of the most famous local

features detector is the Scale-Invariant Feature Transform (SIFT) [20], published by David Lowe in 1999, which will be extensively described in the next Chapter. In the following years SIFT has been the object of countless extensions and modifications, and nowadays it is still widely used in many applications. Other local descriptors used in the literature are SURF (Speeded Up Robust Features) [2], BRIEF [3], and ORB (Oriented FAST and Rotated BRIEF) [25].

Aggregation

After the extraction of visual features from the input image, it is often necessary to aggregate the information into a fixed-length vector representation, to ease the subsequent similarity search. Yansheng Li describes these representations [19] as *middle-level features*, since they embed low-level hand-crafted descriptors into a feature space and they encode their spatial distribution to capture the semantic concept of the images. A popular encoding method is the Bag-of-Words (BoW), that employs k-means clustering algorithm to construct a visual codebook, and, based on the codebook, computes a histogram of local feature descriptors. In addition, the Fisher Vector (FV) has been proposed, which aggregates local descriptors using the Gaussian Mixture Model (GMM). Based on FV and BoW, the solution scheme proposed in Vector of Locally Aggregated Descriptors (VLAD) received a lot of attention, as it considers the cluster center closest to the feature point like Bow, but additionally counts the distance between local features and cluster centers, providing a more detailed description of the feature point.

4.1.2 Global descriptors

Global descriptors consists of those methods that can extract high-level features to comprehensively represent the visual content of an image, describing the entire scene without focusing on the individual details. To achieve an holistic encoding of the picture, traditional approaches adopted as strategy the subdivision of the image in a grid and the application of a local descriptor to generated patch. While providing a global description, this approach does not overcome the challenge of the semantic gap between the high-level and low-level visual features. Instead, the ability to extract high-level visual features and to encode many level of abstraction is the main strength of Convolutional Neural Networks (CNN). A great summary of recent deep learning solutions for CBIR has been done by Chen Liu in [4], and a summary

is shown in fig 4.2.

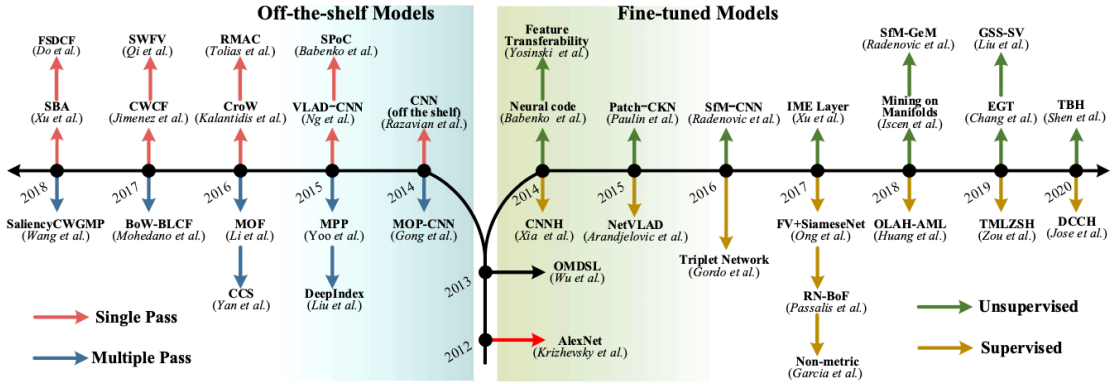


Figure 4.2: Relevant methods in deep image retrieval [4]

Many solutions schemes are based on the deep features extraction by means of Off-the-shelf models, that are used without further updating or finetuning their parameters. Usually these models are CNNs previously trained for classification tasks on very large datasets, where they learn how to encode the characteristics of the input image at different levels. For instance, fully connected layers have a large receptive field, while convolutional layers can preserve more structural details and high-level features; they can also be fused together to obtain more expressive representations. Since the dataset used to train off-the-shelf models mostly concern natural images, their direct application to remote sensing data does not bring great performances. Another option is represented by autoencoders, that can achieve data compression adn dimensionality reduction through a series of hidden layers and be exploited for feature representation also in the field of remote sensing.

Aggregation

On top of the feature extraction, it is often suitable to improve the discriminative ability of the obtained activation maps by means of feature aggregation or feature embeddings, considering the deep convolution features as a description of the local area of an input image. For instance, convolutional feature maps can be directly aggregated by spatial pooling, operation

that is usually followed by L2 normalization or PCA dimensionality reduction; but also max-pooling and avg-pooling are widely used, as well as R-MAC (Regional-Maximum Activation of Convolutions), that performs max-pooling over regions, SPoC (Sum-Pooled Convolutional), and GeM (Generalized Mean Pooling) [15]. Moreover, also in this case it is possible to apply the aforementioned embedding approaches, such as BoW or VLAD.

4.1.3 Learning DCNN representations

Beside the previous classification, it is possible to consider the end-to-end deep learning solutions, that do not distinguish the detection and description steps and take into account the fact that deep features not always outperform the classical hand-crated descriptors. This groups include the finetuning of deep networks pre-trained on image classifications, that are adapted for retrieval tasks, and those specifically designed to find similarities or dissimilarities between images. In particular, it is worth mentioning the Siamese networks, that update their parameters adopting a pairwise constraint in the loss, and the Triplet networks, trained to detect dissimilar and similar inputs simultaneously.

4.2 Siamese network

The method chosen to perform image matching based on the similarity of two input images is the Siamese Neural Network [32]. A Siamese Neural Network is a type of deep learning architecture that consists of two identical sub-networks, as shown in fig 4.3 These sub-networks share the same architecture, parameters, and weights, and are used to process two distinct inputs. The outputs of the two sub-networks are then compared to produce a similarity score, which reflects the degree of similarity between the two inputs. The weights of the Siamese network are trained in a way that minimizes the difference between the outputs of the two sub-networks for positive pairs (images of the same landmark) and maximizes the difference for negative pairs (images of different landmarks). The Siamese architecture is widely used in object detection applications, such as face recognition or signature similarity detection, but also in the case if one-shot learning.

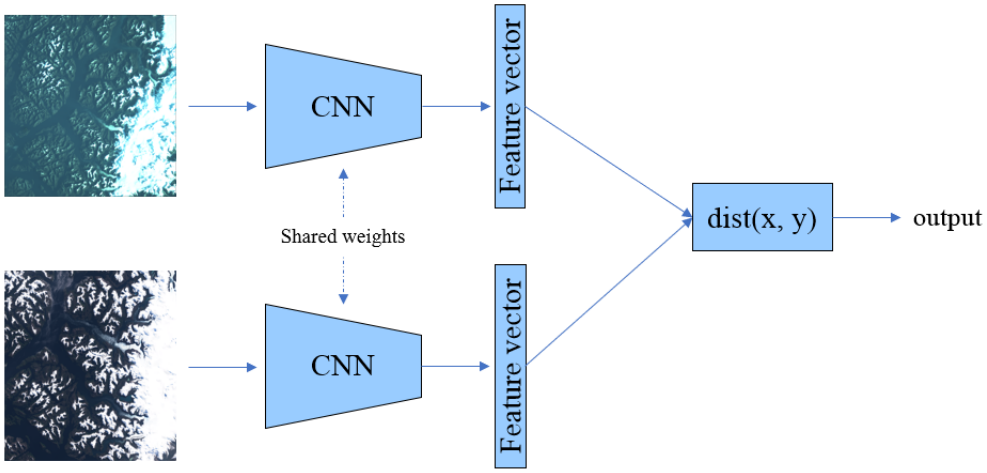


Figure 4.3: Diagram of a Siamese Neural Network

4.2.1 Backbone

In the case of Siamese networks, as we have already mentioned, the goal is to minimize the distance between similar input pairs, while keeping a high distance between those that represents different objects. Here it is important to point out that the object of interest in this case is the semantic similarity of the inputs rather than the direct distance: in fact, it might be possible that a distance measure computed directly on two different input images is smaller than the one computed on a pair of similar pictures. For this reason, the metric, which is usually a euclidean or cosine distance, is calculated on the feature representations that embed the semantic features extracted from a CNN. Therefore, it is very important to select an encoder that maximizes the semantic representation of the input images, in way to preserve only the most descriptive and distinctive features.

Some of the pre-trained networks that can be used as backbone are the following:

- **MobileNet**: it is a family of computationally efficient neural network architectures, specifically designed for mobile and embedded devices. They use a combination of depthwise seaparable convolutions and pointwise convolutions to reduce the computational cost of the network. Depthwise convolutions consist of two separate operations: a depthwise convolution, which applies a single filter to each input channel, and a pointwise convolution, which applies a 1x1 filter to combine the output of the

depthwise convolution across channels. This second operation is used to increase the dimensionality of the output, which can then be fed into subsequent layers.

- **VGG**: popular for its simplicity and effectiveness, VGG is a widely used neural network architecture composed by a series of convolutional layers, followed by fully connected layers (19 layers in total for the version used in this case). It uses small 3x3 convolution filters with a stride of 1 and padding of 1, which helps to preserve the spatial information in the input image. It has been designed by the Visual Geometry Group at the University of Oxford in 2014.
- **EfficientNet**: it includes different version of deep neural network architectures designed in 2019 by a team of Google researchers, who used NAS (Neural Architecture Search) and a compound scaling method to achieve high accuracy and efficiency. This method uses a scaling coefficient which controls the number of parameters and computational cost of the network. The architecture consists of a stem, which process the input image followed by a sequence of blocks, each consisting of a combination of depthwise separable convolution layers, to reduce the computational cost, and inverted bottleneck blocks, which increase the capacity of the network.

4.2.2 Loss function

Choosing the Loss function is one of the most important step in the design of a neural network. It is, indeed, the function that computes the distance between the label predicted by the model and the ground truth: this values needs to be minimised, and it is used as feedback to adjusts the weights of the model.

To ensure that the model can learn the appropriate feature representations, the loss function should sufficiently promote the learning of similarities as well as dissimilarities. In other words, it should encourage the model to learn better and better representations that encode the semantics of the images in the support set and bring related concepts close in the feature space. Let provide a mathematical formulation of the problem: with $S = \{(x1, x2, y), (x1, x2) \in K\}$ being a dataset composed by N pairs of feature vectors, such that a binary label $y \in \{0,1\}$ is associated to every pair $(x1, x2)$, we want z to take the value 0 whenever the feature vectors $x1$ and $x2$ are

semantically similar, and to take value 1 in the opposite case.

As we said, the twin sub-networks behave as feature extractors: both inputs are forwarded through the convolutional layers to obtain two latent features representations; those, are then used to compute the distance, which is finally fed to the loss function L.

The most commonly used loss functions are the Contrastive loss function and the Triplet loss function, whose goal, in both cases, is to learn representations that are close in the feature space for positive samples and far apart for negative samples:

1. **Contrastive Loss function**: it is defined as the negative log-likelihood of the predicted similarity between two samples.

   $$L = (1 - Y) * D_w^2 + Y * max(m - D_w, 0)^2$$

   where D_w is the Euclidean distance between the feature representations of $x1$ and $x2$, Y is the label indicating whether the pairs are positive (1) or negative (0), and m is the *margin*, i.e. an hyperparameter that sets the desired separation between positive and negative pairs in the feature space (by default is 1, the higher the more we force negative pairs to be far away).

2. **Triplet Loss function**: it is defined based on a set of three examples: an anchor, a positive example (similar to the anchor) and a negative example (which is dissimilar to the anchor). The loss is calculated as the difference between the distance of the anchor from the positive example and the distance of the anchor from the negative example, plus a margin:

   $$L = max(d(a, pos) - d(a, neg) + m, 0)$$

   where d is the Euclidean distance and m is the *margin*, a hyperparameter that sets the desired separation between the positive and negative pairs in the feature space. In this case, we can name the network as Triplet Network, which optimize similar and dissimilar pairs simultaneously.

4.3 Training and Evaluation

The training of a Siamese Neural network (SNN) is in general slower and harder than the one of a classic CNN, since for each forward pass both input images must be processed. The network structure is composed of a backbone

CNN pre-trained on ImageNet, which extracts deep features from the images, a couple of Dense layers, used to reduce the dimension of the output and to aggregate in a vector the feature maps produced by the convolutional layers, and a final Distance Layer which computes the Euclidean distance between the latent representations of the input images. For each of these modules, several configurations have been tried, as described in Chapter 6.3.

The use of a pre-trained network, which is adapted to a task and dataset different from those it has been trained for, is referred to as Transfer Learning. The Transfer Learning technique is based on the idea that a model trained on a large dataset efficiently generalizes over a wide range of visual objects, being able to encode in the feature maps several levels of visual elements. Filters learned at lower levels, that detect general patterns (such as lines, curves, dots, etc.), can be leveraged directly in a custom task different from the original without the need to train the network from scratch. Pretrained networks can be used as *feature extractors* by leaving out the original classifier layers, like in the case of the Siamese Network backbone, or they can be adapted to a more specific task by *fine-tuning* only the last layers, responsible for higher-order feature representations. In our case, the first 16 layers were "frozen" in the configurations using VGG19 as a backbone, while only the last block of convolutions has been fine-tuned. Concerning MobileNet, instead, the last two convolutional blocks were fine-tuned.

For training the SNN, we chose Adam as optimizer, for its speed and robustness, and the Contrastive Loss function, to avoid the processing of three images as input, which heavily affects the computational time. Moreover, a learning rate scheduler is introduced to modulate the learning value, which is initially set to 0.0001 and then reduced exponentially after the third epoch.

Given the low number of available pictures from OPS-SAT, it is not possible to create a dataset big enough to train a neural network. Therefore, the networks were trained on a different dataset, trying to minimise the performance issues deriving from the domain shift. A challenge, indeed, that is made even harder by the fact that most pre-trained networks learned features belonging to everyday objects, which do not match the main characteristics of remote sensing images. To tackle this problem, as is explained in the next paragraphs, several computer vision techniques have been used.

4.3.1 SEN12MS Dataset

Because of the low availability of OPS-SAT pictures provided with a ground truth on their location, the creation of a dataset ad hoc big enough to train the network for the task at hand was not possible. Therefore, an alternative solution to train the network is to use Sentinel-2 pictures and pre-process them to meet the characteristics of the application scenario.

In general, the construction of a large and diverse dataset able to cover most kind of landmarks that can be captured from a satellite is not straightforward, mainly because of the size required and the kind of highly representative features that different tasks need to focus on. Furthermore, although many kinds of RS datasets have been publicly released, the scale of available ones in terms of the volume of samples, the number of categories and the number of data modalities is still very limited [19].

Thankfully, it is increasing the number of researchers building annotated datasets that can be exploited in different remote sensing applications. For instance, the WorldStrat dataset [5] has been built in 2022 With the aim to provide one of the world largest and most varied public datasets, that tries to address a wide range of applications. In particular, WorldStrat tackles the issue of most datasets being only representative for the Global Northen areas, and provides a wide coverage to ensure more fairness, accountability and transparency in ML. It covers 10,000km^2 of land, in 4000 distinct locations, including very high resolution multispectral satellite imagery, equipped with different layers of information.

However, since the resolution of OPS-SAT pictures is lower than the one of Earth Observation satellites, the WorldStrat dataset is too detailed and heavy to process for our needs. Therefore, we chose to train the SNN on the SEN12MS Dataset [27], a curated dataset of georeferenced multi-spectral Sentinel 1 and 2 imagery composed by 180,662 scenes globally distributed over all inhabited land masses and each equipped with MODIS (Moderate Resolution Imaging Spectroradiometer) Land Cover maps. Each scene is each divided in image patches of 256x256 pixels and represented by triplet in GeoTiff format, consisting of a Sentinel-1 dual-pol SAR, a Sentinel-2 multi-spectral raster, and a MODIS land cover map. The dataset is based on randomly sampled regions of interest, as explained in the paper, resulting from four different seed values, one for each meteorological season defined for the northern hemisphere. For training the SNN, we used the "Summer" set, composed of 65 globally distributed scenes (like in fig 4.4) which are

divided in 45,753 patch triplets of 256x256 pixels: 80% has been used for training, while the remaining 20% for testing.

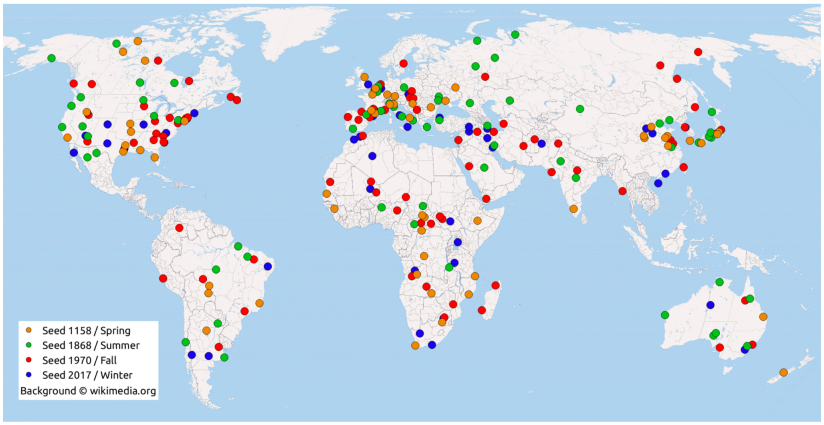


Figure 4.4: Distribution of the SEN12MS Regions of Interest

To generate the input pairs for the Siamese network, the Positive Pairs were constructed by coupling the RGB bands of a patch with the same figure transformed with a number of visual modifications, while the Negative Pairs were made of two randomly selected, non overlapped patches.

4.3.2 Data augmentation

The SEN12MS Dataset, although it is composed of remote sensing images, does not perfectly match the peculiarities of OPS-SAT pictures. Therefore, to make the model more robust to this domain shift we apply an extensive data augmentation, particularly designed to help dealing with colour shifts and perspective transformations.

In particular, since OPS-SAT orbits follows the Terminator line, i.e. the spacecraft is always in the shaded part of the Earth between day and night, the raw pictures appear in gray-blue light.

Moreover, the pixel resolution is lower compared to the images from Sentinel-2, which is a mission specifically designed for the collection of pictures for Earth Observation.

Last but not least, the main challenge is given by the change of perspective that characterize OPS-SAT images, which is the key point of this whole work. OPS-SAT takes pictures in every direction, by using the iADCS for pointing the camera over the specific area it wants to capture: based on the relative

position of the spacecraft, the objects in the image will be distorted by the angle of the point of view. On the other hand, Sentinel-2 operates always at the nadir, which means that it always points the camera directly downward perpendicular to the surface of Earth, minimizing the distortion in the images it takes.

To tackle these issues, since the heterogeneity of OPS-SAT pictures makes it difficult to define a single pre-processing pipeline that works effectively for all of them, it is preferable to use some techniques that make Sentinel-2 pictures comparable to the OPS-SAT ones. In the training of the Siamese Neural Network, the main goal is to provide a dataset that helps the network to focus on the geometrical features instead of the brightness or color of the pictures, providing also some examples in support of perspective projections. To to this, the construction of the pair of images labeled as *similar* is done by randomly selecting a picture from the Sentinel dataset and applying the following transformations:

- Random flip left/right
- Random flip up/down
- Random brightness change
- Random hue change
- Random affine transformation

Some examples of augmented data are shown in fig 4.5. In particular, it is possible to notice the application of hue changes, as well as rotations and stretches of the images. In the third example, a strong change of perspective is applied: since after the transformation the image has a different shape, we fill the void part by mirroring the transformed image on its borders, to avoid the presence of black pixels which could be misleading during the predictions.

4.3.3 Histogram Matching

In many cases the data augmentation performed at training time is not enough to make the model robust to colour differences, and the Histogram Matching technique allows to overcome the problem.

In the context of computer vision, an *histogram* is a plot showing the pixel distribution of an image in terms of intensity, in black and white pictures,

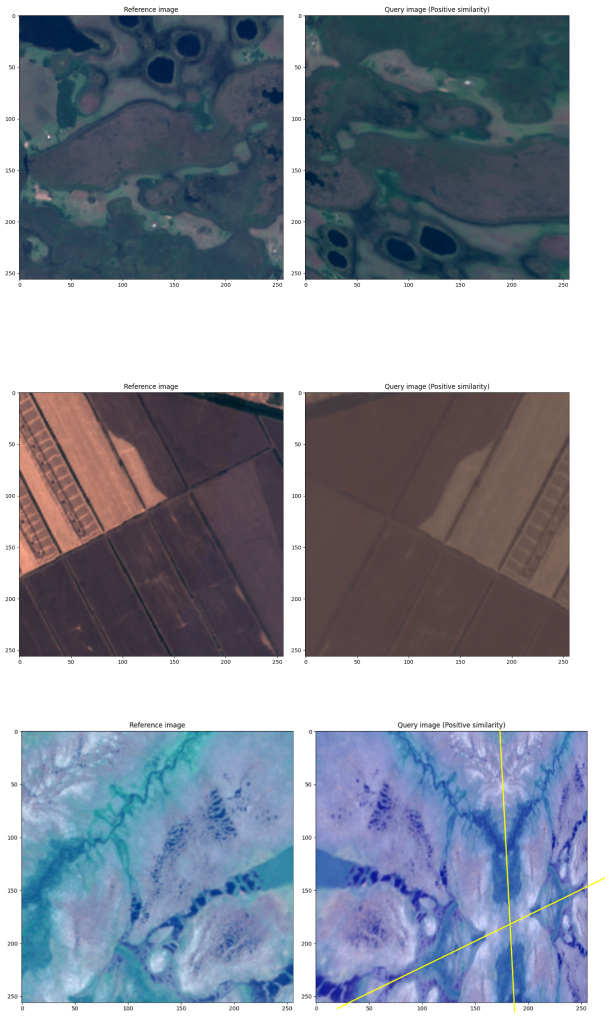


Figure 4.5: Examples of data augmentation

or of colours [1]. The range of possible pixel values is divided in bins, each counting the occurrences of that value inside the image. With the *histogram matching* procedure, given two images each having their own pixel distribution, we modify one of them such that its histogram matches the other one, which results in the homogenization of the hue and contrast between the two pictures 4.6. In the case of multiple channels, the matching is done separately for each channel.

[1] *https://towardsdatascience.com/histogram-matching-ee3a67b4cbc1*

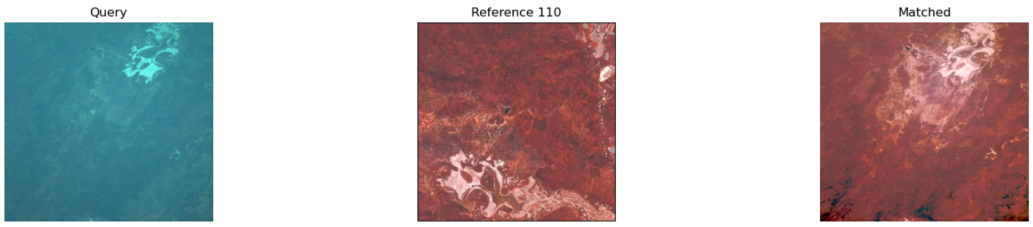


Figure 4.6: Histogram matching: the query histogram is modified to match the reference picture

4.3.4 Evaluation

The choice of the best configuration has been done evaluating the networks on the test set left out from the SEN12MS dataset. As already pointed out, this means that the comparison of the architecture is made on a common ground, but that their performances on the OPS-SAT images might not reflect the scores obtained at this step.

The evaluation of the Siamese Neural Networks is based on the accuracy of the predictions, defined by means of a threshold on the distance obtained as output: if its value is below the threshold, the prediction is considered as positive, i.e. the input images are labeled as similar, otherwise the prediction is negative. The final accuracy is computed as the percentage of correct predictions over the whole test set:

$$acc = \frac{TP + TN}{TP + FP + TN + FN}$$

where $TP = TruePositive$, $FN = FalseNegative$ and the others accordingly. The threshold value can be tuned based on the application, in this case a $threshold = 0.5$ has been used. In this case, being the classes balanced and of the same relevance, accuracy is a metric good enough for the given purpose.

Finally, it must be pointed out that the evaluation based on accuracy is useful during the choice of the model, while inside the final pipeline the outputs of the network are sorted in ascending order, without taking into consideration the label.

Chapter 5

Image Registration

The key step of this work is the registration of the query image with one of the Sentinel 2 dataset, which is the process of aligning two pictures of the same landmark, captured with a different angle, a different illumination and season or a different resolution. The ultimate goal is to obtain the geographic coordinates of the query image at a pixel level, according to the obtained transformation matrix.

Similarly to the image retrieval task, also in this case most of the reviewed methods were designed and developed taking into account natural object images, which differs from remote sensing imagery in many aspects: the shape of the landmarks are often better defined and with higher contrast, and the granule of the details is bigger than in satellite images.

Therefore, the main challenges are again represented by the need to adapt the models to remote sensing images and by the deep difference among the two pictures to overlap. The latter concerns in particular the geometrical distortion of the landmarks due to the different point of view of the satellite, and the cloud coverage, which can be misleading for the feature detection algorithms.

In this chapter, firstly, a review of the literature sheds light on the set of methods addressing the image registration task, then the selected models are more carefully described.

5.1 Related Work

The set of image registration methods includes all the algorithms whose goal is to alight two or more images, warping the first one to the coordinate of the reference image. This process can be seen under an algorithmic framework [18], consisting of six main steps, outlined in fig 5.1:

1. The creation of a Search Space of potential transformations
2. The extraction of most informative features from the two pictures
3. The choice of a similarity metric used to match the two sets of features
4. The definition of a search strategy that finds the optimal transformation
5. The choice of a resampling method, used to produce the corrected image
6. The design of a validation method to evaluate the image registration algorithm

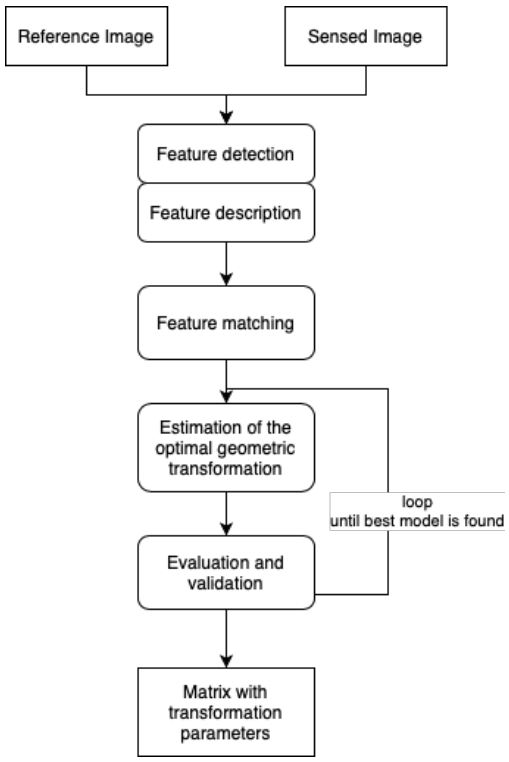


Figure 5.1: Framework of feature-based image registration

Each of these steps can be addressed by means of a wide variety of approaches,

and in many cases multiple steps can be aggregated, and solved together with a single method (for instance, neural networks can easily solve the second and third step end-to-end). Moreover, it is possible to further divide this pipeline into two sub-tasks: on one side, *keypoint matching* techniques are required to extract a set of pairings between the visual features of the two images, on the other side, once a set of valid matches is found, different methods can be used to *estimate the transformation matrix*.

The estimation of the transformation parameters is inherently an optimization problem: in the context of image registration, the most popular method to solve it is the RANSAC algorithm (RANdom SAmple Consensus), which is explained later. Nevertheless, many other approaches exist, such as gradient-based optimization algorithms, end-to-end neural networks, or those referred to as computational intelligence, which includes among others evolutionary algorithms, ant colony optimization, and swarm intelligence [31].

A global solution, able to solve the problem of image registration in a single step is still to be found. In fact, the use of deep learning techniques in end-to-end image registration solutions is still constrained by a lack of training data with sufficient geometrical deformations, despite the fact that they have shown great promise in the detection of features at various levels and in their representation from linear and nonlinear space [6].

5.1.1 Keypoint Matching

Many surveys [6, 21] agree in the division of image matching methods into two main categories, both giving as result a transformation matrix that allows to warp the query image to the coordinates of the reference one:

- **Keypoint features detection**: in this case the algorithm searches for local features, also referred to as *interest points*, that can be described by the appearance of the pixels in a neighbourhood of the point location. Good interest points have the quality of being well localised inside the picture, and of being stable under image perturbations, to guarantee their computation with high repeatability. The primary feature used is the keypoint, a punctual region characterized by invariance to geometric deformations, illumination, etc., and easily located and described, but in general, local features also include lines, curves, edges, and contours. These methods are formed by the three steps of feature detection, feature description, and feature matching.

- **Region-based features detection**: these solution schemes, instead of identifying salient image structures, are based on the comparison of pixel intensity between the two images, measuring the similarity usually by means of sliding windows. These methods are, in general, more sensible to image distortions. The image alignment in this case is obtained by means of an iterative optimization process, where the sensed image is transformed several times trying to overlap it to the reference image until the maximum similarity among them, defined by a specific metric, is achieved.

Because of the higher performances both in terms of accuracy and speed, from now on the work will consider only the keypoint feature-based approaches, and in a particular way those that have shown satisfactory results in the remote sensing domain.

Traditional methods developed in the past few decades are grounded on mathematical theory, and consist of handcrafted operators and filters for the detection of specific class of features. For instance, the Harris Corner Detector is a popular gradient-based algorithm for the detection of edges and corners in an image, that is often used for the extraction of visual features.

Other classical machine learning methods that it is worth mentioning are FAST (Features from Accelerated Segment Test), an efficient corner detector; SIFT, a robust method extracting blob feature points, and its variant SURF (Speeded-Up Robust Features), which relies on integral images and uses an Hessian matrix-based measure to detect the features. While Harris and FAST are only feature detectors, the others are both detectors and descriptors.

5.1.2 Learning methods for Keypoint Matching

Since the breakthrough of convolutional neural networks (CNN) in computer vision, that traces back to 2012 with AlexNet CNN winning the ImageNet competition, many deep learning methods has been developed, demonstrating unparalleled performances with respect to previous techniques. These exceptional results, particularly evident on image-based applications, are due to the ability of CNNs to extract high-level as well as deep features, autonomously learning the best filters by means of their training process. While the hidden layers learn the features, the output layers can be used for feature extraction and feature matching purposes. Also in the case of image matching, a popular architecture is the Siamese or pseudo-Siamese network,

used to find corresponding patches between two input images. However, in the context of feature based image registration, which is characterized by the matching of unstructured or non-Euclidean data points, deep CNNs still struggle to extract the spatial relationships among them [6, 22].

Among the large number of deep learning methods for image matching developed in the last years, SuperPoint, SuperGlue and LoFTR are worthy to be mentioned, for the novelty of the approaches they introduced, and as they can be considered as the current state-of-the-art for the task of image matching.

Despite the great performances of deep learning, to tackle the specific problem of remote sensing image registration classical methods as SIFT and SURF are still widely used, thanks to their robustness and speed, and because of the difficult adaptation of deep learning solutions, originally developed for natural images, to satellite imagery. Moreover, another burden in deep learning models is to prepare annotated training datasets. For these reasons, the research in this direction remains open, and many applications keep using traditional techniques.

SuperPoint

SuperPoint is a fully-convolutional model presented by DeTone, Malisiewicz and Rabinovich [20], that operates on full-size images and jointly computes pixel-level interest point locations and associated descriptors in one forward pass. The model architecture is composed by a VGG-style encoder, that processes and reduces the input image dimensionality, mapping it to an intermediate tensor with smaller spatial dimension and greater channel depth, and by two non-learnable decoders. The Interest Point Decoder exploits a Softmax layer and a reshape operation to restore the spatial resolution of the input, while the Descriptor Decoder produces fixed-length descriptors by applying a bi-cubic interpolation and L2-normalization to the activation maps. The paper introduces also a self-supervised framework to train the SuperPoint network and adapt it to the domain of interest. This procedure includes two main steps:

- *Interest Point Pre-training*: the first step is the training of a base interest point detector on synthetic data, consisting of simplified 2D geometries obtained via synthetic rendering of quadrilateral, triangles and ellipses. This base detector, called MagicPoint, has great performances on the synthetic shapes, but it struggle to generalize to real images.

- *Homographic Adaptation*: this step is introduced for a self-supervised training on real-world images. The idea is to generate pseudo-ground truth interest points to train the interest point detector on the target, unlabeled domain. To do so, the input image is warped multiple times with random homographies to help the model see the scene from many different viewpoints and scales. The resulting model is Super Point.

SuperGlue

If SuperPoint proposed an innovative deep-learning method for the extraction of keypoint features, SuperGlue introduced an equally pioneering method for image matching, starting from a set of interest points previously detected.

Presented in 2020 [26], SuperGlue is a neural network that matches two sets of local features by jointly finding correspondences and rejecting non-matchable points. Its architecture is based on an Attentional Graph Neural Network, designed to compute matching descriptors by letting the features communicate with each other, and by an Optimal Matching Layer, that solves an optimal partial assignment problem using the Sinkhorn algorithm to establish pointwise correspondences from the local features of two input images.

LoFTR

LoFTR (Detector-Free Local Feature Matching with Transformers) [29] is a novel area-based method for image matching. It is based on Transformers, a class of models initially developed for Natural Language Processing (NLP) that have recently been adopted also for computer vision tasks.

The idea of the paper is, in a first moment, to establish pixel-wise dense matches at a coarse level, using Transformer with self and cross attention layers to process the dense local features extracted from the convolutional backbone. Later, the fine-tuning of good matches is done at a finer level, filtering only the high confidence matches and refining them to a sub-pixel level with a correlation-based approach.

5.2 SIFT

The Scale-Invariant Feature Transform algorithm, presented for the first time by David G. Lowe in 2004 [20] is a popular machine learning method for detecting and describing local features in an image, widely used for its simplicity and efficiency. As stated in the paper, SIFT algorithm is invariant to translation, scaling and rotation, and it can handle soft illumination changes and affine or projective transformations.

The main idea of SIFT is to generate a large collection of features that densely cover the image over the full range of scales and rotation. Its implementation is quite articulated, but it is possible to identify 5 main steps:

1. **Scale-space extrema detection**: it aims to identify location and scale that can be repeatably assigned under differing views of the same object, i.e. searching for stable features across all possible scales (for instance, a curve is well appreciable only at a certain scale). The key locations are chosen at the maxima and minima of a Difference of Gaussian (DoG) function applied in the scale-space.

   - The scale-space function is defined over different octaves of the input image, where each octave's image size is half the previous one. Within an octave, images are blurred at different scales, by means of a convolution with a Gaussian kernel.

     $$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y)$$

     where L is the scale space function, G is the Gaussian kernel with σ as scale parameter, and I is the image.

   - This Blurring operation is done for each octave of the Gaussian Pyramid

   - The DoG is computed as the difference between adjacent Gaussian images, as depicted in fig 5.2

   - Finally, a search for local extrema over scale (i.e. across different σ values) and space (8 neighbors pixels of a point) is performed, which gives a list of (x, y, σ) values for potential keypoints located in (x,y), at scale σ.

2. **Keypoint localization**: it is used to refine the potential keypoints extracted. Borrowing a concept used in Harris corner detection to reduce the effect of DoG high response for edges, SIFT exploits Taylor

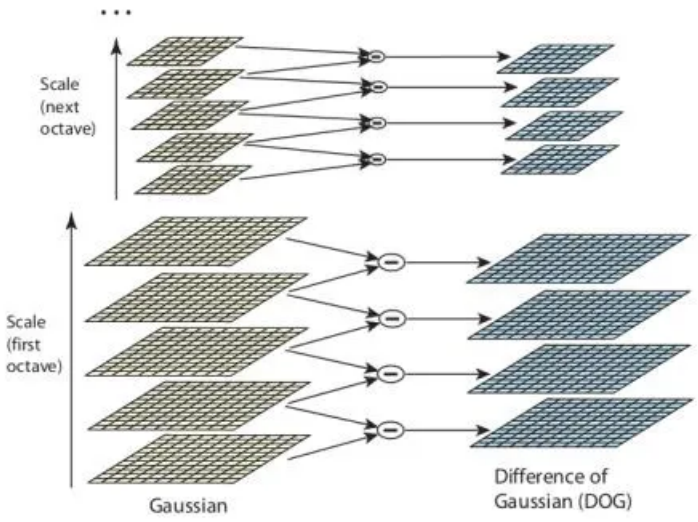


Figure 5.2: Difference of Gaussian between images within an octave

expansions of the scale space function to get a more accurate location of the extrema, and filters out those below a pre-defined threshold value (*contrast Threshold* in OpenCV).

3. **Orientation assignment**: an orientation is assigned to each point to achieve invariance to image rotation. To do so, for each point locations SIFT computes the gradient magnitude and direction of its neighborhood. Then, it creates an orientation histogram with 36 bins covering 360 degrees. Peaks in the orientation histogram correspond to dominant directions of local gradients.

4. **Keypoint descriptor**: After taking a 16x16 neighborhood around a keypoint, it is divided into 16 sub-blocks of size 4x4. For each sub-block, a 8 bins orientation histogram is created, which means that a total of 128 bin values are available. A feature vector is then built by normalizing to unit length the 128 values obtained.

5. **Keypoint matching**: finally, the keypoints between two images are matched by identifying its nearest neighbours. The ration between the closest distance and the second-closest distance can be taken into account to eliminate false matches.

Although quite old, SIFT is still very popular in computer vision applications,

mainly because of its robustness and speed. In fact, it can handle changes in illumination, scale, rotation, and viewpoint with a reduced computational time. However, in many challenging scenarios its performances are overtaken by those of deep learning solutions, which are currently the state-of-the-art in most computer vision tasks.

5.3 RANSAC

RANSAC, also referred to as Random Sample Consensus algorithm, is an iterative method to estimate the parameters of a model proposed by Fischler and Bolles [7]. Starting from a list of keypoint matches between two images, it generates candidate solutions for the parametric transformation by using the minimum number data points required, and finding the best partition of those points in inliers and outliers. In this case are referred to as inliers the points that are conssitent with a dominant motion estimate.

The RANSAC approach of starting from the smallest set possible and proceeding to enlarge it with consistent point makes it very robust to outliers (it can deal with situations where more than 50% of the data points are outliers).

The algorithm is simple to understand, and can be summarized as follows:

Algorithm 1 RANSAC

Define:
S - the number of sampled points required
N - the number of iterations to be done
ϵ - the tolerance threshold used to identify a point with a good fit
τ - the minimum threshold for the fraction of the number of inliers over the total number points in the set
do
1. Randomly select S data points
2. Compute the model parameters using the sampled data points
3. Determine how many points fit the estimated model within ϵ

if inliers ratio $\geq \tau$ **then**
Re-estimate the model parameters using all the identified inliers
end if
while The best model is found

Before running the algorithm, some parameters have to be defined:

- **S**: is the minimum number of points needed to estimate the parameters of the chosen model. For instance, when estimating an affine transformation it is enough to use three points, while a perspective transformation requires four points. The largest S is, the harder to find a set of inliers.
- e: is the outlier ratio $\frac{\#outliers}{\#datapoints}$, i.e. it represents the probability of having an outlier
- **N**: is the number of iterations, which should be chosen such that, with probability p (usually 0.99), at least one random sample set is free from outliers. To determine it, we can start defining the probability of selecting at least one outlier in one trial:

  $$1 - p = 1 - (1 - e)^S$$

  The probability to select at least one outlier in all N trials is then:

  $$1 - p = (1 - (1 - e)^S)^N$$

  Thus, with some manipulation,

  $$\log(1 - p) = \log(1 - (1 - e)^S)^N$$
  $$\log(1 - p) = N \log(1 - (1 - e)^S$$
  $$N = \frac{\log(1 - p)}{\log(1 - (1 - e)^S)}$$

RANSAC is a very popular method used on top of keypoint matching algorithms to estimate the transformation that relates two pictures. Its main strength is that it can robustly deal with outliers, working well in the estimation of models with up to 10 parameters. Moreover, being easy to understand makes its implementation straightforward.

On the other side, its computational time grows quickly with the fraction of outliers and with the number of parameters needed to fit the model.

Over the years many variants have been developed, such as PROSAC (PROgressive SAmple Consensus), in which random samples are initially added from the most "confident" matches, thereby speeding up the process of finding a (statistically) likely good set of inliers. Another version is DSAC, designed to be differential so that it can be used in end-to-end training of feature detection and matching pipelines.

5.3.1 Geometric Transformations Models

Before estimating the parameters of the transformation matrix, the kind of geometrical function to model must be defined. General categories of transformation are rigid, similarity, affine, and projective, as exhaustively explained in [30] (fig. 5.3):

- **Rigid**: it is a combination of rotation and translation, also referred to as *Euclidean transformation* since Euclidean distances are preserved.
- **Similarity**: it additionally allows the variation in scale. It preserves angles between lines.
- **Affine**: it is an operation that preserves parallel lines. It can be written as $x' = \mathbf{A}x$, where $\mathbf{A}$ is an arbitrary $2x3$ matrix:

$$x' = \begin{bmatrix} a_{00} & a_{01} & a_{02} \\ a_{10} & a_{11} & a_{12} \end{bmatrix} x \tag{5.1}$$

  An affine transformation can be estimated starting from 3 control points.
- **Projective**: it is an operation that preserves straight lines. It is also known as *perspective transform* or *homography*, and it describes the relationship between two images of the same scene that are related by a perspective or planar transformation. The projective transformation can be defined as

$$x' = \tilde{\mathbf{H}}x$$

  where $\tilde{\mathbf{H}}$ is an arbitrary $3x3$ matrix. $\tilde{\mathbf{H}}$ is homogeneous, i.e. it is only defined up to a scale (two matrices that differ only by a scale are equivalent). This transformation can be estimated starting from 4 control points.

Any remote sensing image inherently present some geometrical distortion given the altitude of the satellite, which are commonly addressed by the application of affine transformation on the sensed image. Although from a more precise geometrical perspective the two sensed images are related by a perspective transformation, in practice, the affine transformation is generally used in multi view image registration, where the distance of the camera is large as compared to the scanned scene.

To compute the parameters of the matrix describing an affine transformation, at least three pairs of point matches are needed.

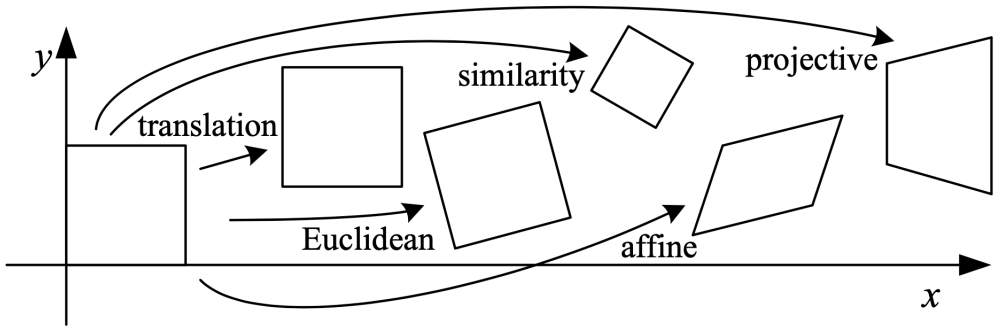


Figure 5.3: Basic set of 2D planar transformations

5.4 Evaluation

The evaluation of an image registration algorithm aims to determine how precise is the overlap of the two input images in terms of number of pixels. To briefly recap, the process followed by feature-based methods starts from the extraction of keypoint features from the reference and sensed images, it then performs a matching between the features of the two images, and, after having filtered out the outliers, it uses the remaining inliers points to estimate the geometrical transformation ϕ that enables the registration.

As pointed out in [23, 8], the most important factors to take into account are the following:

- **Repeatability**: it is a criterion that gives a measure of the stability of the detector. The higher the repeatability, the higher the number of feature points extracted in the corresponding positions of the two input images.

- **Correct Matching Rate (CMR)**: clearly, the number of detected matching pairs should be large, and, more importantly, the inliers ratio should be as high as possible to guarantee a good accuracy in the registration. It is defined as the number of correct mathings over the total number of mathcing pairs. At the same time, it is possible to define the **Matching error** as the number of false matches.

- **Distribution of Matching Pairs**: the features extracted should be

uniformly distributed over the images, to allow the estiamtion of a transformation model that can handle the local distortions between remote sensing images.

- **Distinctiveness**: it represents the uniqueness of the feature descriptor. The more distinguishable a descriptor is, the better the results in the image registration.
- **Localization error**: it is the displacement of the control points coordinates due to their inaccurate detection. Usually there exists a trade-off between the number of detected matching pairs and the mean localization error.
- **Alignment error**: it is denoted by the difference between the mapping model used for the registration and the actual image geometric distortion. Because of the assumptions done in any case to simplify the mathematical description of the problem, and because of the intrinsic approximation that exists in a model, the alignment error is always present.

Chapter 6

Experimental results

Once defined a roadmap to follow for solving the problem of attitude estimation, the first goal is to produce a proof of concept to demonstrate the feasibilty of the proposed solution and to show how the different steps can be connected together. The pipeline implemented is not intended as a final operative method, but it is rather a baseline well prepared for future developments.

In this chapter the workflow of the demonstrative pipeline is presented, followed by an analysis of its performances with a focus on its strengths and weaknesses.

6.1 Proof of Concept

The Proof of Concept, created as a Jupyter Notebook, goes through all the steps required to compute the attitude, from the information retrieval about the state of the satellite at the moment it captured the picture, to the obtaining of the quaternions, computed from three geolocalized points of the query image.

Each major step of the pipeline is independent, and can be easily replaced by a different method.

6.1.1 Satellite State reconstruction

The starting point is a query picture and the related capture timestamp, such as the one reported in fig 6.1, which is an image of the Australian Carneige Lake taken by OPS-SAT the 5th of February 2022. Before running the notebook, it is also necessary to download the satellite telemetries covering a time window that includes the timestamp of the query picture. In particular, we need:

- The recorded *quaternions*, that can be downloaded from the webMUST software in a csv file format. It is possible to obtain both the quaternions from the iACDS and the cACDS, but since the iACDS is used only in precise pointing mode, its data are not always available.
- The *TLE*, which can be requested from the CelesTrack website.

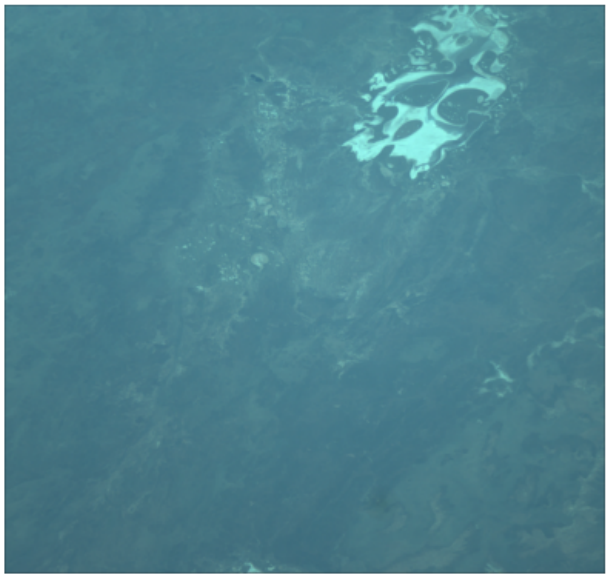


Figure 6.1: Example of query picture. Carneige Lake, Australia.

From the information about the quaternions, the TLE and the timestamp, it is possible to retrieve the satellite state at the time the picture has been taken:

```
SAT STATE:
Timestamp: 2022-02-05 09:41:34+00:00
Position: [ 4876395.32150191  3809582.2336253  -3067273.74523708]
Lat, Lon, Alt: (-26.366498998277475, 117.08094441258085, 528381.5625)
```

```
Euler attitude (roll, pitch, yaw):
    (110.25707277151217, 18.995423411478463, 108.43704784094791)
angle to nadir: 25.360767848928276
pointing to earth: True
```

In particular, computing the intersection between the Earth geoid and the axis -Z of the satellite, which is its pointing direction, we get latitude and longitude of the point captured by the onboard camera, as well as its field of view. This data comes from the historical telemetries and it is not always accurate; in fact, the output coordinates of these computations are often shifted with respect to the actual landmark observed, and sometimes they are not available or completely off (e.g. reporting that the satellite is not pointing to Earth). In the case of our example, the axis -Z has the following values of latitude, longitude and altitude:

```
ll_minus_z = [  -26.9459838 ,   119.63004202, -5670.33333944]
```

6.1.2 AoI definition

The above information are used to draw the Area of Interest in fig 6.2, shaped as an ellipse around the position of the satellite on the Ground Track. The Area of Interest should be as small as possible, but large enough to include any possible landmark visible on the ground from the position of the satellite. For this reason, the ellipse shape has been chosen, placed with its longest axis along the orbit and sloped of 97.5 degrees (the precision of this detail should be improved by propagating the TLE to understand if OPS-SAT is traveling in an ascending or descending direction).

6.1.3 Construction of the reference database

At this point, we can download the Sentinel pictures covering the land inside the AoI from the Copernicus Open Access Hub. The library `sentinelsat` allows to shape the query for the database by selecting, in addition to the target geographical region, the time window we are interested in and the required cloud cover percentage. As explained in Chapter 3, only more recent products are available online, while the historical data are saved in a Long Term Archive, whose access is more time consuming and limited by a per user quota. Therefore, to create the reference database, in general it is suggested to define a time window around the timestamp of the query picture, whereas for older images it is better to select a more recent period.

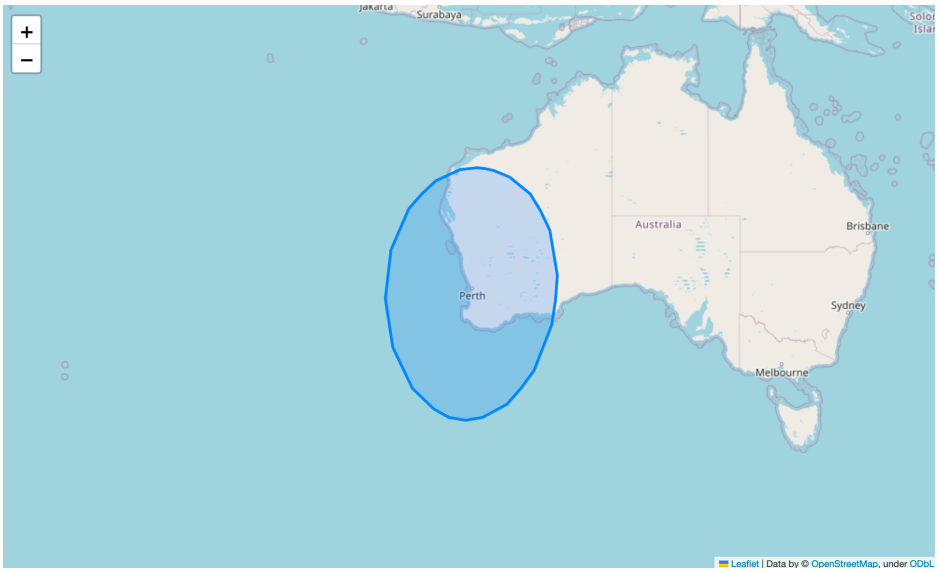


Figure 6.2: Area of Interest around OPS-SAT position on the Ground Track

Since each product contains on average more than 1GB of data, at this step the download is limited to the quicklook images, which are lightweight versions of the full resolution picture, downsized to 512x512 pixels.

6.1.4 Image Retrieval

Once the reference database is created, we can proceed with the Image Retrieval step, which consists of predicting by means of a Siamese Neural Network a similarity score for each pair composed by the query and one of the reference images. The predictions are then sorted, and only the top K (with K equal to 5 or 10) are forwarded to the next step of the pipeline. Since the network is computing the distance between the feature vectors obtained on the forward pass, the lower the score, the more similar the two input images are. Practical experiments showed that, despite the data augmentation techniques used during training, the network is very sensible to the colours. This represents an issue especially in pictures of certain areas such as Australia, where the land captured by Sentinel-2 appears bright red, while OPS-SAT pictures are always covered by a blue shadow. To tackle this problem, before feeding the pair to the model, the histogram of the query image is matched to the one of each reference as in fig 6.3.

As mentioned before, only the reference images with the lowest difference

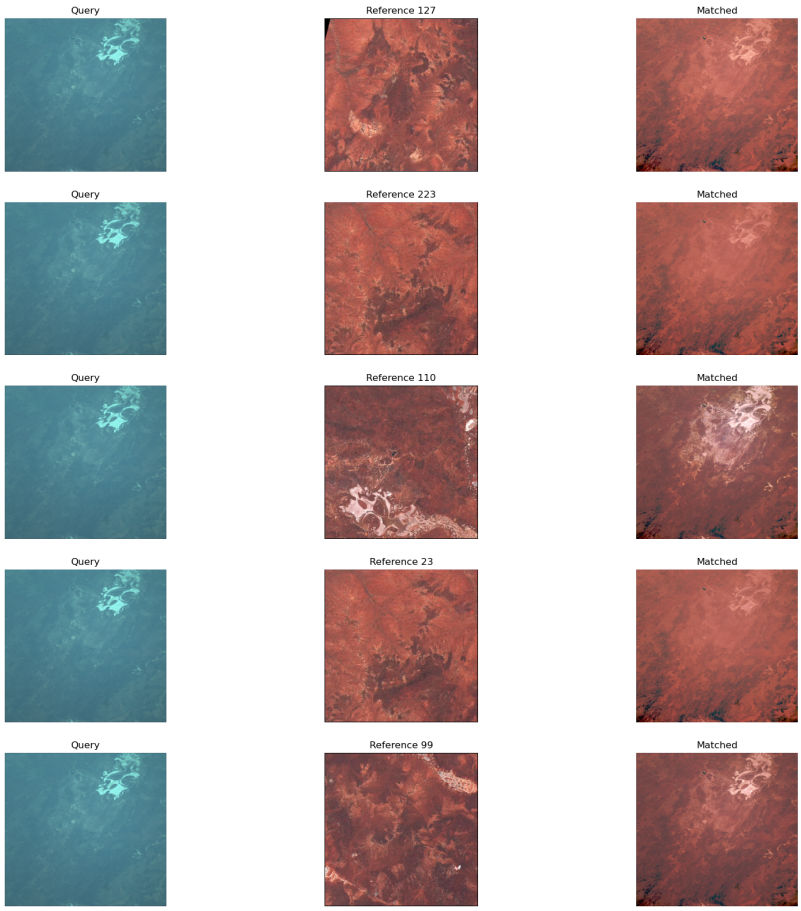


Figure 6.3: Histogram matching technique applied to the query image

score are retained and feed to the Image Matching algorithm, while the remaining ones are discarded. The output of the network, sorted by descending similarity is shown in fig 6.4.

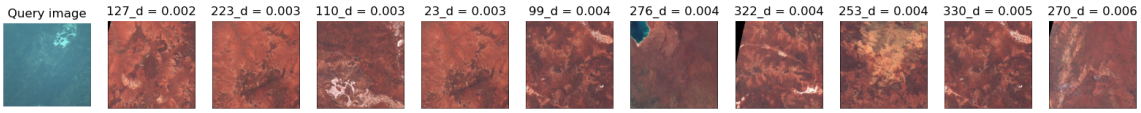


Figure 6.4: Reference images sorted by the similarity score obtained from the Siamese Neural Network. It is possible to notice that the target landmark appears in the third most similar figure.

Another way to visualize the output of the Siamese network is to show where

the top K most similar images are located (fig 6.5) or to display the distribution of the distance score through a heatmap.

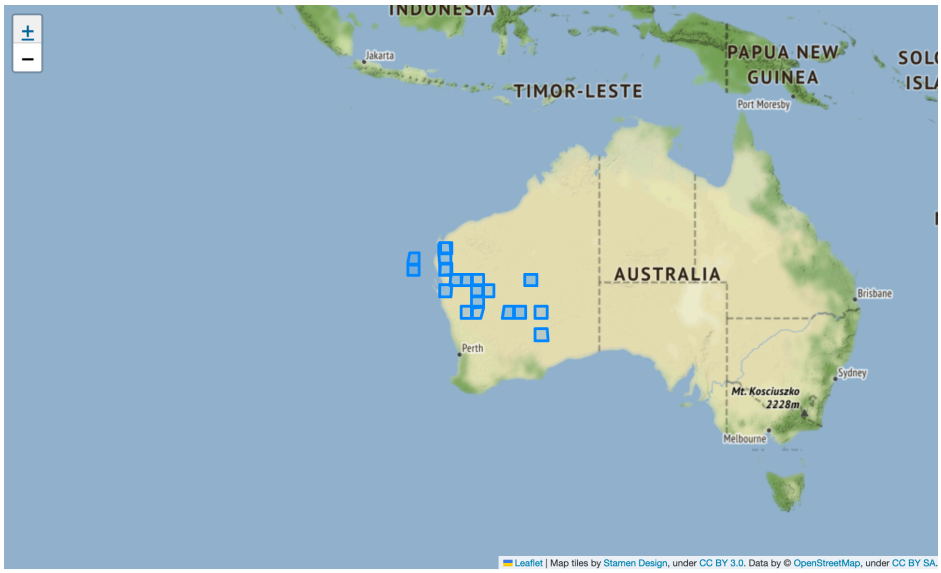


Figure 6.5: Location of the top 20 most similar images.

6.1.5 Image Matching

The stage of Image Matching has the goal to find keypoint correspondences at a pixel level between two pictures and to estimate a transformation that allows to overlap them. It receives as input a couple of images, composed by the query and one of the selected references; for each pair it produces as output a set of keypoint pairs matched between the two scenes and the corresponding descriptors. This step not only finds precise matches between the input pairs, but it is also used to refine the similarity search performed coarsely with the Image Retrieval method, as showed in the results reported in fig 6.6.

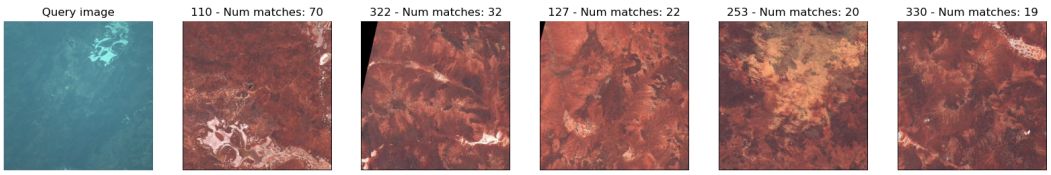


Figure 6.6: Ranking of reference images based on the number of matches found by SIFT.

The new ranking presented in fig 6.6 is based on the number of matches found by the SIFT algorithm, which computes a set of keypoints and descriptors from the black and white version of each input picture, and then matches them by means of a KNN algorithm. Also in this case, the outputs proved that they can benefit from histogram matching, even though the images are processed in black and white. Moreover, it can be proved that the intermediate step of Image Retrieval is, in general, beneficial for the detection of the target landmark in the reference database, and necessary in the proposed architecture: the application of SIFT directly on the entire database to perform a similarity search failed in finding the right match.

On top of SIFT, the RANSAC algorithm is used to filter out spurious matches, dividing the total set of keypoint pairs in inliers and outliers, as shown in fig 6.7. Usually, it is applied to the reference image ranked first in the SIFT output, but a visual check of the correctness and eventual correction might be required. RANSAC could also be used to directly estimate the transformation matrix to overlap the two images, but the experiments outcome pointed out the need to add an extra step for a more robust estimation of the homography matrix.

Being two independent algorithms, it is possible to replace SIFT with any other keypoint matching method. It is clear that the goal is to produce as many matches as possible, in order to get a large set of inliers and a robust estimation of the transformation.

6.1.6 Homography matrix transformation

As discussed in Chapter 6, it is suitable to add an iterative method to evaluate the precision of the image registration and find the best transformation matrix, which abstractly represents the geometric mapping function from the sensed image to the reference one. Therefore, instead of using the homography matrix estimated by RANSAC, the transformation is computed by means of a function of the `skimage` library, which provides the affine transformation that best fits the mapping. The obtained matrix is then used to transform the reference image, so that it can be overlapped with the query, as well as the keypoints, like in fig 6.8

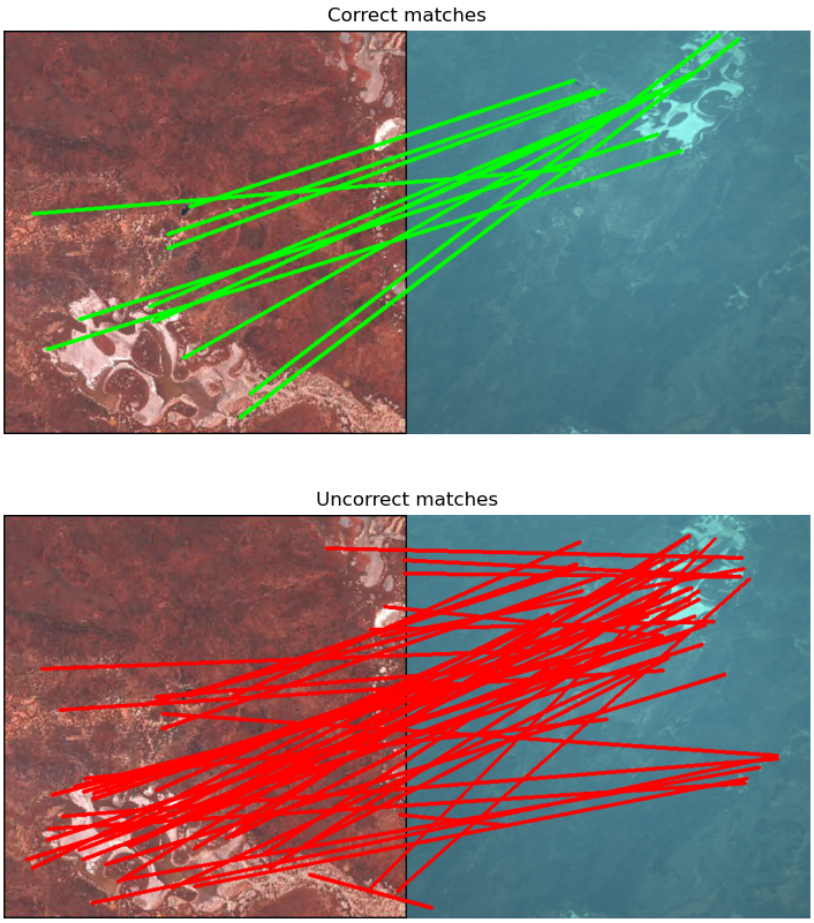


Figure 6.7: Set of inliers (above) and outliers (below) obtained with the RANdom SAmple Consensus algorithm.

6.1.7 Coordinates extraction

Even though the Image Registration is the key step for geo-localizing the OPS-SAT picture, there is still the need to extract the coordinates of at least three pixels from the reference image. To lighten the whole process, at the beginning it has been decided to work on quicklook images, that are downsampled and, instead of having pixel-wise geographical information, they are only provided with their corner coordinates. Therefore, two different approaches are available to obtain the keypoints location:

- Computing them geometrically with reference to the corner coordinates

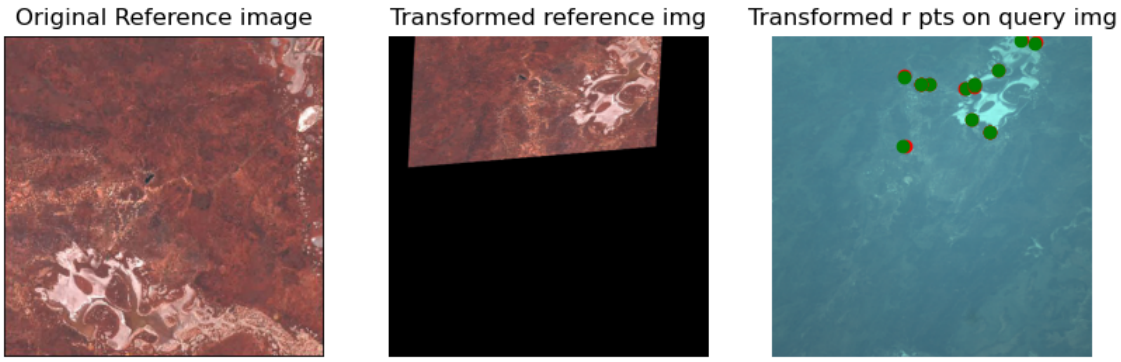


Figure 6.8: Overlapping of the reference image onto the query. In the query (right) the green dots are the reference keypoints transformed, while the red ones are the keypoints of the query.

- Downloading the complete product from the Copernicus OAH and retrieve that info directly from the tiff file.

Downloading the complete product provides better results in terms of precision, because they are obtained from full resolution images, but it has two downsides: it requires to run again SIFT and RANSAC since the shapes of the images are different, and the size of the data to download is very large (around 1GB), which adds latency in the whole process. For these reasons, it has been decided to compute the coordinated starting from the quicklook image boundaries, which provides a coarse but still acceptable accuracy.

To apply the first method proposed, we make the assumption that the land represented in the picture is flat, that allows us to apply planar gemotry formulas and simplify the computations. The procedure consists of the following steps:

1. Obtain the resolution in meters of the reference image
2. Identify the upper left corner of the picture/tile
3. Shift South by a number of km proportional to the y pixel of the keypoint
4. Shift east by a number of km proportional to the x pixel of the keypoint
5. Repeat for each keypoint of interest (at least 3)

The algorithm described above has to be intended as a guideline, as it might require some adaptations in the case the Polygon of the tile is not rectangular.

For moving in the coordinate reference system, the `haversine` library proved to be a very useful tool.

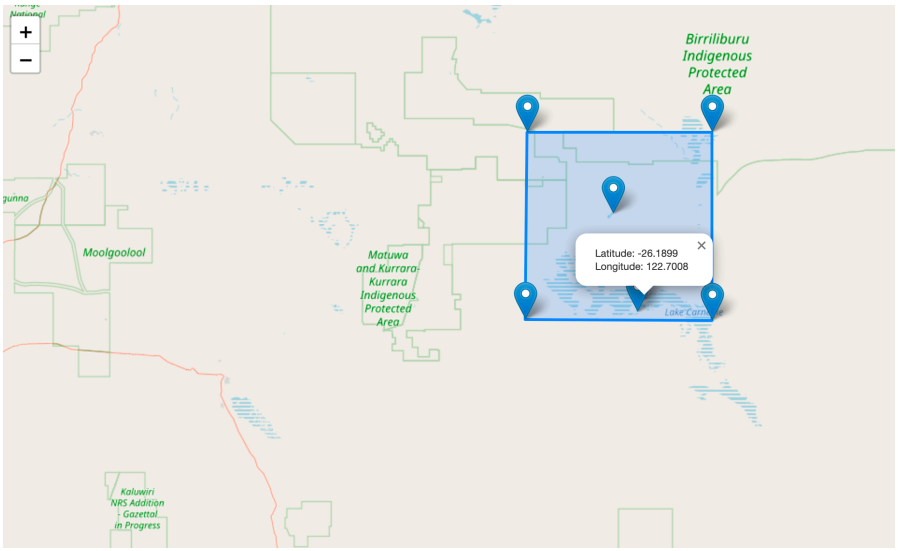


Figure 6.9: Reference tile with info about the geographical coordinates of the corners as well as three of the matched keypoints.

In the figure 6.9 it possible to identify the four corners of the reference image and the keypoints geo-localised with the method just explained.

With the help of a python script provided by the OPS-SAT Team, it is finally possible to retrieve the attitude of the spacecraft. The script requires as input three points from the query image, equipped with geographical and pixel coordinates, as well as the TLE, the timestamp of the picture, and its resolution. It is then able to retrieve, by solving a geometrical problem, the quaternions of the satellite and its pointing vector.

In Fig 6.10 we can see that our method successfully localize the query image, and allows to obtain attitude values that coherently relate the picture with the position of the spacecraft in the orbit. The orange mark, which represents the pointing vector computed from the quaternions recorded in the telemetries, is clearly shifted from the target, and thus misleading when there is the need to recognize the landmarks in the picture.

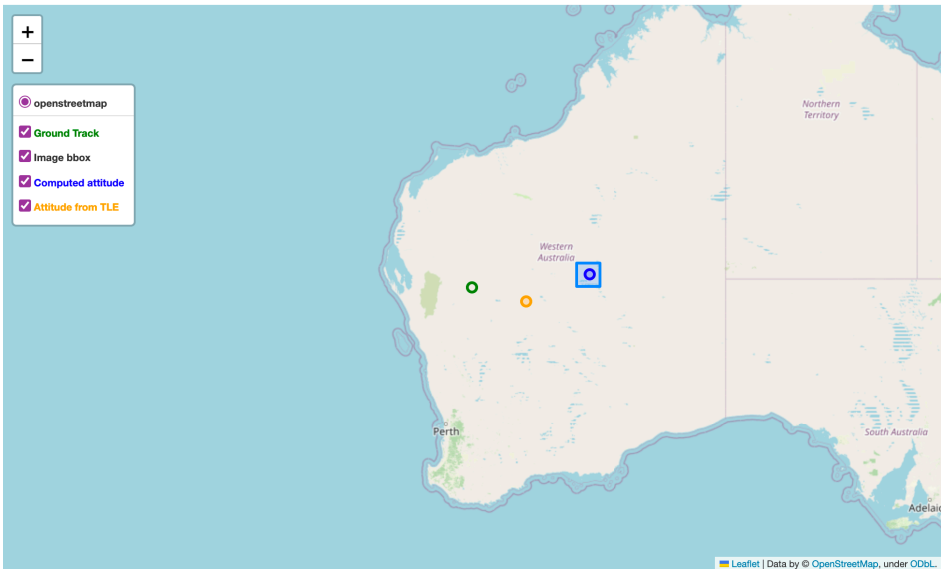


Figure 6.10: The approach proposed successfully localizes the query image and provides an attitude value coherent with the position of the spacecraft.

6.2 Failure and Success Scenarios

The proposed method represents a solid and successful basis for developing an offline visual-based attitude estimator. The proof of concept demonstrates that the architecture can execute all the steps required to estimate the attitude parameters of the spacecraft in relation to a picture taken by the on-board camera. Moreover, its modularity makes it straightforward to increase the efficiency and the precision of each step independently from the others, enabling a continuous and eager development of the software.

In particular, the current approach has been proved to produce satisfactory results on pictures with landmarks well defined, such as the Carneige lake in Australia, islands or mountains, like the ones depicted in fig 6.11. In those cases, characterized by high contrast features, both the image retrieval and keypoint matching algorithms can detect the similarities. The use of histogram matching technique helps in those cases where the major obstacle is represented by the color shift between the two representations. However, sometimes also the rotation of the landmark can affect the outcome, especially for what concerns the Siamese network. In fact, it was noted that the performances of the network increases if the input images are rotated in the same direction.

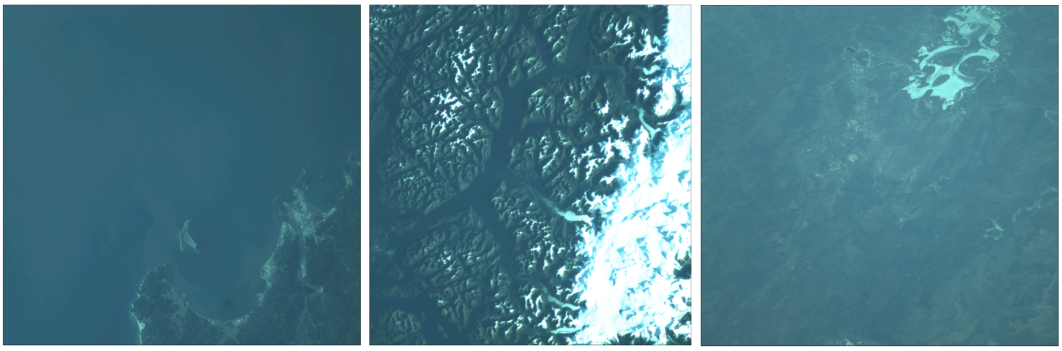


Figure 6.11: Success scenarios.

Nevertheless, the presence of clouds undermines the detection of the right features, and even the application of masking techniques was not beneficial. Another obstacle is the geometric transformation caused by the perspective (fig 6.12: when the satellite point of view is heavily off-nadir, the landmarks appear distorted and the algorithms struggle to find similarities. This issue can be partially overcome with a targeted data augmentation, but evidences prove that this is not enough for a successful keypoint matching.

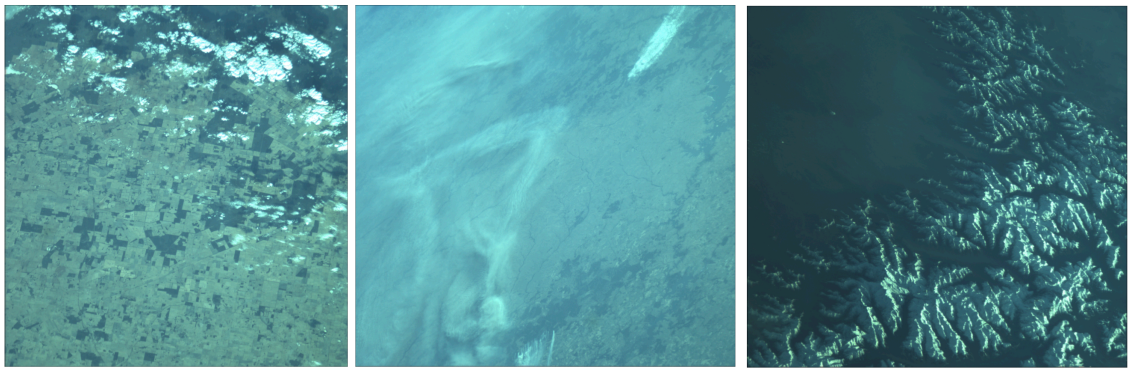


Figure 6.12: Failure scenarios. The first two images are characterized by the presence of clouds, which undermines the success of the algorithms; the 2nd and 3rd, instead, are taken with a large angle, which makes the landmark very distorted.

6.3 Methods comparison

This section presents a more detailed analysis of the performances of the different models implemented, explaining the methods used for the evaluation.

6.3.1 Siamese Networks

The Siamese network is an architecture that well suits the problem at hand, since it has been specifically designed to detect similarities among a pair of input pictures.

To calibrate the model different backbone networks has been tried, pre-trained on object detection with ImageNet. On top of the CNN, a couple of dense layers are used to aggregate the features extracted and reduce the dimension of the output. The configurations that brought some acceptable results are described in 6.1. It is important to observe that without having a ground truth, and being the OPS-SAT dataset limited in size, it is not possible to perform an extensive and objective evaluation of the network performances related to the final application: the one presented in this section refer to the test set left out from the SEM12MS dataset, as explained in Section 4.4.4. From the performances summarised in the table 6.2, we can verify that Mobilenet, which has half of the parameters of VGG, reaches a pretty high accuracy on the test set, while the prediction time is comparable to the one of VGG19. This can be explained by the fact that, in terms of computational time, the bottleneck is the pre-processing applied on the input pictures. In particular, the data augmentation procedure is the main responsible of the high latency of the predictions.

The confusion matrices reported in 6.13, instead, show the number of input pairs that are correctly labeled on the main diagonal, and the mistakes in the remaining cells. Interestingly, they reveal that the most common mistake is to label as similar inputs that are instead different. The kind of errors can be balanced by lowering the distance threshold under which a pair is considered as similar.

Taking into consideration the scores for the positive class, computed from the confusion matrix and reported in tab 6.3, the most precise model is also in this case VGG19 v12, but best trade-off between weight and performances is the one of Mobilenet as already highlighted. The metrics reported are used to perform an evaluation by class, and are mainly used when classes have

Table 6.1: Description of backbone models for the Siamese Neural Network

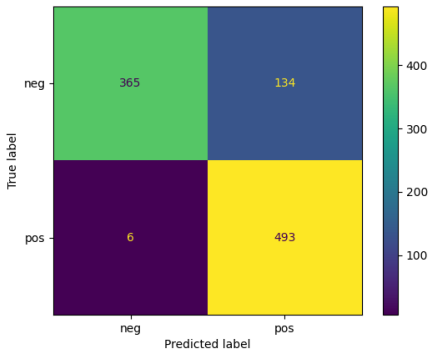
Model	**Description**
Mobilenet	- 2 dense: 100 + 70 - Augmentation = 'medium' - Bach_size = 16 - # Frozen layers = 234
VGG19 v10	- 2 dense: 256 + 128 neurons - Batch_size = 16 - Augmentation ='hard' - Learning rate = 1e-4 - # Frozen layers = 17 - Train dataset size = **2k samples**
VGG19 v11	- 2 dense: 256 + 128 neurons - Batch_size = 16 - Augmentation ='hard' - Learning rate = 1e-4 - #Frozen layers = 17 - Train dataset size = **5.5k samples**
VGG19 v12	- 2 dense:**512 + 256** neurons - Batch_size = **32** - Augmentation ='hard' - Learning rate = 1e-4 - # Frozen layers = 17 - Train dataset size = **5.5k samples**

Table 6.2: Comparison of backbone models for the Siamese Neural Network

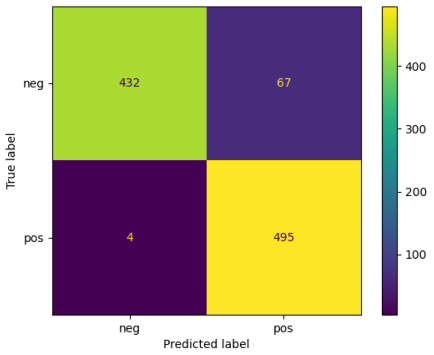
Model	**Size (#params)**	**VAL acc**	**TEST acc**	**Time (GPU) for 1k samples**
Mobilenet	10.5 mln	–	91.08	**2.07min**
VGG19 v10	20.2 mln	0.86	85.97	2.07min
VGG19 v11	20.2 mln	0.89	**92.88**	2.14min
VGG19 v12	20.5 mln	0.91	**92.88**	3.17min

different importance:

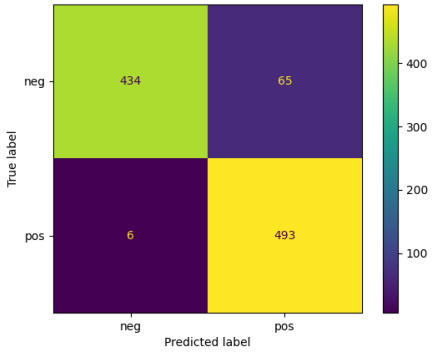
- $Precision(p) = \frac{TP}{TP+FP}$



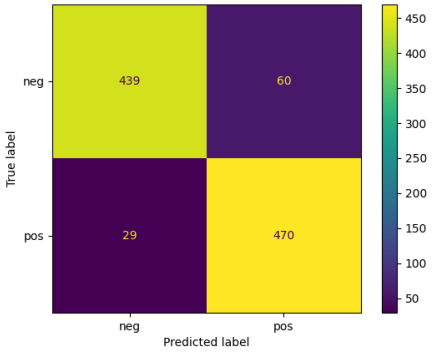
(a) Confusion Matrix for VGG v10



(b) Confusion Matrix for VGG v11



(c) Confusion Matrix for VGG v12



(d) Confusion Matrix for Mobilenet

Figure 6.13: Evaluation of Siamese networks on test dataset

- $Recall(r) = \frac{TP}{TP+TN}$
- $Fmeasure = \frac{2rp}{r+p}$

Table 6.3: Metrics analysis concerning the positive class, i.e. similar input pairs

Model	**Accuracy**	**Precision**	**Recall**	**F Measure**
Mobilenet	91.08	0.89	0.94	0.91
VGG19 v10	85.97	0.79	0.98	0.87
VGG19 v11	**92.88**	0.88	**0.99**	**0.93**
VGG19 v12	**92.88**	**0.99**	**0.99**	**0.93**

The experiments carried out on the whole pipeline, taking into consideration

OPS-SAT images, allow us to make some important considerations based on visual observations of the network outputs. In that case, the network is used to predict the distance between the query and each reference picture. The outputs are then sorted in ascending order, and only the top 10 images are taken into consideration. A trial is considered successful if the matching reference image appears among the top 10 selected. First of all, even though all networks suffered from the domain shift, they were able in some cases to retrieve the correct landmarks among the reference database. Then, it has been observed that applying pre-processing to the input to correct the blue hue did not improve the results; the best option was to match the histogram of the query to each reference image before feeding the pair to the network. Overall, the results can be much improved, but they still show how this approach is a promising path.

6.3.2 Image registration

For what concerns the image registration step, the chosen solution scheme is composed of SIFT, for the extraction of feature vectors, a KNN matcher to couple the obtained keypoints, and RANSAC, which divides the matching points into inliers and outliers.

During the development of the proof of concept, other feature extractors have been taken into account, such as ORB and SuperPoint. However, ORB showed lower performances both in terms of the number of detected keypoints and the Correct Matching Rate (CMR). Similarly, SuperPoint was able to detect very few features inside the input picture, mainly because the pre-trained network under use was optimized to work with natural images. The use of SuperPoint in this context would probably benefit from fine-tuning its parameters on satellite images, which is let as a suggestion for further development of this project. SIFT, instead, produced acceptable results in all the tests performed, although they depended on the previous Image Retrieval step: the features detection algorithm has been applied on all the top K candidate matching images selected by the Siamese Network, and it showed the ability to correctly re-rank them based on the number of keypoint features found. Moreover, the number of features extracted was, in general, large enough to find a set of good matches between the images.

In the Proof of Concept, the metric considered to evaluate the registration process was the *alignment error*, consisting of the computation of the overlapping inaccuracy by means of the Root Mean Squared Error, measured in

pixels:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(rx_i)^2 + (ry_i)^2} \tag{6.1}$$

where rx and ry are the residuals (in pixels) in the x and y directions between the ground truth control points and the transformed points. The image pair is considered to be matched correctly if the RMSE is below a pre-defined threshold. In practice, the correctness of the transformation is more quickly verified by a visual check of the overlapped images.

The computation of the RMSE has been done according to the following steps:

1. The two input pictures with the corresponding matching points are taken as input
2. The transformation ϕ is computed, based on a subset of the inlier tie points (the size of the subset is determined by the minimum number of points required to estimate the considered geometrical transformation, 3 in the case of an affine transformation).
3. The points left out at the previous step are transformed on the reference coordinate system according to ϕ, and the RMSE is computed with respect to the control points.

It is important to observe that if SIFT cannot find enough good keypoints, RANSAC will filter out many of them. As a consequence, computing the RMSE on the inliers set can be very misleading, as the estimation would be based on very few points. For the example reported in the Proof of concept the results obtained are the following:

```
Matches before ransac: 70
Number of inliers: 14
Minimum RMSE: 1.9439693189636955
CMR = 14/70 = 20%
```

In this specific application, where the final goal is to obtain the pointing direction of the spacecraft, an error of few pixels in the registration of the images can be considered as negligible. The CMR, representing the inliers ratio with respect to all the matches found, can be improved with a better calibration of the RANSAC parameters or with better performing feature detector and matching algorithms.

Chapter 7

Conclusions

This Master's Thesis has at its core the presentation of a method for estimating the attitude of a spacecraft, specifically the ESA CubeSat OPS-SAT, starting from the pictures captured by the onboard camera.

The technologies currently in use for estimating satellite attitude require the calibration, weighting, and filtering of multiple sensor inputs and the bespoke tuning of complex estimation algorithms. This makes their adoption often tricky and hardly portable and rises the risk of introducing errors in the recorded telemetries. In particular, when there are errors in the calibration of the sensors or in the recording of the attitude values, such as gaps or misalignment with the timestamps, it becomes very hard and time expensive to retrieve the correct geographical coordinates of the received pictures.

The goal of this thesis was to design and demonstrate a potential solution to this issue, based on Artificial Intelligence and computer vision techniques, able to provide insights on the actual attitude of OPS-SAT at the moment it captured an image, having as input only the picture of interest.

The objective was accomplished through the creation of a pipeline composed by several modules, each of which served as input for the next one and contributed to the visual-based geo-localization of the sensed image and successive estimation of the attitude parameters of the spacecraft.

The process of design and development of this framework followed a series of progressive steps. The first phase was devoted to studying and understanding the problem of attitude estimation, its physical and mathematical background, and the software already implemented by the OPS-SAT team

which is currently in use. Given the specific scope of the application, with few examples of similar projects, and the numerous potential directions, a thorough analysis of the landscape of image-based methods exposed in the literature was then conducted, as well as an in-depth study of most used computer vision techniques.

Different approaches have been taken into account, such as performing object detection on a given set of landmarks, but the consideration of constraints resulted in the selection of the image matching approach aimed at geo-localizing the sensed picture. Once defined a solution scheme, the following phase was required to address the unavailability of a ground truth for OPS-SAT pictures, which could be handled either by applying self-supervised models or by searching for a labeled dataset of satellite images that could reduce at minimum the domain shift. Moreover, since directly performing keypoint matching between the sensed image and the whole reference database showed to be too ambitious, it has been decided to split this process into a first image retrieval step, to filter only the most promising candidate references, and a subsequent keypoint matching for a more precise registration of two images. Finally, the implementation of a proof of concept is used to demonstrate the feasibility of the proposed approach and to show how the different steps can be connected together.

The major contribution of this work is having produced a solid baseline for the development of an offline visual attitude estimator, enabled with machine learning and deep learning techniques, which can be further improved and continuously updated. Its main strength resides, in fact, in the modularity of the structure, that allows to easily replace each building block with a newer version. Furthermore, this demonstrates how promising AI-systems are in the context of satellite operations, and shows how they can challenge the performances of state-of-the-art attitude control systems built on traditional approaches.

7.1 Way forward

As already stated, the solution scheme proposed in the proof of concept has been constructed as a baseline, to show how the problem can be addressed and to demonstrate the whole functioning of the pipeline, from the retrieval of the telemetries describing the state of the satellite at the moment it captured the image, to the obtainment of the new set of quaternions. Therefore, the

main limitation of this framework resides in its performances, in terms of speed as well as robustness, which have significant room for improvement. At this point, the aim would be to replace the current algorithms with more robust and accurate methods at the state-of-the-art.

The thorough analysis of the literature carried out continuously during the development of the project let emerge numerous pathways that can be followed to further improve the accuracy of each step, the robustness regarding the most challenging cases of OPS-SAT pictures, and the efficiency in terms of computational time. The possible improvements are countless, but it is surely worth it to start optimizing the early stages of the pipeline, since their output greatly affects the proper functioning of the models built on top of them. For instance, it would be a good starting point to precisely draw the area of interest, by propagating the TLE to know the direction of the satellite pass. Concerning the image retrieval algorithm, which is the key step for a successful operation, a promising solution to cope with the case that geographical landmarks get buried in the complex and cluttered backgrounds is the attention mechanism. This techniques is used, in fact, to focus on certain parts or features of the input image, while ignoring irrelevant or noisy information. About keypoint matching techniques, instead, two algorithms that might provide remarkable results if properly trained are, as already mentioned, SuperGlue [26] and LoFTR [29] networks.

Last but not least, the most exciting and ideal target is the creation of an autonomous attitude estimator that could operate on board the satellite. The main barrier in this sense is represented by the limited computing resources available on the spacecraft, and in particular way by the storage capacity, which prevents us from uploading the reference database. Even on the ground, the construction of the set of reference images and the download of the pictures covering the area of interest represent the main bottleneck of the pipeline, both from the storage space perspective and the time required.

Nevertheless, this application offers extensive and fascinating research opportunities, which will foster the development of autonomous space missions and strengthen the partnership between Artificial Intelligence and Space Operations.

Bibliography

[1] "ops-sat (operations nanosatellite)". *eoPortal*.

[2] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features (surf). *Computer vision and image understanding*, 110(3):346–359, 2008.

[3] Michael Calonder, Vincent Lepetit, Mustafa Ozuysal, Tomasz Trzcinski, Christoph Strecha, and Pascal Fua. Brief: Computing a local binary descriptor very fast. *IEEE transactions on pattern analysis and machine intelligence*, 34(7):1281–1298, 2011.

[4] Wei Chen, Yu Liu, Weiping Wang, Erwin Bakker, Theodoros Georgiou, Paul Fieguth, Li Liu, and Michael S. Lew. Deep learning for instance retrieval: A survey, 2021.

[5] Julien Cornebise, Ivan Orsolic, and Freddie Kalaitzis. Open high-resolution satellite imagery: The worldstrat dataset – with application to super-resolution. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022.

[6] Feng Wang Cuiyin Liu, Jishang Xu. "a review of keypoints detection and feature description in image registration", 2021.

[7] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[8] Vrushank H Gandhi, Sandip R Panchal, and PG Student. Feature based image registration techniques: An introductory survey. *International Journal of Engineering Development and Research (IJEDR)*, 2(1):368–375, 2014.

[9] Vitor CF Gomes, Gilberto R Queiroz, and Karine R Ferreira. An overview of platforms for big earth observation data management and analysis. *Remote Sensing*, 12(8):1253, 2020.

[10] Noel Gorelick, Matt Hancher, Mike Dixon, Simon Ilyushchenko, David Thau, and Rebecca Moore. Google earth engine: Planetary-scale geospatial analysis for everyone. *Remote sensing of Environment*, 202:18–27, 2017.

[11] https://content.sifted.eu/wp-content/uploads/2023/01/19081913/Spacetech 1.pdf. "spacetech: the big business of space on earth". *Sifted Reports (Deloitte)*, 2023.

[12] https://sentinels.copernicus.eu/web/sentinel/user-guides/sentinel 2-msi/. Sentinel 2 data format.

[13] https://www.copernicus.eu/en/about copernicus. European union "about copernicus".

[14] https://www.spacefoundation.org/2022/07/27/ the-space-report 2022-q2/. "space foundation releases the space report 2022 q2 showing growth of global space economy". *Space Foundation*, 2022.

[15] HeeJae Jun, Byungsoo Ko, Youngjoon Kim, Insik Kim, and Jongtack Kim. Combination of multiple global descriptors for image retrieval. *arXiv preprint arXiv:1903.10663*, 2019.

[16] Shoei Koizumi, Yuhei Kikuya, K. Sasaki, Yuto Masuda, Yohei Iwasaki, Kei Watanabe, Yoichi Yatsu, and Saburo Matsunaga. Development of attitude sensor using deep learning. 2018.

[17] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017.

[18] Jacqueline Le Moigne. Introduction to remote sensing image registration. In *2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, pages 2565–2568. IEEE, 2017.

[19] Yansheng Li, Jiayi Ma, and Yongjun Zhang. Image retrieval from remote sensing big data: A survey. *Information Fusion*, 67:94–115, 2021.

[20] David G Lowe. Object recognition from local scale-invariant features. volume 2, pages 1150–1157. Ieee, Proceedings of the seventh IEEE international conference on computer vision, 1999.

[21] Jiayi Ma, Xingyu Jiang, Aoxiang Fan, Junjun Jiang, and Junchi Yan. Image matching from handcrafted to deep features: A survey. *International Journal of Computer Vision*, 129:23–79, 2021.

[22] Indranil Misra, Mukesh Kumar Rohil, S Manthira Moorthi, and Debajyoti Dhar. Feature based remote sensing image registration techniques: a comprehensive and comparative review. *International Journal of Remote Sensing*, 43(12):4477–4516, 2022.

[23] Sourabh Paul and Umesh C Pati. A comprehensive review on remote

sensing image registration. *International Journal of Remote Sensing*, 42(14):5396–5432, 2021.

[24] Angel Porras-Hermoso, Javier Cubas, and Santiago Pindado. On the satellite attitude determination using simple environmental models and sensor data. In *Journal of Physics: Conference Series*, volume 2090, page 012116. IOP Publishing, 2021.

[25] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: An efficient alternative to sift or surf. In *2011 International conference on computer vision*, pages 2564–2571. Ieee, 2011.

[26] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. SuperGlue: Learning feature matching with graph neural networks. In *CVPR*, 2020.

[27] Schmitt, Qiu M., Hughes a nd L.H., Cl., Zhu, and X.X. Sen12ms a curated dataset of georeferenced multi-spectral sentinel-1/2 imagery for deep learning and data fusion. 2019.

[28] Matthias Schramm, Edzer Pebesma, Milutin Milenković, Luca Foresta, Jeroen Dries, Alexander Jacob, Wolfgang Wagner, Matthias Mohr, Markus Neteler, Miha Kadunc, et al. The openeo api–harmonising the use of earth observation cloud services using virtual data cube functionalities. *Remote Sensing*, 13(6):1125, 2021.

[29] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. Loftr: Detector-free local feature matching with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8922–8931, 2021.

[30] Richard Szeliski. Computer vision: Algorithms and applications. *https://szeliski.org/Book/*.

[31] Yue Wu, Jun-Wei Liu, Chen-Zhuo Zhu, Zhuang-Fei Bai, Qi-Guang Miao, Wen-Ping Ma, and Mao-Guo Gong. Computational intelligence in remote sensing image registration: A survey. *International Journal of Automation and Computing*, 18:1–17, 2021.

[32] Sergey Zagoruyko and Nikos Komodakis. Learning to compare image patches via convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4353–4361, 2015.