West University of Timisoara

Faculty of Mathematics and Computer Science

Study Program:
Compter Science in English

Bachelor Thesis

COORDINATOR:

Associate Prof. Marc Eduard Frîncu

GRADUATE:

Maria Minerva Vonica

Timişoara
2019

West University of Timişoara

Faculty of Mathematics and Computer Science

Study Program:
Compter Science in English

Image alignment and glacier change prediction on satellite images using artificial inteligence

COORDINATOR:

Associate Prof. Marc Eduard Frîncu

GRADUATE:

Maria Minerva Vonica

Timişoara
2019

Abstract

Climate change has become one of the biggest threats the world has to deal with in the near future, and yet, there is still not enough information to completely understand the phenomenon, how fast it grows and the chain reactions it will imply given that not enough measures are taken in time. Given that glaciers are the most sensitive indicators of climate change, a solution to widening the information field about the phenomenon is implementing a model of prediction on satellite images for analyzing the changes in snow coverage on glaciers during a period of over a decade and all around the globe.

Building this application had the goal to create a tool for gathering valuable data on how fast the climate change happens by analysing the downfall of snow coverage on glaciers during a period of time and predicting by how much they will continue to shrink in the near future. Since more information means a better understanding of the phenomenon, we also made simple to use and easy to access for anyone who wants to get involved and make a contribution.

This paper contains information on the area of analysis, the application's structure and processing mechanics. The idea is to give the users an option to easily access and download satellite data to work on, then process it by calculating the NDSI (Normalized Difference Snow Index) in order to highlight the snow on the glaciers, generating a number representing the snow ratio in each scene. By having a sample of images of a glacier taken during a big number of years and analyzing its changes in snowfall, we can see then analyze and predict the future changes over the next period, based on the snow coverage ratio for each image. Optical flow and image substracting will be applied to highlight the changes visual wise.

Contents

List Of Figures **iii**

List Of Tables **v**

1 Introduction **1**

1.1 Motivation and goals 2

1.2 The context of this thesis 2

1.3 Our Contribution 3

2 Data set Structure **5**

2.1 Satellite Imagery 5

2.2 Landsat 8 5

2.2.1 Worldwide Reference System 6

2.2.2 Operational Land Imager (OLI) 7

2.2.3 Thermal Infrared Sensor (TIRS) 7

2.3 Resolutions 7

2.4 Data set Collection 9

2.4.1 Collection process 9

2.4.2 Landsat Image Collection 9

2.5 Data set Images 9

2.5.1 Landsat 8 Bands 9

2.5.2 Green Band 10

2.5.3 SWIR1 (Infrared) Band 11

2.5.4 The Normalized-Difference Snow Index 11

2.5.5 Landsat scene name convention 13

2.6 Programming Environment: PyCharm 13

2.7 Python Programming Language . . . 13
2.8 Git . . . 14
2.9 Libraries . . . 14
2.9.1 Geospatial Data Abstraction Library (GDAL) . . . 14
2.9.2 NumPy . . . 14
2.9.3 OpenCV . . . 15
2.9.4 sat-search . . . 15

3 The application **16**
3.1 Functional description . . . 16
3.1.1 Satellite Data Set Building . . . 16
3.1.2 Data Set Information Extraction . . . 20
3.2 Pre-processing . . . 21
3.3 Processing . . . 22
3.3.1 The Normalized-Difference Snow Index . . . 22
3.3.2 Image Alignment . . . 23
3.3.3 Diference . . . 25
3.3.4 Movement . . . 26
3.4 The user interface and guide . . . 27
3.4.1 Download GUI . . . 27
3.4.2 Process GUI . . . 28
3.4.3 Display GUI . . . 28
3.5 Difference and Movement GUI . . . 28
3.6 Main use cases . . . 33

4 Design and implementation **34**
4.1 Download . . . 34
4.2 Alignment and NDSI . . . 36
4.2.1 Alignment . . . 36
4.2.2 NDSI . . . 41
4.3 Display and ARIMA prediction . . . 44
4.3.1 Data set normalization . . . 44
4.3.2 Data set plotting . . . 44
4.3.3 Data set predicting . . . 45

4.3.4 Difference and Movement images 45

5 Performance Evaluation **48**

6 Conclusions **51**

6.1 Future Development 52

Bibliography **52**

A Glossary **58**

A.1 Acronyms 58

List of Figures

1.1 Profile data on application run, detailed. 4

2.1 Landsat 8 satellite [18]. 6
2.2 Worldwide Reference System land coverage [19]. 6
2.3 OLI wavelength comparison between Landsat 7 and 8 [14]. 7
2.4 Landsat 8 spectral resolution [21]. 8
2.5 Landsat 8 Band 3 (Glacier (ID) AT4J143FA006 at (10.803, 47.007) coordinates. [24] 10
2.6 Landsat 8 Band 6 (Glacier (ID) AT4J143FA006 at (10.803, 47.007) coordinates. [25] 11
2.7 NDSI and contrast between green and SWIR1 bands. 12

3.1 Earth Explorer [44]. 17
3.2 Search Inventory interface (NSIDC) [48]. 18
3.3 Extract Selected Regions interface (NSIDC) [**?**]. 19
3.4 Download folder hierarchy. 20
3.5 Metadata file. 21
3.6 Output folder hierarchy. 22
3.7 NDSI and contrast between green and SWIR1 bands. 23
3.8 Two unaligned Landsat 8 scenes, overlapped. 24
3.9 Zoomed in two unaligned Landsat 8 scenes, overlapped. 24
3.10 Two aligned Landsat 8 scenes, overlapped. 24
3.11 Zoomed in two aligned Landsat 8 scenes, overlapped. 25
3.12 Difference for object state. 26
3.13 Difference for object movement. 26
3.14 Download CLI and GUI diagram. 29

3.15 Process CLI and GUI diagram. 30
3.16 Display CLI and GUI interactive prediction plot. 31
3.17 Prediction interactive plot CLI and GUI. 32
3.18 Main use case of the application. The user can choose between downloading, processing or displaying and predicting data. Prediction can only be made if display is chosen, and the interactive plot will pop-up. 33

4.1 Download process class diagram. 35
4.2 Matches between the reference image (left, green) and image to be aligned (rigH, SWIR1). 38
4.3 Image after warping the affine matrix on the SWIR1 band. 38
4.4 Matches between the reference image (left, green) and image to be aligned (right, SWIR1), after box feature find. 39
4.5 Matches between the reference image (left, green) and image to be aligned (right, SWIR1), before prune matches improvement. 40
4.6 Matches between the reference image (left, green) and image to be aligned (right, SWIR1), after prune matches improvement. 40
4.7 The aligned image after applying all four features. 41
4.8 The NDSI class diagram.. 42
4.9 The alignment and NDSI processing for a path and row directory class diagram. 43
4.10 Plot of glacier AR1I0111B128, with coordinates(-70.341, -36.785), on path and row 232, 086. 44
4.11 Picking two NDSI images to create their difference and movement images. 46
4.12 The difference and movement images for the selected points on the plot. . 47

5.1 Las Heras Department, Provincia Mendoza, Argentina (AR1J11132208, 233, 82). 48
5.2 Las Heras Department, Provincia Mendoza, Argentina (AR1J11132208, 232, 83). 49
5.3 Panjshir, Afganistan (AF5Q112C0242, 152, 32). 49
5.4 Las Heras Department, Provincia Mendoza, Argentina (AR1J11132208, 232, 82). 50

List of Tables

2.1 Green Band Specifications [24, 13]. 10
2.2 SWIR1 Band Specifications [26, 13]. 11
2.3 Normalized-Difference Snow Index variables. 12

4.1 Prediction error formula. 45

A.1 Acronyms table 58

Chapter 1

Introduction

Since the creation of the first Earth observation satellites, a gate to new opportunities of better understanding our planet and its behaviour has opened. The intent of the Earth observation programs is to monitor changes such as weather behavior, climate change, astronomical photography, vegetation states, droughts, earthquakes or other natural disasters, such that by analyzing the collected images, patterns which lead to understanding the cause of the behaviour are found. In order to be able to have a clear understanding of those behaviours, massive amounts of data need to be collected, organised and processed. With the large number of observation satellites orbiting the atmosphere and collecting massive amounts of data daily, deep analysis of the data sets must be done.

Nowadays, one of the most important problems on a global scale is climate change, which threatens to definitely change the human life as it is known today. The changes in weather patterns, sea levels, weather events and greenhouse effects are more and more common, which will lead to permanent damage to the environment, society and human life overall.

Given that the tools of understanding changes over the past years have been created with the launching of the first observation satellites in the orbit, all these changes can be tracked and predicted, so the dangerous effects of climate change and other disastrous events can be lowered. We have the tools to understand events which happened in the past, so we can create a safer, more balanced, and predictable future.

1.1 Motivation and goals

The motivation of elaborating this thesis is based on the context of the impact the climate change will have on life if not enough measures are taken and there is not enough understanding of it, as well as the availability of massive amounts of data which consists of satellite images from almost a decade back (Landsat 8 launched in 2013), and a world glacier inventory dating form 2012. The main reasons for elaborating a thesis on this theme are:

- The threats the climate change actually imply if not enough measures are taken as soon as possible;
- The possibility to better understand this phenomenon by analyzing satellite images which contain useful information;
- The availability of massive amounts of data which consists of satellite images from almost a decade back (Landsat8 launched in February 2013).

The **goal** of the thesis is to analyze satellite images focusing on changes in glaciers and snow in order to get a better overview of the changes in the environment in the last decade.

1.2 The context of this thesis

Climate change has become one of the biggest problems in the past years, human activities being one of the main factors by inducing global warming [1]. Due to heavy industrialisation of the society since the Industrial Revolution around the VII century, the levels of carbon dioxide in the atmosphere have grown beyond the capacity of natural elimination, creating the greenhouse effect.

The greenhouse effect is the process in which the radiation from a planet's atmosphere warms the planet surface above that normal temperature [2]. The main causes of this change are human activities such as combustion of fossil fuels (coal, oil, natural gas), deforestation, soil erosion and agriculture [3]. If the change continues at this rate, there is an estimation that the average surface temperature will surpass °3 C this century [4]. Though it may not seem a lot, there is a scenario created over 30 years into the future explaining what this increase actually means: from 2020 to 2030 the temperature is set

to rise above °1.6C; by this report, it is estimated that the average temperature will increase with °3C around 2050, in which case even if gas emissions are stopped, there will be another rise of at least °1C [5].

One of the ways of creating an overview of the last decades of climate change is by analysing the change in the glaciers, since they are among the most sensitive indicators of climate change [6]. The retreat of a glacier is caused by an increase of the temperature and less snowfall, a process which has been accelerated recently, due to the global warming [7].

In the context of these changes, which are already beginning to impact the society and environment, I strongly believe that we have to take advantage of every opportunity to better understand and predict these changes, so proper preparations and actions could be taken as soon as possible.

1.3 Our Contribution

This paper uses image analysis techniques paired with artificial intelligent algorithms to create a time series analysis on the snow coverage of glaciers around the globe, while using methods which prove to be highly performant and accurate:

- Alignment of images with accuracy rate of **98%** percent for the successful processed ones;
- Successful alignment of approximately **85%** images for a Landsat 8 iamge;
- Snow coverage prediction with approximate **20%** mean error.

. The results were achieved by calculating the NDSI for extracting snow pixels from a Landsat 8 image on a set of images for a glacier which has been split into path and row coordinates, paired up with alignment of the resulting images for displaying their differences over the years. The Normalized Difference Snow Index (NDSI) was obtained by calculating a formula which uses the green and SWIR1 bands from a scene and applying a threshold in order detect snow. The method of snow extraction was chosen to be NDSI calculation due to snow's high reflectance in the visible spectrum and high absorption in the short-wave infrared one.

After the snow coverage has been extracted, the NDSI, green and SWIR1 images are aligned based on a reference image, such that creation of an image which highlights the

presence or absence of snow between two scenes, and one with highlight the movement can be created without pixel discrepancy errors. By reducing the misalignment to almost 0 shift, the results are reliable. Prediction of snow coverage is done by creating a data set containing snow ratio information extracted from the NDSI images, which is normalized (outliers are removed) and split into a train section and a test one, also having future predictions. The results are plotted for easier data interpretation and they highlight the pattern of snow coverage in the past years, as well as future ones, with a small error percent (approximately 20%). An exact estimation of the error rate cannot be given due to the fact prediction is highly dependent on NDSI values, which are based on a high number of glacier variables such as the quality of image, the seasonal variation and other factors.

The thesis application can be used as an command line interface, as well as a graphical user one, in order to include a user group of both specialized and non-specialized in the field. The processing time and efficiency are optimized such that the only intensive process remains reading, writing and downloading the data. In order to keep track on execution time and cost, we implemented profiling data for each process, which can be viewed with **snakeviz**, a browser based graphical viewer for the output of Python's cProfile module [58].

The following result from profile data shows that reading and writing of images takes the most time by far, compared to all the other processes, which is normal since the images have a high resolution (8000x8000 tiff images of 16bit depth).

ncalls	tottime	percall	cumtime	percall	
4	8.756	2.189	8.756	2.189	~:0(<imread>)
2	4.694	2.347	5.987	2.994	ndsi.py:16(calculate_NDSI)
3	3.191	1.064	3.191	1.064	~:0(<imwrite>)
8	1.677	0.2097	1.677	0.2097	~:0(<method 'astype' of 'numpy.ndarray' objects>)
4	1.188	0.2971	1.188	0.2971	~:0(<method 'compute' of 'cv2.Feature2D' objects>)
256	1.149	0.004488	1.149	0.004488	~:0(<method 'detect' of 'cv2.Feature2D' objects>)
5	0.9674	0.1935	0.9674	0.1935	~:0(<normalize>)
2	0.6498	0.3249	1.034	0.517	alignment_ORB.py:173(downsample)
1	0.4176	0.4176	0.6794	0.6794	ndsi.py:53(get_snow_image)
3	0.359	0.1197	0.359	0.1197	~:0(<warpAffine>)
1	0.2618	0.2618	0.2618	0.2618	~:0(<method 'copy' of 'numpy.ndarray' objects>)
1	0.1984	0.1984	0.1984	0.1984	~:0(<drawMatches>)
1	0.1816	0.1816	0.1816	0.1816	ndsi.py:77(get_snow_pixels_ratio)
1	0.04051	0.04051	6.891	6.891	alignment_ORB.py:62(ndsi)
4	0.01321	0.003302	2.351	0.5877	alignment_ORB.py:196(boxedDetectAndCompute)
1	0.006004	0.006004	4.798	4.798	alignment_ORB.py:123(align)
1	0.002494	0.002494	2.912	2.912	alignment_ORB.py:305(align)
2	0.000545	0.0002725	0.000545	0.0002725	~:0(<built-in method builtins.print>)
1	0.00048	0.00048	3.314	3.314	alignment_ORB.py:135(write)

Figure 1.1: Profile data on application run, detailed.

.

Chapter 2

Data set Structure

2.1 Satellite Imagery

Satellite images are images of Earth or other planets collected by imaging satellites operated by governments and businesses around the world [8]. The first collected images date from 1959, when the Explorer 6 was launched, followed by the Landsat program in 1972 which still functions to this date, holding the position of the largest program for image acquisition of Earth from space [8].

2.2 Landsat 8

The data used in this thesis is collected by the Landsat 8 Earth observing satellite, launched on 13 February 2013, as a collaboration between NASA and USGS [11]. NASA took over the construction of the satellite, while USGS focused on maintaining it by calibration, generation of data scenes and storing them. The Landsat program goal is to collect and store 30 meters spatial resolution multispectral image data affording seasonal coverage of the global landmasses for a period of at least 5 years, while also ensuring that the data is consistent during this period, in terms of geometry, calibration, coverage characteristics, output product quality and data availability to ensure that studies over time can be done [9][11]. The satellite products are free to use, stored in the Landsat Archive.

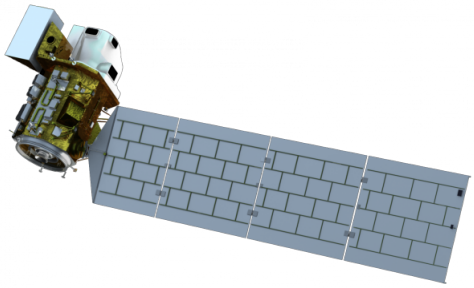

Figure 2.1: Landsat 8 satellite [18].

2.2.1 Worldwide Reference System

The Landsat data uses the WRS2 coordinate system, which uses path and row variables, which cover all the land surface of the Earth. The row equals the latitudinal center of a frame, while the path moves along the path and scans the land below [19]. The combination of the two creates the center of a Landsat scene which is identified as unique based on this notation. (more on Landsat naming convention). The rows and paths of the world land coverage are shown below:

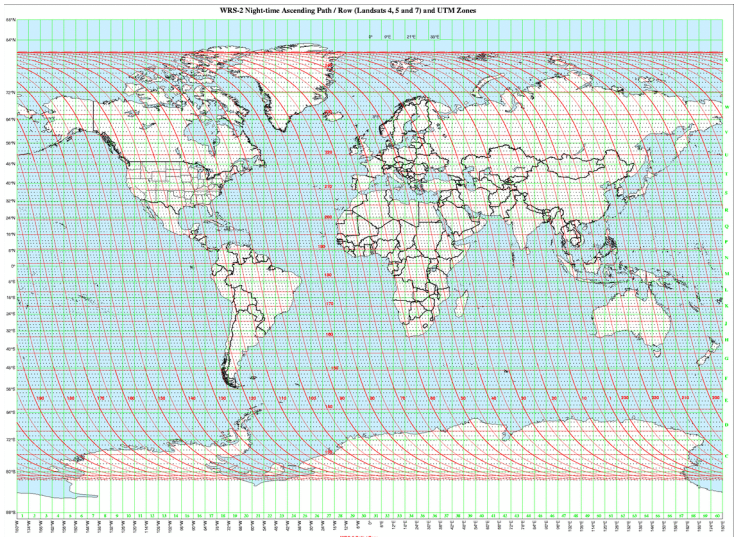

Figure 2.2: Worldwide Reference System land coverage [19].

The collection used for image analysis in the thesis is Landsat Collection 1 Level-1. Landsat 8 is the latest and most technologically advanced satellite in the Landsat program. It has been equipped with two sensors, OLI and TIRS.

2.2.2 Operational Land Imager (OLI)

The OLI sensor is a push-broom sensor with a four-mirror telescope and a 12-bit quantization. It collects data from the visible, near infrared and short wave infrared spectral bands, as well as panchromatic band [14]. The sensor can capture areas of 185 kilometers [15], and it is constructed such that the resolution is good enough to distinguish objects like cities, farms, small islands and other objects which were not seen as detailed in the previous Landsat satellites, due to the different spatial resolution, as seen in the below chart (Figure 2.2):

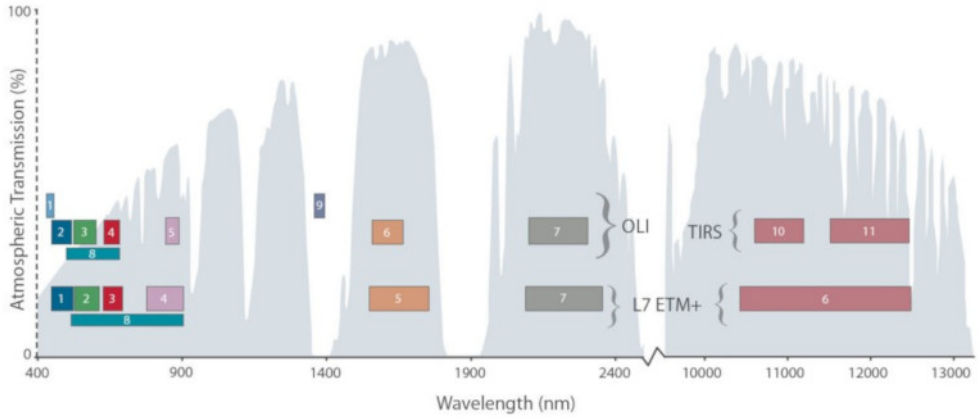

Figure 2.3: OLI wavelength comparison between Landsat 7 and 8 [14].

2.2.3 Thermal Infrared Sensor (TIRS)

The TIRS works by detecting the wavelengths of light emitted by the planet, which have a varying intensity depending on the temperature of its surface. The mechanics of measuring the wavelengths use quantum physics, and are stored in two thermal bands (band 10 and band 11) [17]. Since in the thesis only band 3 and 6 will be used for processing, the thermal bands 10 and 11 will not be necessary.

2.3 Resolutions

When dealing with satellite image data, the resolution has an important role in its value. There are four types of resolutions: *radiometric, temporal, spectral, spatial.*

Radiometric resolution The radiometric resolution represents the ability of a digital sensor to distinguish between grey-scale values [10]. The collected images are either 8-bit, 11-bit, 12-bit or 16-bit, depending on the sensor. Having large values of bit depth increase the number of features an image has and the ability to spot them.

Temporal resolution The temporal resolution represents the frames of image collection, measured in temporal attributes such as hours or days. The Landsat 8 observing satellite captures an average of 700 images per day [11], and a 16-day repeat circle, referenced to the Worldwide Reference System-2 (WRS2) [16].

Spectral resolution The spectral resolution represents the number of spectral bands in which the sensor can collect reflected radiance [12] [13]. The data collected for the elaboration of this thesis is from the Landsat 8 satellite, which has a multispectral resolution, consisting of 11 bands, as described below:

	Bands	Wavelength (micrometers)	Resolution (meters)
Landsat 8 Operational Land Imager (OLI) and Thermal Infrared Sensor (TIRS) Launched February 11, 2013	Band 1 - Coastal aerosol	0.43 - 0.45	30
	Band 2 - Blue	0.45 - 0.51	30
	Band 3 - Green	0.53 - 0.59	30
	Band 4 - Red	0.64 - 0.67	30
	Band 5 - Near Infrared (NIR)	0.85 - 0.88	30
	Band 6 - SWIR 1	1.57 - 1.65	30
	Band 7 - SWIR 2	2.11 - 2.29	30
	Band 8 - Panchromatic	0.50 - 0.68	15
	Band 9 - Cirrus	1.36 - 1.38	30
	Band 10 - Thermal Infrared (TIRS) 1	10.60 - 11.19	100
	Band 11 - Thermal Infrared (TIRS) 2	11.50 - 12.51	100

Figure 2.4: Landsat 8 spectral resolution [21].

Spacial resolution The spatial resolution is evaluated by how much of the image a pixel represents, which gives its detail. An approximate Landsat 8 scene size is 170 km north-south by 183 km east-west [15]. The Landsat 8 satellite is equipped with two sensors, the Operational Land Imager (OLI) and Thermal Infrared Sensor (TIRS). These two sensors provide coverage at a spatial resolution of [14]:

- 15 meters: panchromatic;
- 30 meters: visible, NIR, SWIR;
- 100 meters: thermal.

2.4 Data set Collection

2.4.1 Collection process

The satellite collects the images by moving across the path and scanning the land under it, while the instrument signals are transmitted to Earth and correlated with telemetry ephemeris data to form individual framed images. During this process, the continuous data are segmented into individual frames of data which represent the scenes [19].

2.4.2 Landsat Image Collection

The image collection used in this thesis is the Landsat Collection 1 Level-1, which is generated by the OLI and TIRS sensors of the Landsat 8 satellite. The implementation of the Collection represents a significant change in the management of the Landsat Archive by ensuring consistent quality through time and across instruments, along with additional changes like meta data and file names [20].

Collection 1 Tier 1 The aim of the collection tiers is to ensure that the data gathered from the satellite is of good quality and easy identification, such that it can be used for time-series pixel-analysis [20]. The Landsat8 scenes are processed and available for download in an average of 5-6 hours; after being reprocessed with definite ephemeris, updated bumper mode parameters and refined TIRS parameters, the products are transitioned to Tier 1 or 2, a process which usually takes between 14 and 26 days [20]. The scenes are divided into Tier 1 and Tier 2 based on the quality for the product. The images which are used in this thesis are Tier 1 images, and were chosen because of the higher quality and noted to be time-series analysis safe.

2.5 Data set Images

2.5.1 Landsat 8 Bands

The Landsat 8 bands are TIFF images with a resolution between 7000 and 8000 pixels, with 1 pixel representing 30 meters. One scene covers about 210 km and it is updated at every 16 days. The bit-depth of the images is 16 and are captured in gray-scale. For the elaboration of the thesis, only the third and sixth bands will be used in the calculation

of the Normalized Difference Snow Index, so they will be detailed below with examples of images for each.

2.5.2 Green Band

Green band image specifications:

Wave-length	0.53- 0.59 micrometers
Spacial Resolution	30 meters
Depth	16-bit
Resolution	aproximately 7500x7500 pixels
Format	Gray-scale

Table 2.1: Green Band Specifications [24, 13].

Due to its wave-length, the green band is one of the three bands from the visible spectrum, along the blue and red band [26]. The following image is an example of a green band taken from the AT4J143FA006 glacier:

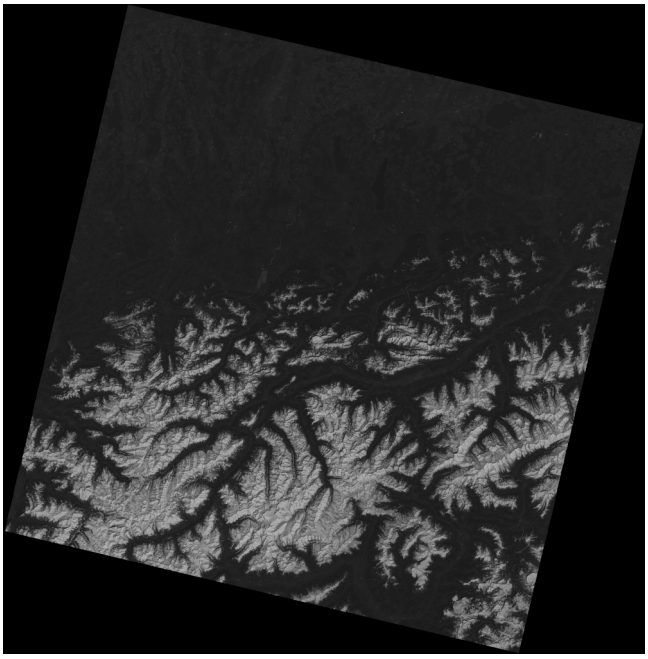

Figure 2.5: Landsat 8 Band 3 (Glacier (ID) AT4J143FA006 at (10.803, 47.007) coordinates. [24]

2.5.3 SWIR1 (Infrared) Band

SWIR1 band image specifications:

Wave-length	1.57 - 1.65 micrometers
Spacial Resolution	30 meters
Depth	16-bit
Resolution	aproximately 7500x7500 pixels
Format	Gray-scale

Table 2.2: SWIR1 Band Specifications [26, 13].

The SWIR1 band covers one part of the shortwave infrared spectrum, invisible to the human eye. It enhances geology objects such as rocks and soils which look similar in other bands [26], making them visible and easy to analyse

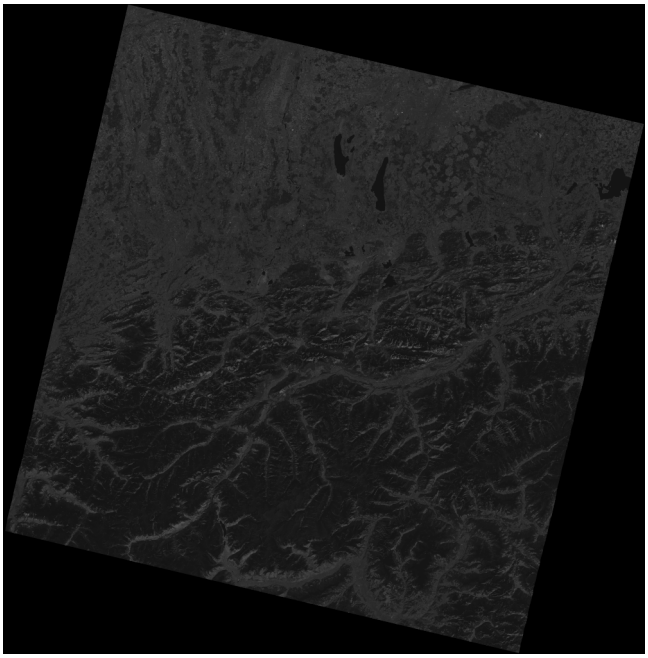

Figure 2.6: Landsat 8 Band 6 (Glacier (ID) AT4J143FA006 at (10.803, 47.007) coordinates. [25]

2.5.4 The Normalized-Difference Snow Index

The NDSI represents the normalized difference of the two described bands, the green band in the visible spectrum and the SWIR1 one in the shortwave infrared spectrum.

By calculating the NDSI between the two bands, the result will **enhance snow in a Landsat scene**, due to snow's high reflectance in the visible spectrum (0.66 mm) and high absorption in the short-wave infrared one (1.6 mm) [27, 28]. The mapping is done by computing the following formula:

$$ndsi = \frac{green - swir1}{green + swir1}[28]$$

ndsi	result which holds the normalized image
green	third band of the Landsat 8 satellite
swir1	sixth band of the Landsat 8 satellite

Table 2.3: Normalized-Difference Snow Index variables.

The NDSI is used at mapping the glaciers from the collected Landsat 8 images, so that the creating the data set consisting of snow pixels over total pixels is accurate. For **snow mapping**, the NDSI is applied on the two images with a **threshold of contrast** so the gray-scale will be just black and white, meaning glacier or no glacier pixel. This is done so that when processing the images on time-series, the values are easier to parse and predict. By applying the formula on the Figures 2.6 and 2.7 and a threshold of contrast of 0.5, the following result is created:

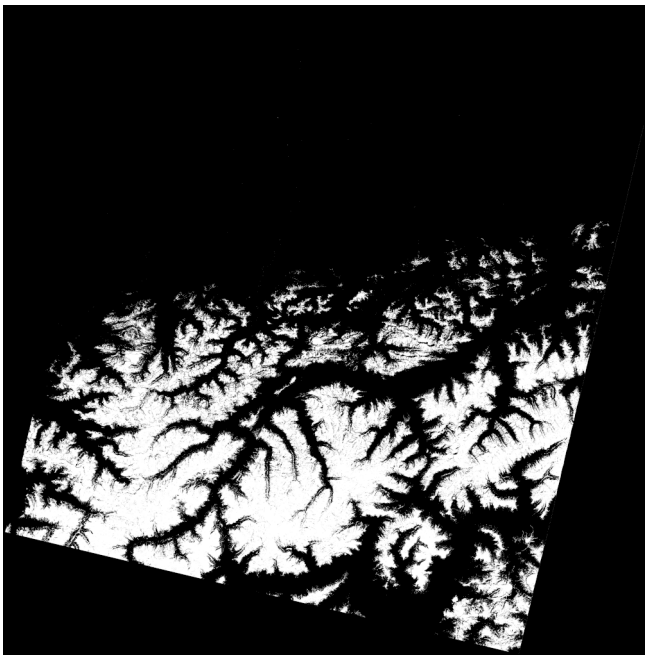

Figure 2.7: NDSI and contrast between green and SWIR1 bands.

2.5.5 Landsat scene name convention

Each Landsat scene has an unique name which holds general information to analyze it at a high level without the need of processing or opening the image beforehand. The identifier for the Collection 1 Level 1 has the following general name:

LXSPPPRRRYYYYDDDGSIVV [43]

- **LSX**: The firs three letters describe what type of satellite and sensor were used;
- **PPPRRR**: The following six letters describe where the scene is located by the WRS2 path and row location
- **YYYYDDD**: The next seven letters represent when the scene was acquired, using the Julian calendar by counting the number of days passed starting January 1 [43];
- **GSIVV**: The last five letters describe how the data was received from its ground stations, which are data capture facilities to obtain transmitted satellite data [43].

2.6 Programming Environment: PyCharm

The application of the thesis has been written in the PyCharm IDE, which is specialized in Python programming language. The IDE ensures clean code writing by analyzing it, code debugging implemented in the graphical user interface, safe testing by adding unit test support and supports revision control with systems like Git, which has been used for version control of the writing of the application. The IDE is cross-platform with Windows, mac OS and Linux [29] The environment has been chosen for the development of the application because it includes features such as code assistance, easy python code re-factoring and revision control with Git.

2.7 Python Programming Language

Python is an interpreted, high-level, general-purpose programming language, created by Guido van Rossum and first released in 1991 [30] The goal of python is simplicity, so that the programmers can achieve logical constructs in their projects at whichever scale, supporting this by being a high-level language, having an implicit garbage collector which

takes care of the memory and granting access to many open-source libraries which can be used to as a puzzle to create something. Due to its flexibility and simplicity, Python has been chosen as the main and only language the application is written in.

2.8 Git

Git is a free, open-source distributed version-control system for tracking changes in the source code during software development, created by Linus Torvalds in 2005 [35][32]. Its goal is to ensure a good work-flow during the creation of a project by keeping track of any chances in all the files of the project during its development, as well as fast and easy usage, data integrity[35] [33] and support for distributed, non-linear work-flows [34] [35]. Git works by flagging a directory as a repository of the project, therefore keeping track of each file and its history in that location, independent of network access or a central server [35]. It, therefore, creates a more structured project development, making it easier to implement different features without breaking the stable version, go back and change things which were not supposed to be released, maintain the released versions and fix bugs. Using Git to keep version control of the application has proven very efficient and time-saving in most situations, ensuring easy fixes, trace-backing, scaling and feature adding.

2.9 Libraries

2.9.1 Geospatial Data Abstraction Library (GDAL)

GDAL is an open-source translator library for raster and vector geo-spatial data formats [38], presenting a single abstract data model to the calling application for all supported formats, while also providing strong command line interface utilities for data translation and processing [39]. In the application, GDAL is used as an image reading tool as an alternative to the OpenCV one.

2.9.2 NumPy

NumPy is the fundamental package for scientific computing with Python, containing powerful tools for N-dimensional arrays handling [40] such as optimized matrix algebra,

easy conversion from images to matrices for pixel-level algebra, which is used in the application for processing the TIFF data as matrices rather than images. NumPy was chosen as the mathematical handler of the application for its capabilities of handling big data files by using high-level mathematical functions to ensure a trustworthy file process, therefore ensuring a better performance [41].

2.9.3 OpenCV

OpenCV is a library of programming functions mainly aimed at real-time computer vision, supporting deep-learning. The initial goal of OpenCV was to handle Intel Research intensive CPU-applications by providing open and optimized code for basic vision infrastructure, and common infrastructure so developers could build easier to read and transferable code [42]. The library was used in the application for image alignment and display working with the TIFF images collected from the satellite, as well as image alignment and difference and movement image creation.

2.9.4 sat-search

Sat-search is a Python 3 library and a command line tool for discovering and downloading publicly available satellite imagery using a conformant API [56]. It is used in our application for searching and downloading Landsat 8 images from the archive.

Chapter 3

The application

3.1 Functional description

3.1.1 Satellite Data Set Building

Earth Explorer

An accurate analysis on glacier change in time needs a large enough data set of images to work with. The Landsat 8 images are free to use and in huge numbers, but they are not that easy to access in bulk. An example of an user friendly tool to access and view the images is **Earth Explorer** [44], which is used for searching catalogs of satellite and aerial imagery with the option of downloading data over chronological time-lines, wide range of specifying search criteria and a long list of satellite and aerial imagery to choose from [45]. The only downside of the tool is that searching and downloading data sets takes quite a lot of time, since its parameters need to be manually set, and for each found entry, the user must choose what to download individually, then use another application to launch the bulk download and finally have access to the local data. Overall, the process of collecting a qualitative and large enough data set using Earth Explorer is very time consuming, and is desirable only if the user wants to view the samples on the map before actually downloading them. the An example of the Earth Explorer tool is shown below, with searching of images from path 67 row 18, from Landsat 8 Collection 1 Level 1.

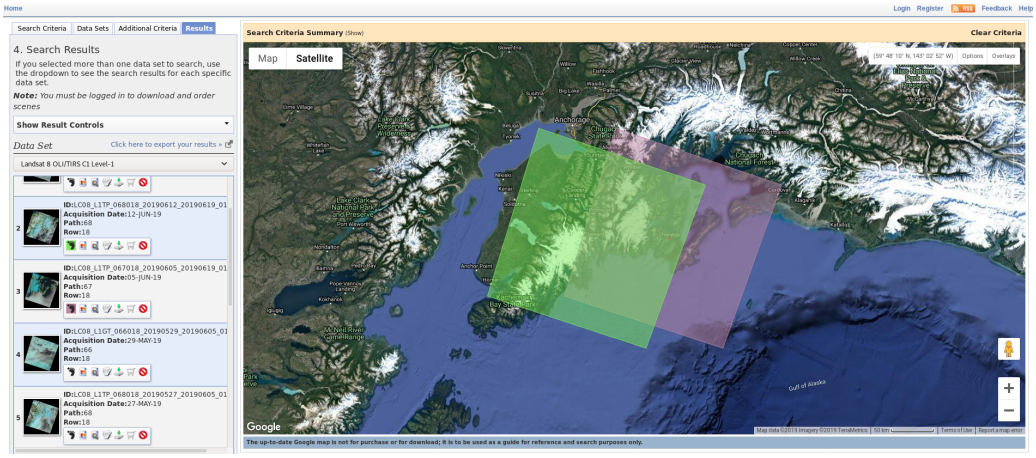

Figure 3.1: Earth Explorer [44].

As a solution to make sure that creating the data set when no visualization beforehand is needed is as trivial as possible was implementing searching and downloading of Landsat 8 scenes as a feature of our application with a simple command line interface call, as shown below:

When dealing with collecting of Landsat 8 data, there are x problems to take in consideration:

- Where is the data collected from?
- How are the glacier scenes found?
- Where are they downloaded to?

Where is the data collected from?

Landsat scenes are available to download from the Landsat Archive, which stores all the captured scenes since the Landsat 8 satellite was launched. All the information about the data is easy to access by filling in parameters such as path and row variables, specific to Landsat, cloud coverage, date of gather, coordinates and many more. The possibility to narrow or widen the area of search are huge given that all these parameters can be combined so a desirable set is achieved, then downloaded. **USGS Landsat Global Archive** contains the information about how they are structured and where they are exactly found [46].

How are the glacier scenes found?

In order to be able to download only necessary data from the Landsat Archive, information about where exactly the glaciers are located is crucial to have. In order to simplify the process of finding information about coordinates of glaciers around the globe, we used instead the **World Glacier Inventory** [47], which contains the coordinates of over a **130.000 glaciers** around the globe. The CSV file containing their id's and coordinates is included in the application in order to facilitate the searching and giving the option to achieve results as fast as possible, without also worrying about building a suitable data set. If the user wants to create their own data set instead, there are two more options provided by the World Glacier Inventory team:

- Search Inventory interface;
- Extract Selected Regions interface.

Search Inventory interface Search Inventory interface is a tool provided by the National Snow and Ice Data center (NSIDC) [47] which enables searching of glacier items for inspection by parameters such as glacier: name, number, coordinates, altitude, size and length;

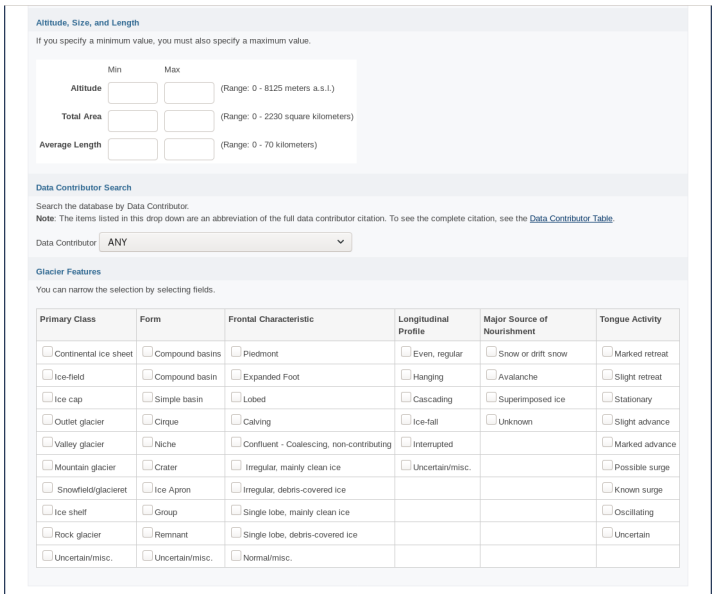

Figure 3.2: Search Inventory interface (NSIDC) [48].

Extract Selected Regions interface The third option is to search for glacier based on their region, such that in case the user wants information based only on location, a suitable data set can be easily created rather than searching each item by hand and keeping only the ones from the desired region. An example of the search web interface is displayed below:

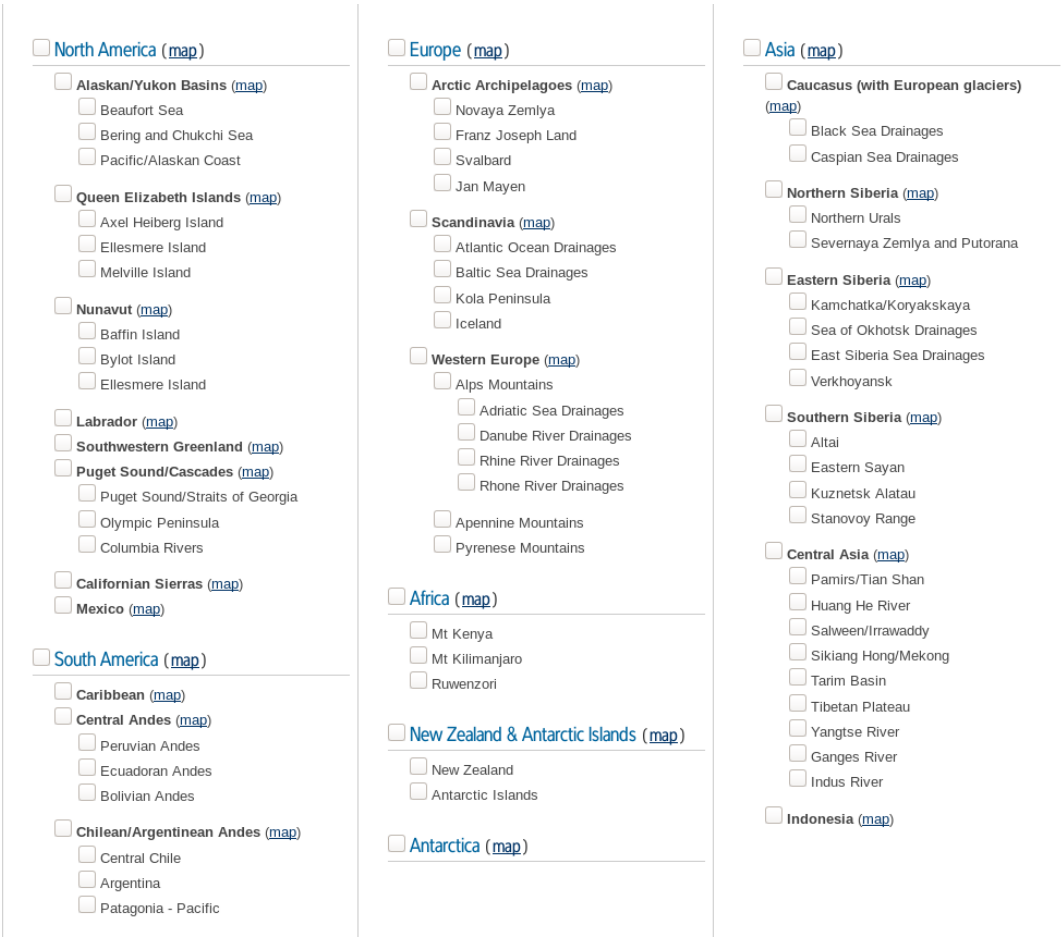

Figure 3.3: Extract Selected Regions interface (NSIDC) [**?**].

By choosing exact features from the searching tool, a smaller and more specialized data set is created in order to conduct a focused research.

Where are they downloaded to?

After the data set formed of Landsat 8 scenes has been built, the CSV containing the scenes information is used in order to begin the download of the selected files. In order to make the application as simple as possible from the user point of view, the selected items will be downloaded to a location of their choice on their computers. The folders will be

structured based on the glacier's id, their paths and rows and the files which represent the results of the processing. A example of the folder structure for downloading is presented below:

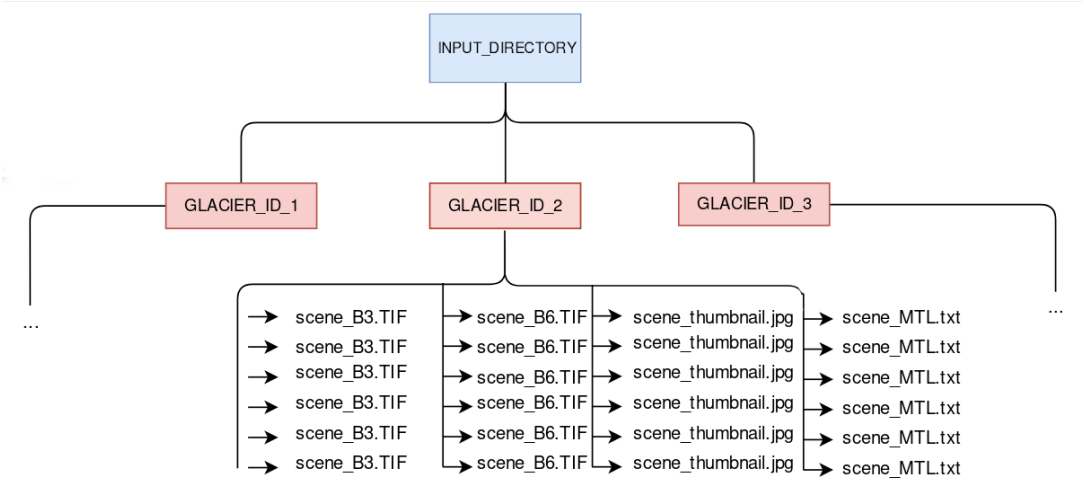

Figure 3.4: Download folder hierarchy.

3.1.2 Data Set Information Extraction

In order to process the gathered Landsat 8 scenes, we will use the information provided by the meta-data file downloaded with each scene, as well as the scene's name which is formed by the Landsat 8 convention, detailed in the section Data set Images, Landsat scene name convention. With the information extracted from these sources, we can get to the prepossessing stage of the application.

Scene Metadata (MTL)

Each Landsat 8 scene comes with a meta data file containing useful information about the package, such as: **scene ID, output format, WRS path and row, date of acquiring, capture time, coordinates, bands of scene and many more.** There are **approximately 150 variables describing the scene**, which makes it an useful file to have in order to extract information when necessary. Below is shown an example of a portion of the MTL file for a scene:

```
GROUP = L1_METADATA_FILE
  GROUP = METADATA_FILE_INFO
    ORIGIN = "Image courtesy of the U.S. Geological Survey"
    REQUEST_ID = "0501705111055_00001"
    LANDSAT_SCENE_ID = "LC81191582017131LGN00"
    LANDSAT_PRODUCT_ID = "LC08_L1GT_119158_20170511_20170511_01_RT"
    COLLECTION_NUMBER = 01
    FILE_DATE = 2017-05-11T06:13:06Z
    STATION_ID = "LGN"
    PROCESSING_SOFTWARE_VERSION = "LPGS_2.7.0"
  END_GROUP = METADATA_FILE_INFO
  GROUP = PRODUCT_METADATA
    DATA_TYPE = "L1GT"
    COLLECTION_CATEGORY = "RT"
    ELEVATION_SOURCE = "GLS2000"
    OUTPUT_FORMAT = "GEOTIFF"
    SPACECRAFT_ID = "LANDSAT_8"
    SENSOR_ID = "OLI_TIRS"
    WRS_PATH = 119
    WRS_ROW = 158
    NADIR_OFFNADIR = "NADIR"
    TARGET_WRS_PATH = 119
    TARGET_WRS_ROW = 158
    DATE_ACQUIRED = 2017-05-11
    SCENE_CENTER_TIME = "03:18:28.5807450Z"
    CORNER_UL_LAT_PRODUCT = -36.38192
    CORNER_UL_LON_PRODUCT = -71.66107
    CORNER_UR_LAT_PRODUCT = -36.41157
    CORNER_UR_LON_PRODUCT = -69.02900
    CORNER_LL_LAT_PRODUCT = -38.51580
    CORNER_LL_LON_PRODUCT = -71.73781
    CORNER_LR_LAT_PRODUCT = -38.54782
    CORNER_LR_LON_PRODUCT = -69.02984
```

Figure 3.5: Metadata file.

3.2 Pre-processing

The pre-processing phase of the application implies traversing the directories which contains glaciers and build the folder structure to hold the results of the processing of the data set. Each glacier will have its own directory, formed of path and rows directory for all its pairs. The path and row directories will contain the results of the processing, the aligned NDSI images, as well as a csv file with information about the results of the processing, which will be described in The folder splitting hierarchy is designed as follow:

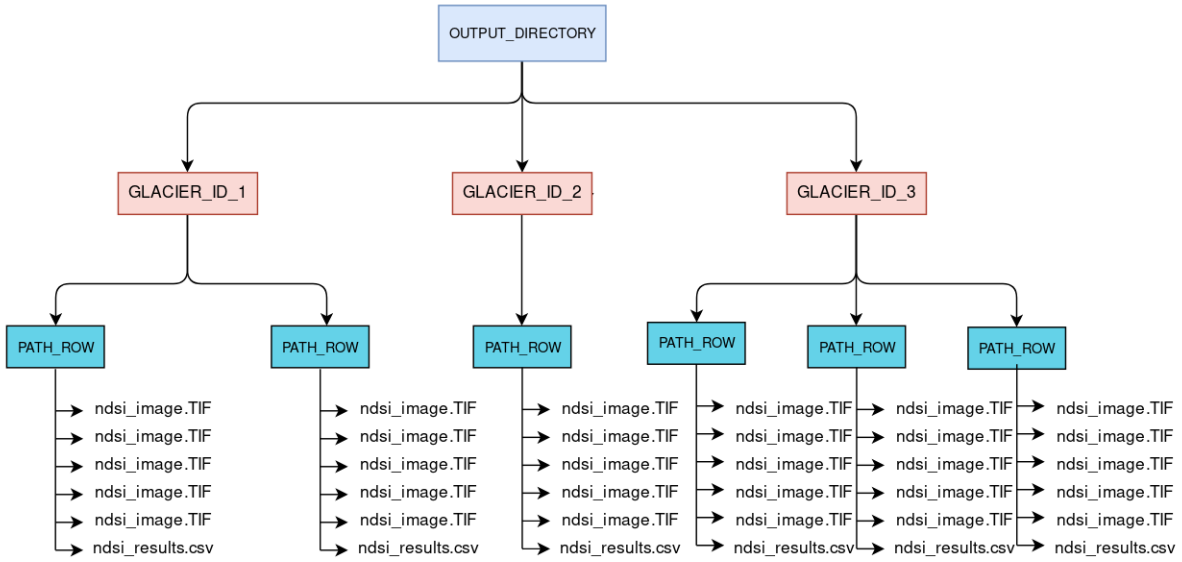

Figure 3.6: Output folder hierarchy.

3.3 Processing

In order to achieve a **time series forecast on snow coverage**, processing of the images is necessary. After the data set has been built, the information extracted and the pre-processing done, the paired images (by green and SWIR1 bands) will be processed with the following actions:

- **Normalized Difference Snow Index (NDSI);**
- **Alignment;**
- **Object appearance and disappearance detection;**
- **Object movement flow.**

3.3.1 The Normalized-Difference Snow Index

The NDSI represents the normalized difference of the two described bands, the green band in the visible spectrum and the SWIR1 one in the shortwave infrared spectrum. By calculating the NDSI between the two bands, the result will enhance snow in a Landsat scene, due to snow's high reflectance in the visible spectrum (0.66 mm) and high absorption in the short-wave infrared one (1.6 mm) [27, 28]. The mapping is done by computing

the following formula:

$$ndsi = \frac{green - swir1}{green + swir1}[28]$$

The NDSI is used at mapping the glaciers from the collected Landsat 8 images, in order to enhance them and make it easier to observe, analyses, apply image alignment based on their features and predict their movements based on pixel-analysis. For snow mapping, the NDSI is applied on the two images with a threshold of contrast so the gray-scale will be just black and white, meaning glacier or no glacier pixel. This is done so that when processing the images on time-series, the values are easier to parse and predict. By applying the formula on the Figures 2.6 and 2.7 and a threshold of contrast of 13000, the following result is created:

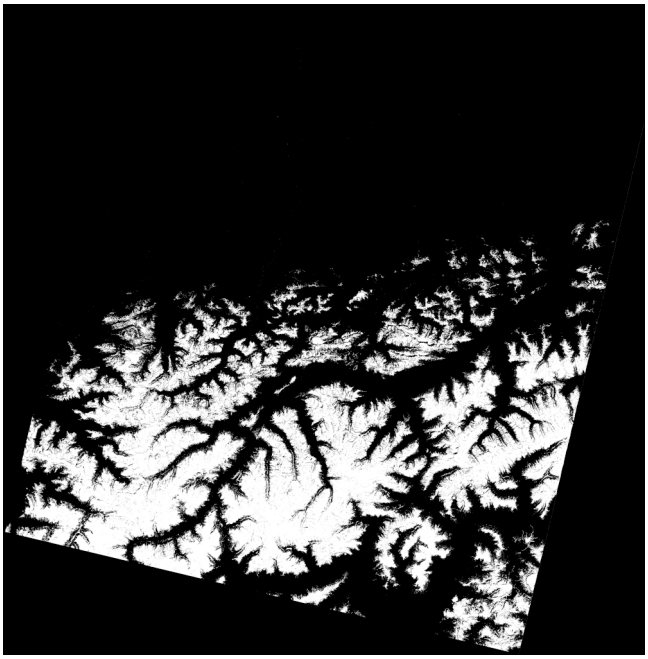

Figure 3.7: NDSI and contrast between green and SWIR1 bands.

3.3.2 Image Alignment

As a part of creating a forecast on a data set of images, alignment between them is necessary, due to pixel discrepancy from image to image. The difference is created by Landsat's trajectory when orbiting the Earth, which is not precise, therefore, not always passing on the same trajectory so the images are aligned to pixel level. Below will be exemplified the discrepancy between two green bands from two different scenes form the same path, row and glacier:

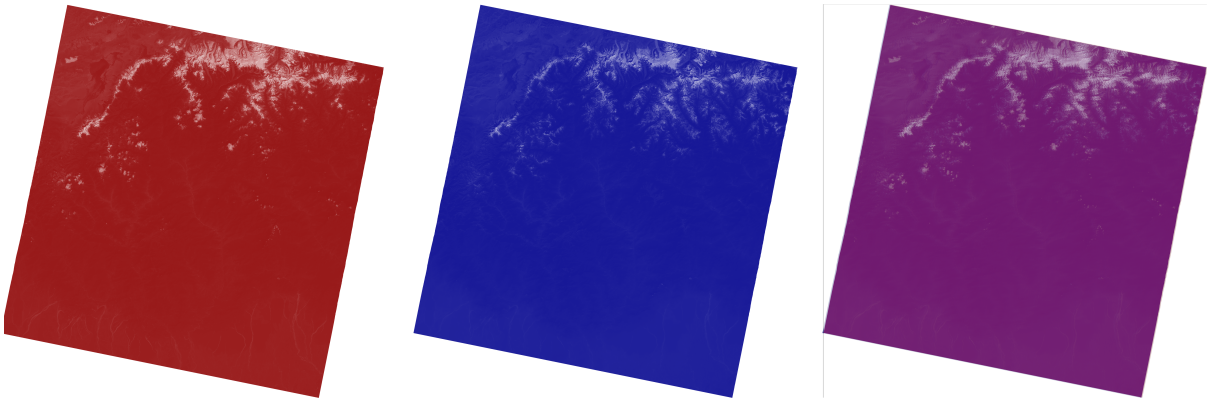

Figure 3.8: Two unaligned Landsat 8 scenes, overlapped.

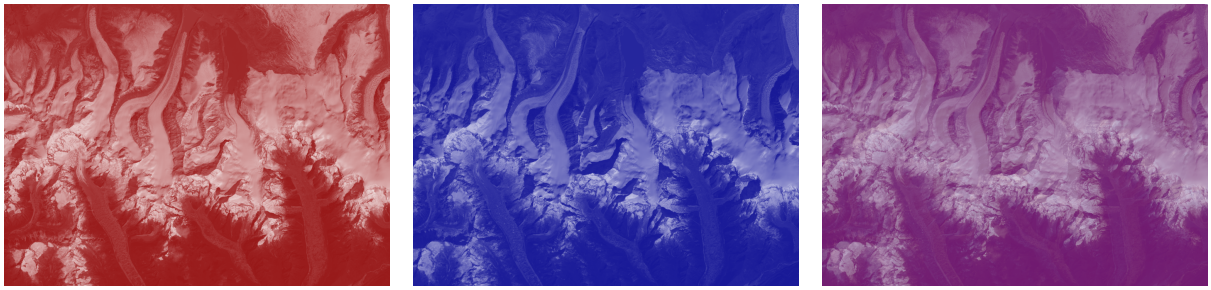

Figure 3.9: Zoomed in two unaligned Landsat 8 scenes, overlapped.

Given that Landsat 8 has a spatial resolution of 30, each pixel represents 30 meters in the real world. A discrepancy of 100 pixels creates a large spatial error, which lads to errors in determining whether there was snow in the same location between images. A discrepancy even as little creates an unreliable analysis on evolution of snow coverage and glaciers. Therefore, aligning the images is necessary in order to proceed with the results. The following images are the same ones as above, only aligned:

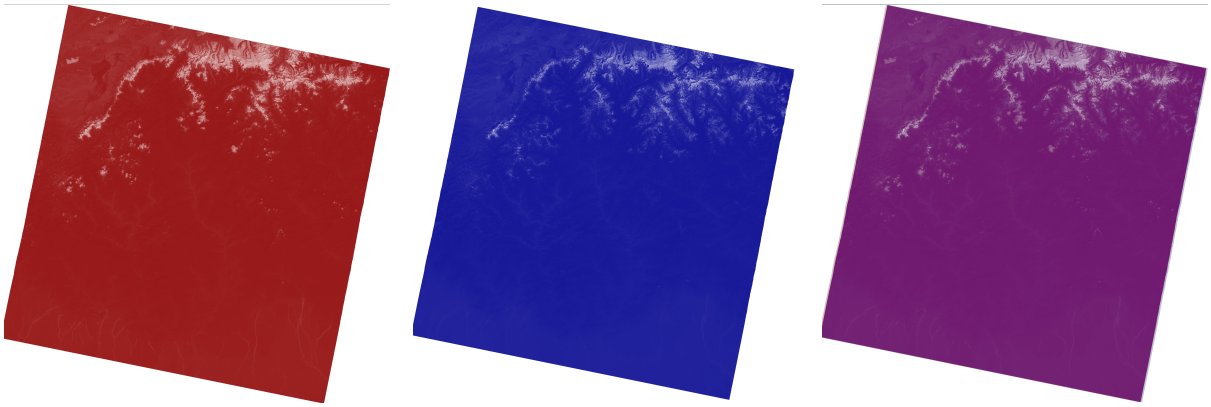

Figure 3.10: Two aligned Landsat 8 scenes, overlapped.

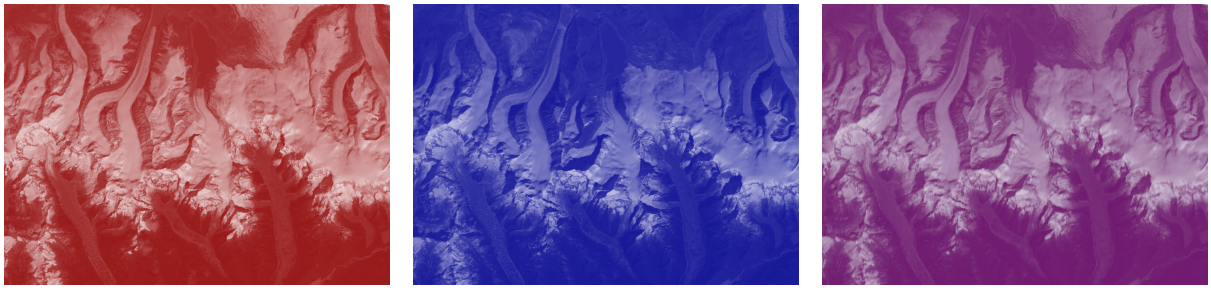

Figure 3.11: Zoomed in two aligned Landsat 8 scenes, overlapped.

Several methods for achieving close to perfect alignment results have been used, which will be described in the Design and Implementation section. chapter. The images above have been chosen as example due to the appearance of glacier melting, snow and dry land, which are the characteristics taken in consideration for determining the object appearance or disappearance, as well as its movements in time. By also being very similar, the discrepancy can be easily seen in the initial ones. The creation of the aligned data set from the initial one ensures that these processing actions can be completed with great results, rather than working on unaligned images.

3.3.3 Diference

There are two cases for defining the state of the snow coverage in a satellite image, in our case:

- The snow is in movement motion.
- The snow appeared or disappeared suddenly.

Differentiating is between two NDSI images is done in order to detect appearance or disappearance of snow masses or glaciers, which analyzed on a time series of images, highlights the difference of glacier on focus with passing years. Successfully determining the state of objects in scenes leads to information such as how constant the changes are, how fast they are moving, the mass of snow melted and even complete disappearance of some glaciers due to high temperature. The following pictures illustrate glacier and snow masses change, as well as their color maps in order to highlight the changes for visual interpretation:

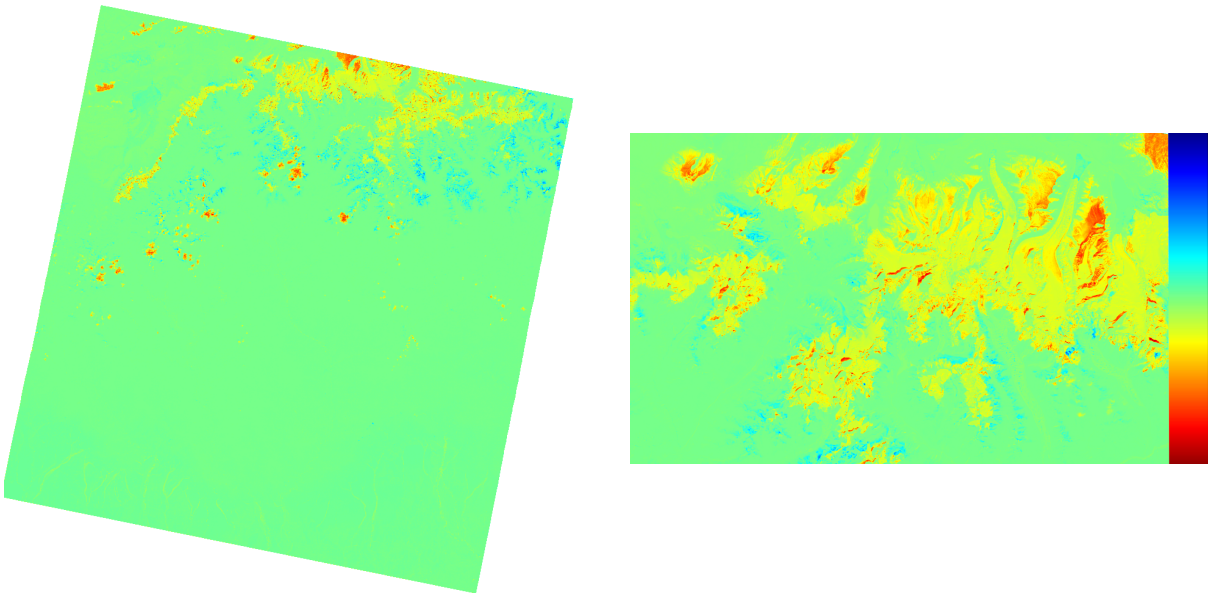

Figure 3.12: Difference for object state.

3.3.4 Movement

The movement of objects from an image is equally important in extracting information, such as the angle of the rotation, its magnitude, information about the speed of the changes and their scale. Applied on an NDSI scene, the following result is obtained:

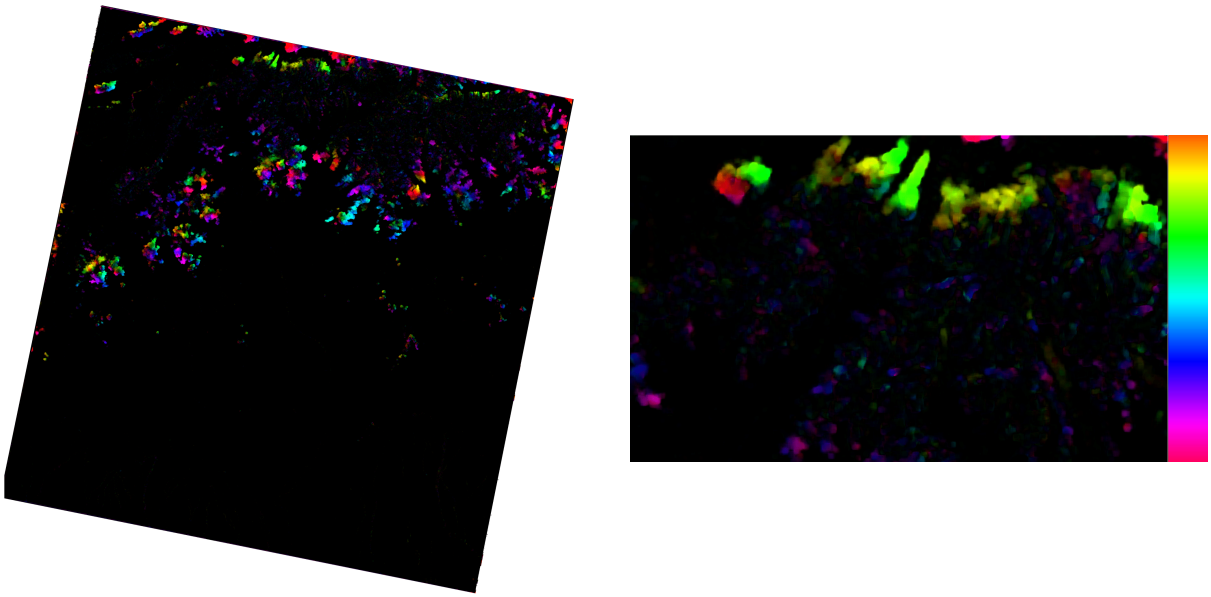

Figure 3.13: Difference for object movement.

3.4 The user interface and guide

The user has two options regarding the interface, for better convenience, since some users are more inclined to using a graphical user interface rather than a command line one. The following are the options:

- **Command Line Interface (CLI)**
- **Graphical User Interface (GUI)**

The **command line interface** has been chosen for reasons such as **simplicity, efficiency, faster results**, which can be opted for by users which have a higher experience in programming and dealing with such an interface.Command line applications have been used since the development of the first computers and are still being used today for the same reasons specified before, as well as access to other programs and features in the system which would be hardly accessible to graphical user interfaces. Since the thesis application is more process driven rather than having many visual use cases, opting for keeping a simple command line interface was the best choice in the case of those who which to achieve simpler and faster results.

The **graphical user interface** is taken as an options for users which do not have much interaction with the command line environment, rather being used to GUI's. By including this option, the number of possible users is enlarged, including also the ones which are not that experienced in the IT field, but still want to use the application without much trouble.

We will describe each process separately, with their GUI and CLI interfaces.

3.4.1 Download GUI

The download process can be started by pressing **START PROCESS** button, and stopped with **STOP PROCESS** one. The diagram which represents the interaction with the graphical user interface, as well as the command line one can be found in 3.14 figure. The displaying from command line can be started as:

```
python3 main.py download --csv util/files/wgi_feb2012.csv --dir
    /path/to/download/ -j 3
```

3.4.2 Process GUI

The processing can be started by pressing **START PROCESS** button, and stopped with **STOP PROCESS** one. The diagram which represents the interaction with the graphical user interface, as well as the command line one can be found in 3.15 figure. The displaying from the command line can be started as:

```
python3 main.py download python3 main.py process --input
    /path/to/input/images --output /path/to/download/ -j 3
```

3.4.3 Display GUI

The displaying of the results can be started by pressing **DISPLAY** button, and stopped with the **EXIT** button from the graphic pop-up or its own. The display interface prints the **statistical information** processed from calculating the NDSI, and generates a **prediction of snow coverage**, displaying the results in an interactive plot. By pressing on a plot entry, the **difference and movement images** are created. The diagram which represents the interaction with the graphical user interface, as well as the command line one can be found in 3.16 figure. The displaying from the command line can be started as:

```
python3 main.py display --csv /path/to/csv/ndsi_144_039.csv
    /path/to/download/images -j 3
```

3.5 Difference and Movement GUI

The calculation of difference and movement images can be triggered by clicking two of the items found in the interactive plot. The following diagram represents the interaction with the graphical user interface.

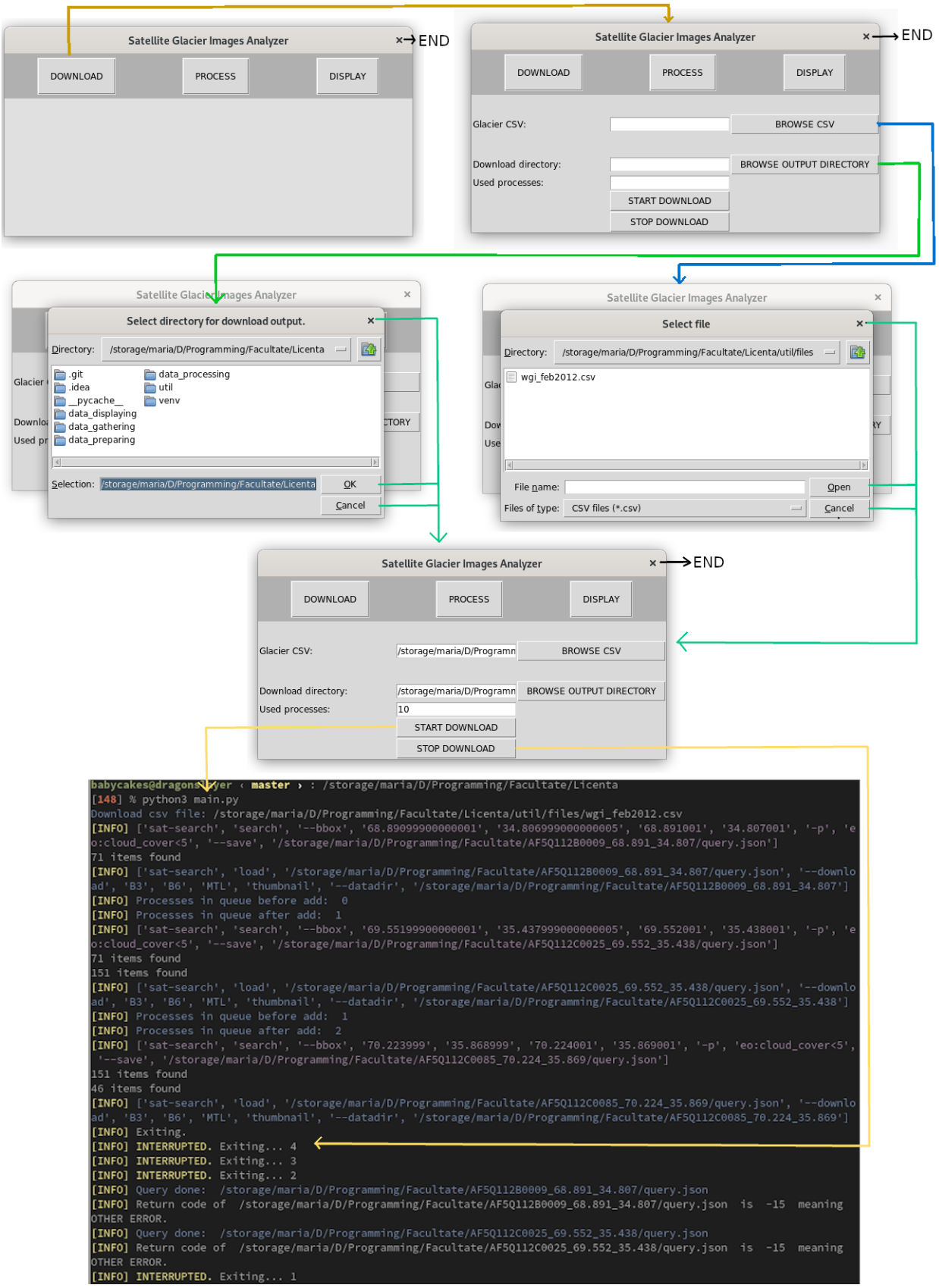

Figure 3.14: Download CLI and GUI diagram.

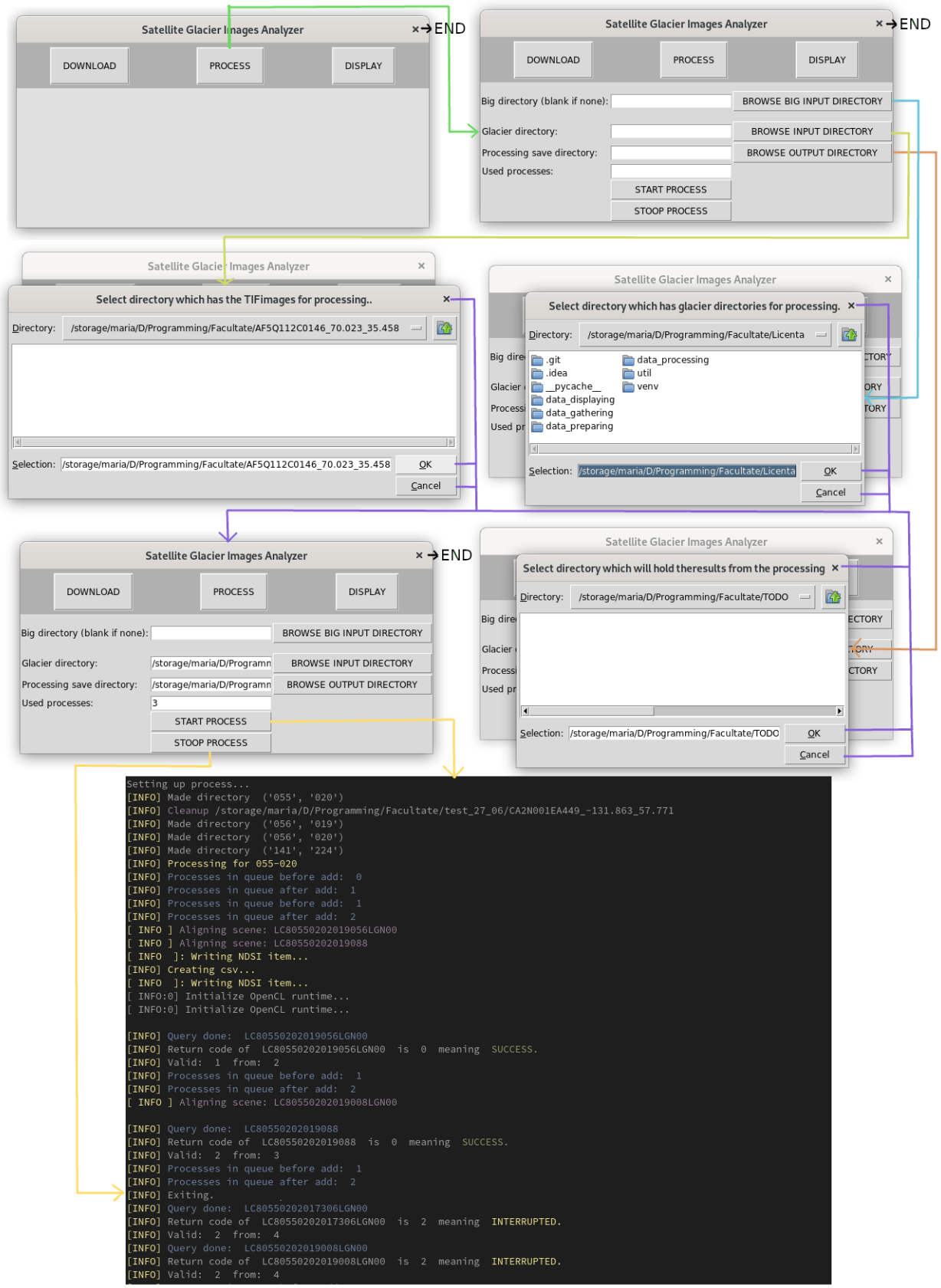

Figure 3.15: Process CLI and GUI diagram.

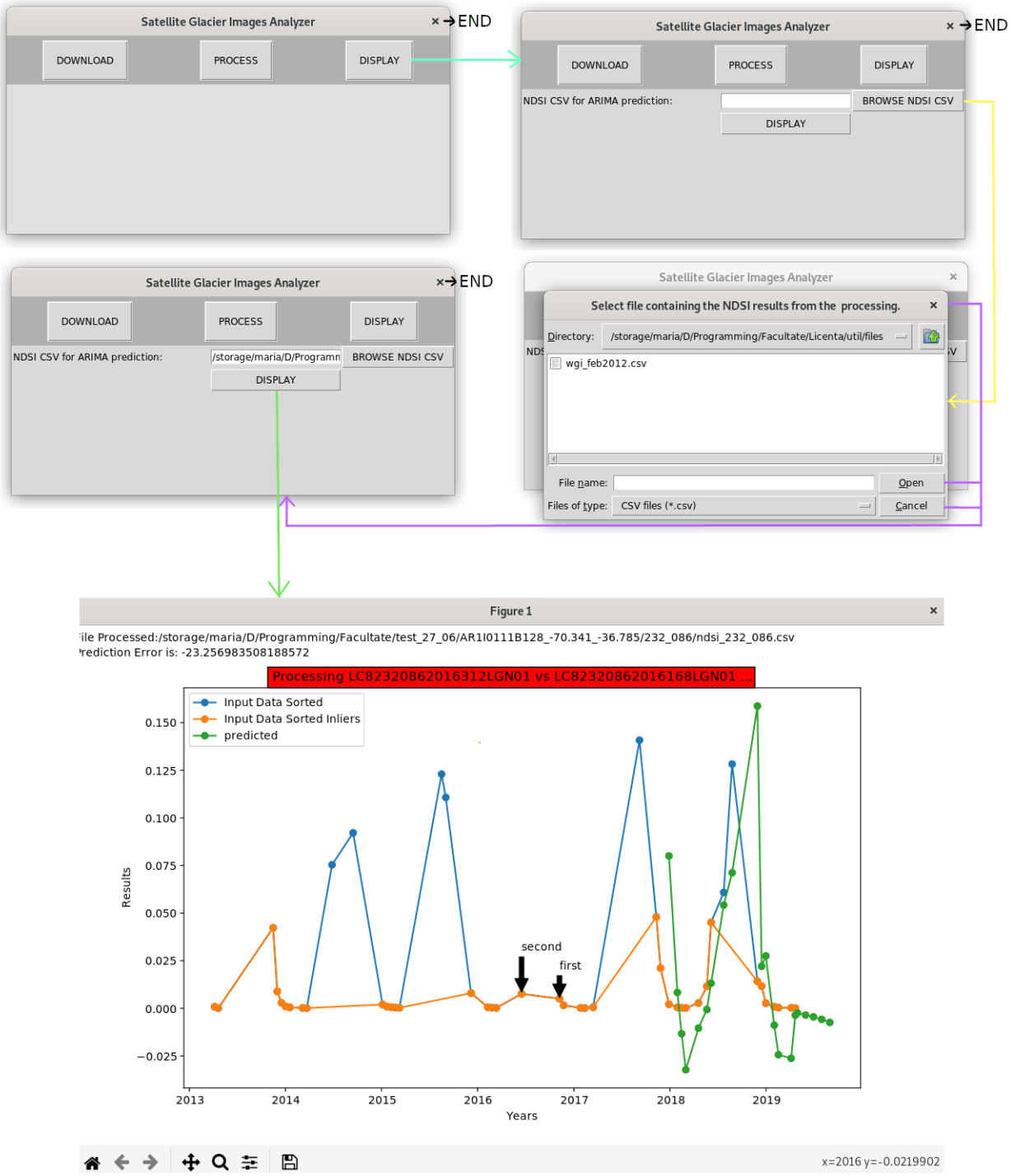

Figure 3.16: Display CLI and GUI interactive prediction plot.

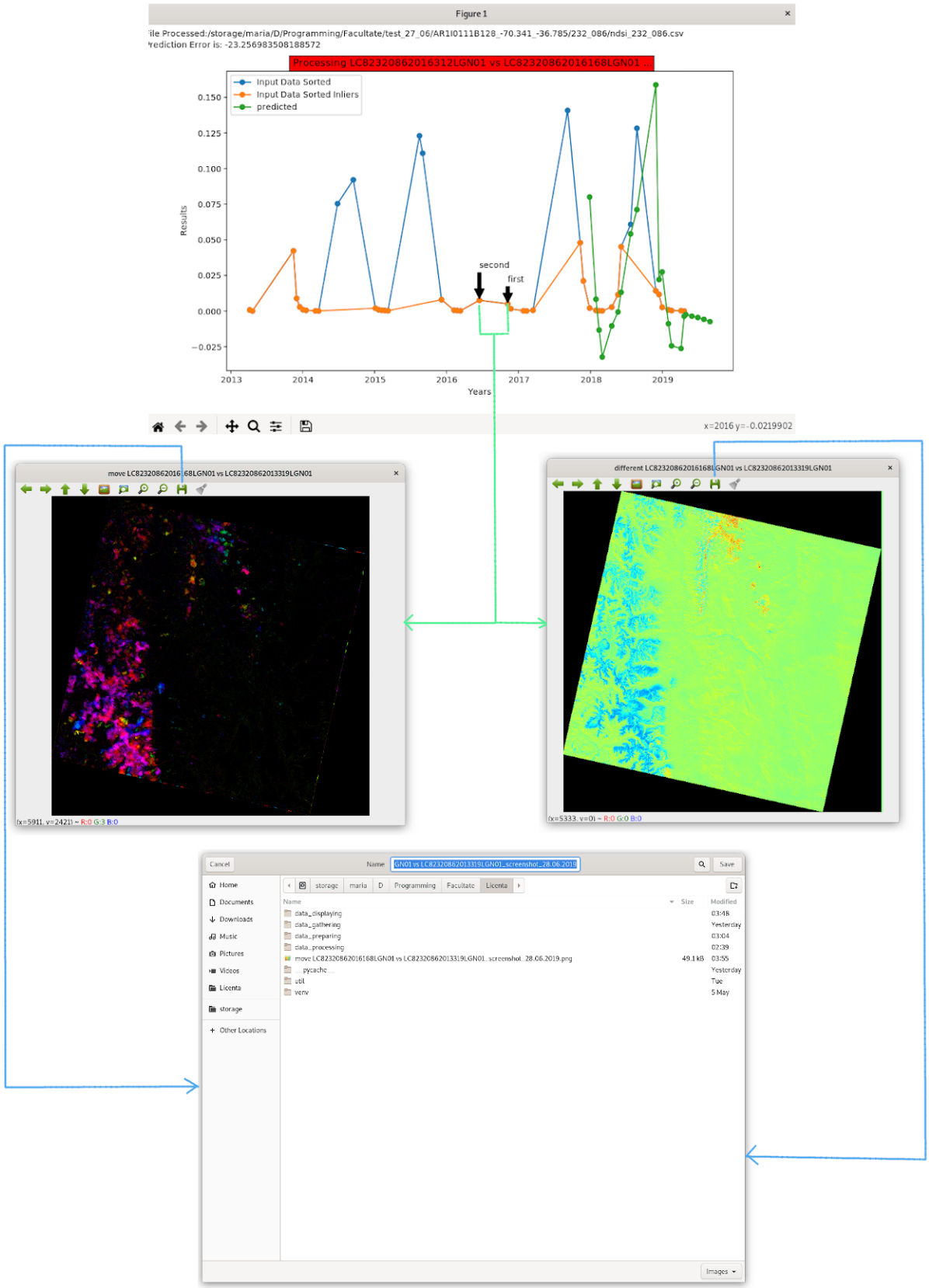

Figure 3.17: Prediction interactive plot CLI and GUI.

3.6 Main use cases

The main use case of the application consists of three actions: **download, process, display, predict**. The use cases of those are the same as described in the user interfaces chapter, while the main use case is the following:

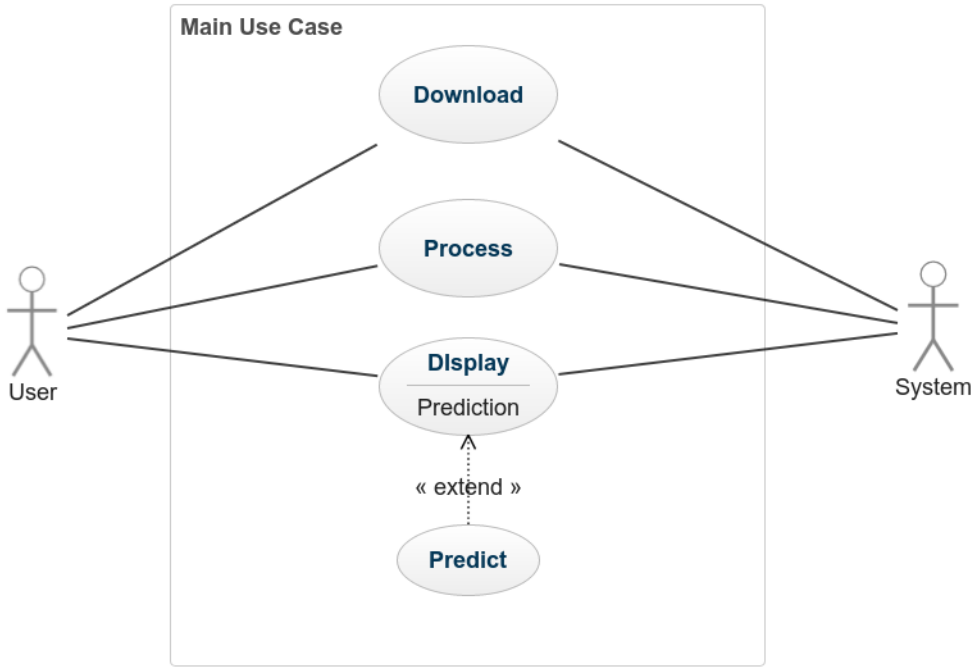

Figure 3.18: Main use case of the application. The user can choose between downloading, processing or displaying and predicting data. Prediction can only be made if display is chosen, and the interactive plot will pop-up.

Chapter 4

Design and implementation

From the implementation point of view, the application will be described by splitting it into four parts: **download, process, display, predict**, since the three can be run independently, and each is a main part from the whole. The general view is that the application can search and download images if the user does not have any, which are handled by using the **sat-search library**; if the user already has a directory with images, the next step is to **calculate the NDSI** in order to extract the snow pixels from the images, and align them, so that implementing the creation of difference and movement images can be done without discrepancy between pixels, which would result in blurry, incorrect images. The resulting images are written to a directory which the user chooses, along with two csv files, one which holds the results from the NDSI processing (ratio of snow pixels found in each image, along with other information which will be discussed in more detail in the processing section), and an alignment csv, which contains data on how many successful alignment have been created, and the parameters used for achieving such results. The NDSI csv is necessary in order to calculate the prediction on snow coverage for the path and row of a glacier, as well as starting the **interactive plot which will be used to view the results**, and calculate the difference and movement pictures between two chosen images.

4.1 Download

The download process is implemented based on **sat-search**, by using the CLI or GUI, paired with the .csv file which contains information on the glaciers to be searched and

downloaded, procured by the user beforehand (either by using Earth Explorer or World Glacier Inventory's searching engines, displayed in Figures 3.1, 3.2 and 3.3). The structure of the .csv file depends on the chosen parameters to download (the process of downloading from the World Glacier Inventory [47] is described in the Functional Description chapter). When the user procures the file containing the necessary information for starting the download, it can be used to start searching. The searching, as well as the download, are implemented using **multiprocessing**, which fastens the waiting time, which is necessary, since it is quite expensive due to the large download file resolution (one .TIF image is between 35 and 65 MB, which makes just one folder of images approximately 1-3 GB, depending on how many images are found.), as well as downloading from the Landsat 8 Archive. Searching returns a **json search file**, which is passed to download. Since we use multiprocessing to improve the execution time, ensuring that a download process does not start before a search process, the searching was implemented synchronous, while the download is asynchronous, since the speed of download depends on network parameters. The images are downloaded in a folder chosen by the user, having a file hierarchy described in the Functional Application chapter (Figure 3.4). The class diagram of the downloading process is described below:

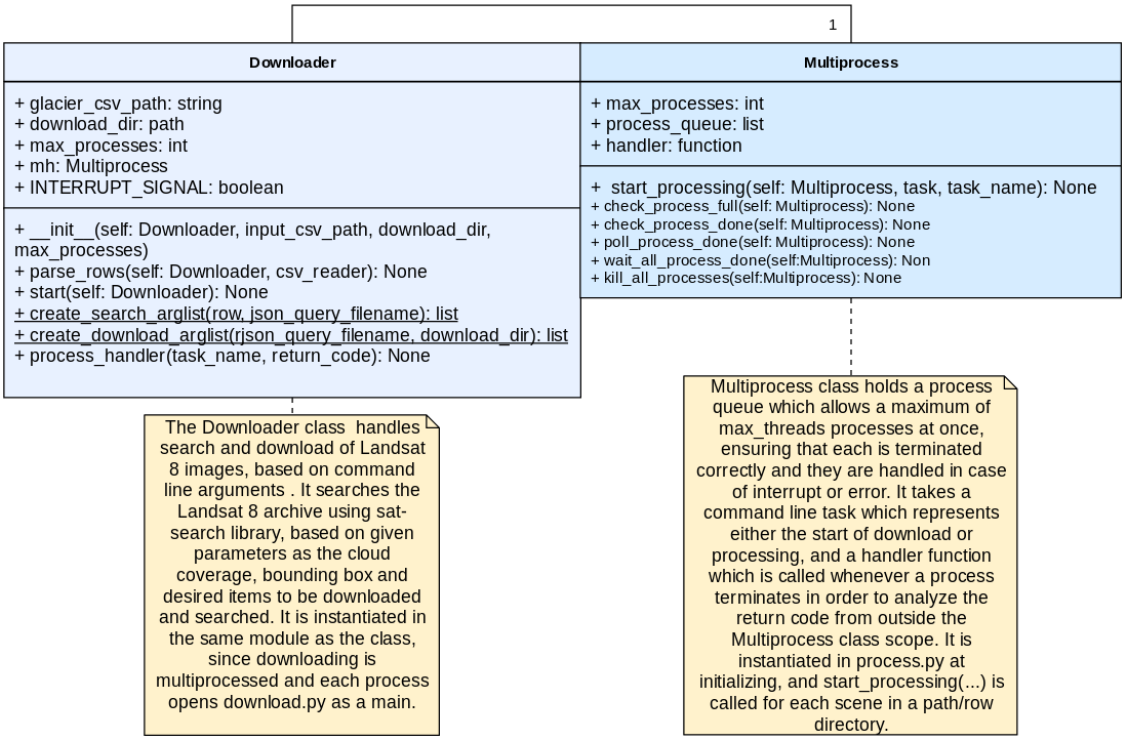

Figure 4.1: Download process class diagram.

4.2 Alignment and NDSI

4.2.1 Alignment

Accurate alignment of images in a path row folder is required in order to ensure a correct creation of difference and movement pictures (described in Functional Application, Figures 3.12 and 3.13). The **goal** of this processing part is to achieve results with as little misalignment as possible, meaning distances between pixels with the same coordinates close to 0. Alignment of two images is done by **collecting features** from each image and trying to **match them** in order to obtain a list of matches which will be used by the **warping algorithm** to distort the image on which is applied, such that there is no misalignment between them at the end.

In our case, since the images are almost similar visually, we rely on having a strong **corner detection algorithm to find edges in mountains** and other geographical features from the images, since they are stable in time and can be reliable as a **key-point**. After ensuring that the keypoint detection was successful, we need to calculate the **descriptors** in order to uniquely identify a small path of data in a huge image (the key-points are found as a circle of 16 pixels in the image which has a resolution of between 7000x700 and 8000x8000). With the key-points assigned to their descriptors, we have to make the ties of features between the two images: since each image has a set of key-points and descriptors, if the algorithm finds that two of the last are equal, that means it found a feature which will be used as a matching line between the reference image and the current one. We need to **set a limit** on how many key-points are found, as well as how many matches are considered good. In our case, the best results were by considering **5000 key-points** with **25%** of the **matches** considered good (taken by top). The next step after trimming the bad matches out is to calculate a **warp matrix** in order to distort the image such by doing that, there will be no misalignment, or as one as close to 0, between the pixels of the reference image and the ones from the image which we apply the warp on.

In order to achieve such results between two images, we used the following computer vision intelligent algorithms:

- **Oriented FAST and Rotated BRIEF (ORB)** [51];

- **Harris corner measure** [52];
- **Random Sample Consensus (RANSAC)** [53].

ORB is a fusion of **Features from accelerated segment test (FAST) keypoint detector [49] and Binary Robust Independent Elementary Features (BRIEF) descriptor** [50] with many modifications to enhance the performance. First it use FAST to find key-points, but its corner detection is not performant enough, failing to find enough good results; Since we need powerful corner detection on mountain features, we paired up ORB's key-point and descriptor detection with **Harris corner measure [52] to find top 25% points among them**, which also uses pyramid to produce multiscale-features for large files. [51].

After creating the matches, we use the **RANSAC** algorithm create the **warping**. RANSAC is an iterative method to estimate parameters of a mathematical model from a set of observed data that contains outliers, when outliers are to be accorded no influence on the values of the estimates [54]. We chose this we know that between our images there is an **affine transformation** (translated, rotated, scaled), and this technique helps filter out bad matches only leaving the ones that satisfy the motion model between two images.

By using this starting point, the following **improvements** were applied in order to get to a **correct alignment percent** of almost **98%** for approximately **85% files in a path row folder**:

1. **Two feature sets**: taking features from the **reference and current image** (green or swir1);
2. **Box split**: splitting the two images in **8 boxes** and taking keypoints and descriptors from each;
3. **Prune matches**: splitting the two images in 8 boxes and taking features from each, while also **pruning the matches** which were not almost straigH;
4. **Three feature sets**: Taking features from the **reference and both green and swir1 matching bands**, while also splitting them in 8 boxes and taking keypoints and descriptors from each and applying pruning of matches.

1. Two feature sets When aligning the images, there are two cases: either we align a **green band** or a **swir1 band**. By analyzing the results of this approach on **237** glaciers, we observed that the **mean value of ratio successful alignments** on a path row directory was **0.054868**. An example of a match result and align result using this approach is displayed below:

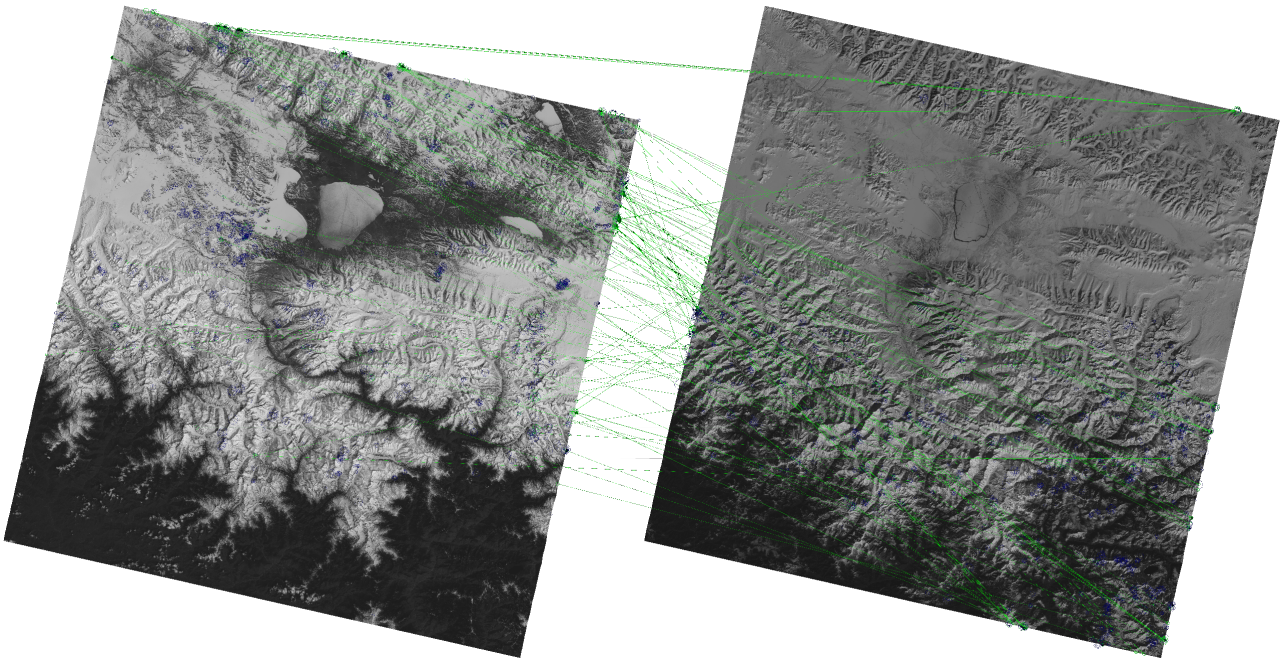

Figure 4.2: Matches between the reference image (left, green) and image to be aligned (rigH, SWIR1).

Figure 4.3: Image after warping the affine matrix on the SWIR1 band.

2. Box split An improvement to the first approach was to **split the images into 8 boxes**, and apply ORB and Harris on each box; This step was needed because in most of the results, features were only found on a box of 25% of the image, which completely unbalances the affine matrix calculation. By this approach, we were able to find much better results, on the same **237** glaciers, we observed that the **mean value of successful**

alignments on a path row directory was **0.3854**, which means an **improvement** with **0.330532**.

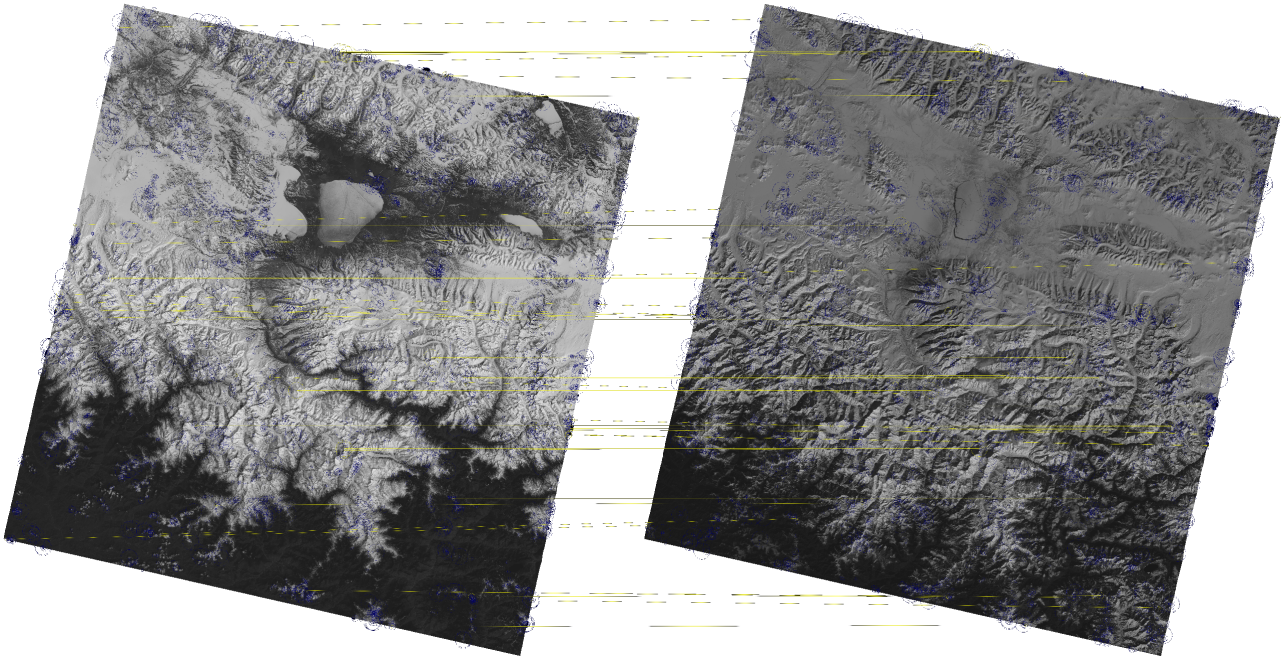

Figure 4.4: Matches between the reference image (left, green) and image to be aligned (right, SWIR1), after box feature find.

3. Prune matches Even if the last approach gave better results, the ratio could be brought up, by **pruning the bad matches**. Since the Landsat 8 satellite could not have errors of more than approximately 200 pixels, any match which has a point discrepancy of **more than 200 pixels** on the xy axis represents a **bad descriptors match**, and therefore should not be taken into consideration. By this feature, we were able to find result with a mean of **0.733**, on the same **237** glaciers, which means an **improvement** with **0.3476**. The following example of before and after prune match is displayed below:

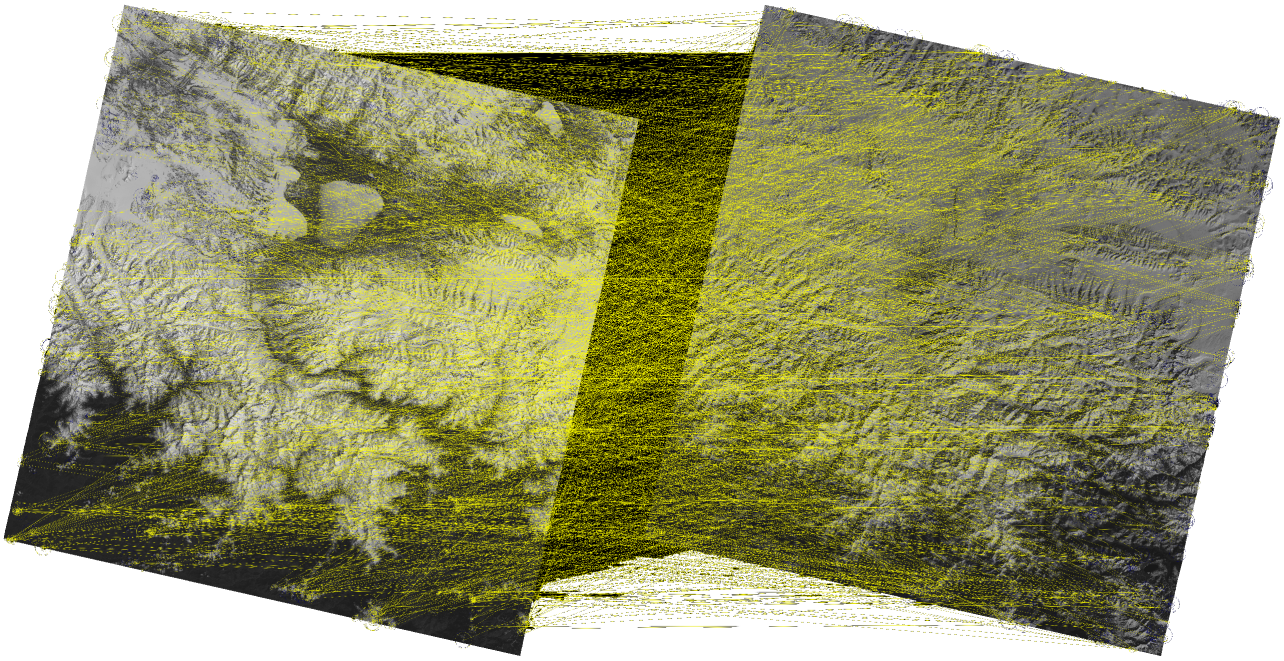

Figure 4.5: Matches between the reference image (left, green) and image to be aligned (right, SWIR1), before prune matches improvement.

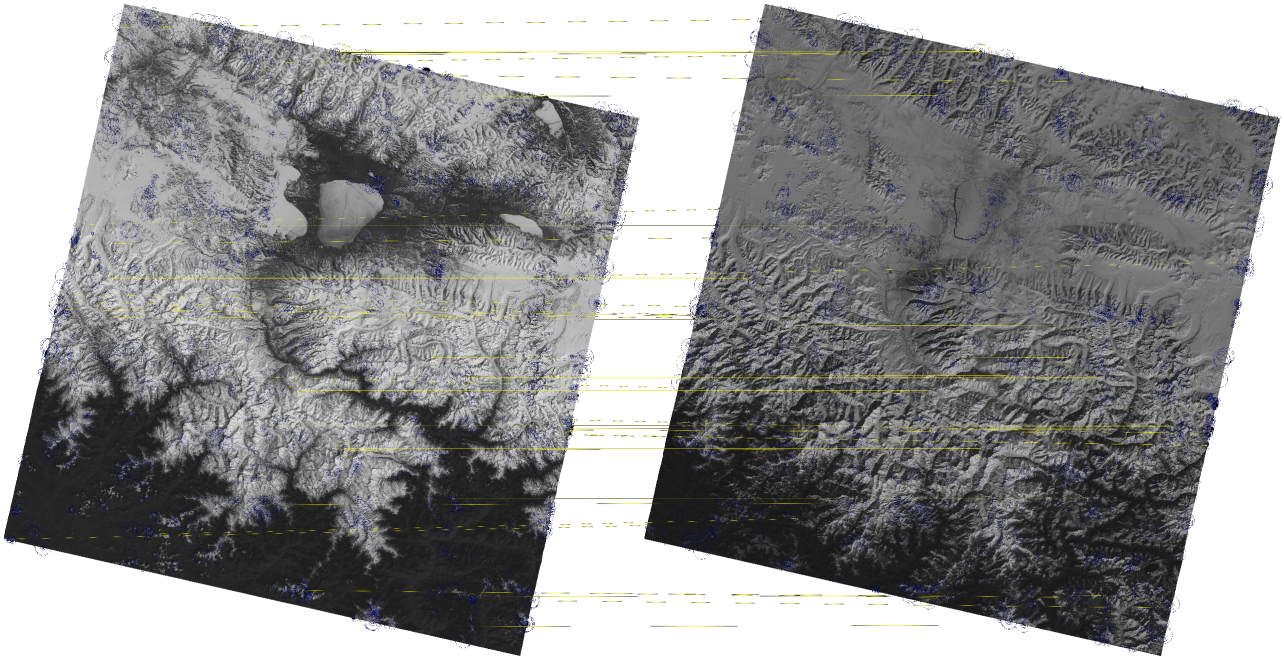

Figure 4.6: Matches between the reference image (left, green) and image to be aligned (right, SWIR1), after prune matches improvement.

4. Three feature sets Finally, the improvement which achieved a ratio of successful alignments over total trials of **0.896** was gathering features not only from the reference image and the current green or SWIR1 one, rather both, since they are already aligned between each other. The Landsat 8 has pixel discrepancy between different scenes, but all the bands are aligned. By taking key-points from the reference, current band and its pair, we were able to achieve an improvement which is more than enough in producing aligned images (NDSI, green and SWIR1). The following image is the result of the **alignment using all the four features**.

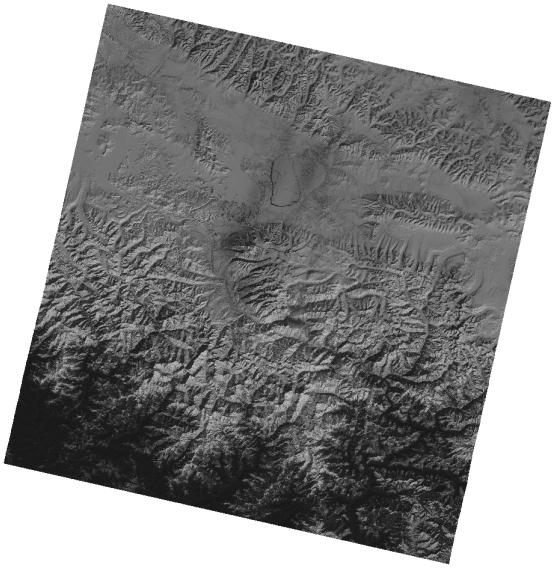

Figure 4.7: The aligned image after applying all four features.

4.2.2 NDSI

The NDSI is calculated as described in the Functional Decription, by reading the **image as a numpy array**, and applying the formula

$$ndsi = \frac{green - swir1}{green + swir1}[28]$$

on the n-dimensional arrays. A **threshold of 0.5** is used for distinguishing between snow and other objects in the image. The NDSI is calculated for each pair of green and SWIR1 images along with alignment, such that when alignment is started, the NDSI is already created and can be included in the **scene with NDSI** image (NumpySceneWithNdsi class). The **snow ratio** of the image is introduced in the **NDSI csv** for **construction of the prediction data set**. The NDSI has the class diagram displayed in Figure 4.8.

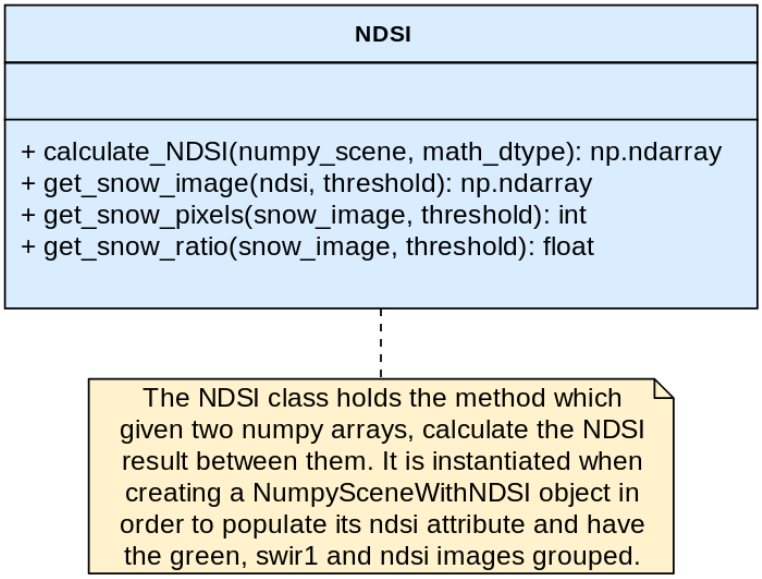

Figure 4.8: The NDSI class diagram..

The alignment and NDSI processing is done by the **ProcessImage class**, which binds the two processes together in order to achieve the goal of creating a path and row folder of aligned scenes with computed NDSI. The processing including NDSI and AlignORB has the following class diagram, as shown in Figure 4.9.

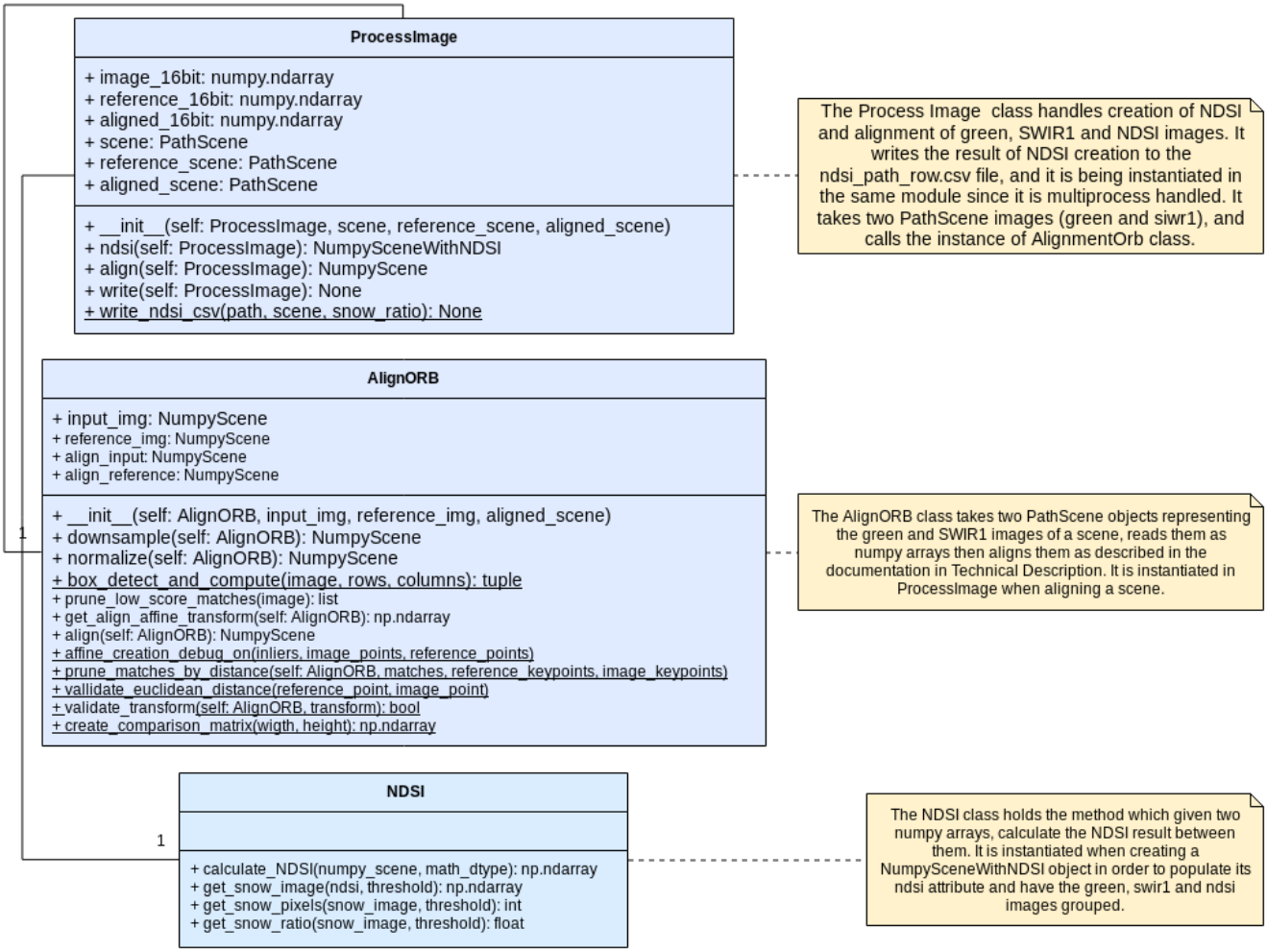

Figure 4.9: The alignment and NDSI processing for a path and row directory class diagram.

4.3 Display and ARIMA prediction

4.3.1 Data set normalization

The display process handles creating the plot for **data set prediction and creation of difference and movement images**. When calling the display process, the **ndsi_path_row.csv** file is analyzed using the data set handler module (class DatasetHandler) as:

- **Sorting the data set entries such that the images are in chronological order.**
- **Detecting and removing outliers from the data set.**

After the data set has been normalized, it is ready to be redirected to **plotting and predicting**.

4.3.2 Data set plotting

The normalized set of snow ratios resulted from the NDSI process is displayed in an interactive plot created with **matplotlib** library, as:

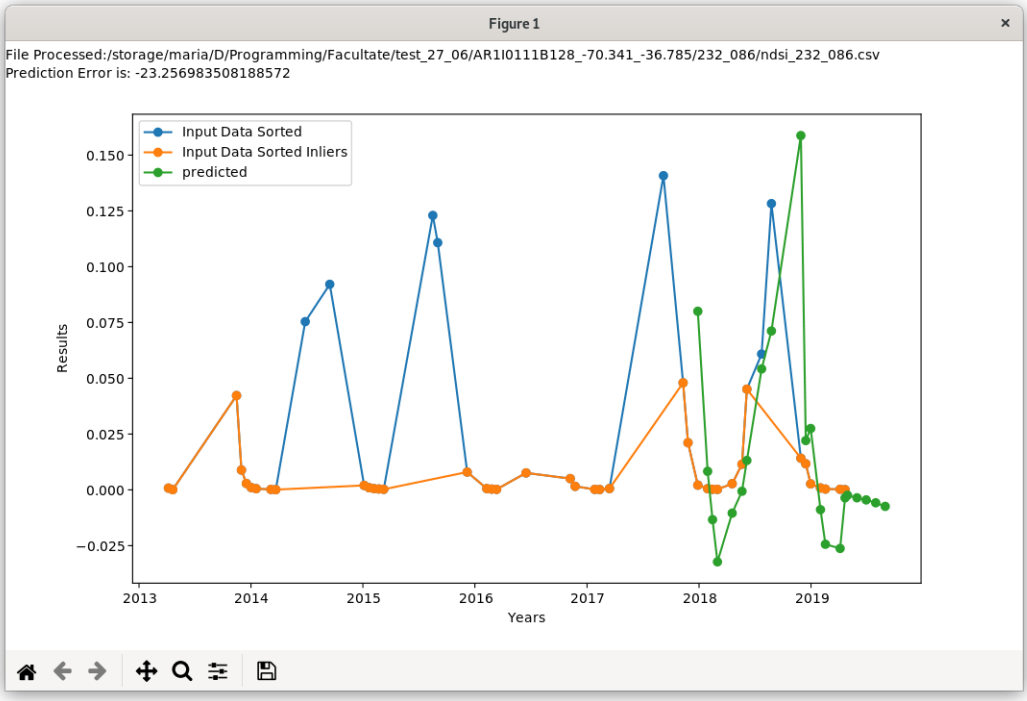

Figure 4.10: Plot of glacier AR1I0111B128, with coordinates(-70.341, -36.785), on path and row 232, 086.

4.3.3 Data set predicting

We used the ARIMA predicting algorithm with (4, 2 and 0) parameters [57] by splitting the data set into **66% train** and **33% test**, in order to give ARIMA a view over the data set and ensuring that the prediction is as accurate as possible, with a **small mean error**. We ensured that the prediction object matches the good position one by finding the nearest neighbor so the error is calculated both on x and y axes (time and snow ratio). **The mean error** is calculated by summing up the differences between the prediction and its nearest neighbor measurement, then divided by the total number of predictions and the max value of the snow ratio index. The **maximum value** is used instead of the nearest neighbor value because the **discrepancy between objects is very small**, close to zero, which **leads to artificially high errors**.

$$ERROR = \frac{d_1 + d_2 + ... + d_n}{N * MAX}$$

ERROR	Error between the prediction and the actual value of the snow ratio.
$d_1, d_2, ..., d_n$	Differences between predictions and nearest neighbor measurement.
N	Total number of predictions.
MAX	The maximum snow ratio value.

Table 4.1: Prediction error formula.

For the example in Figure 4.10, the **mean error is 23**, which means that the prediction was has an error percent of 23%.

4.3.4 Difference and Movement images

The interactive plot created in matplotlib **generates the difference and movement images between two picked plot items**.The date of the scene corresponding to the x axis is searched for in the ndsi_path_row.csv file, which is used to find the NDSI images on which the difference and movement will be made. The following figure exemplifies picking of first and second images for this calculation:

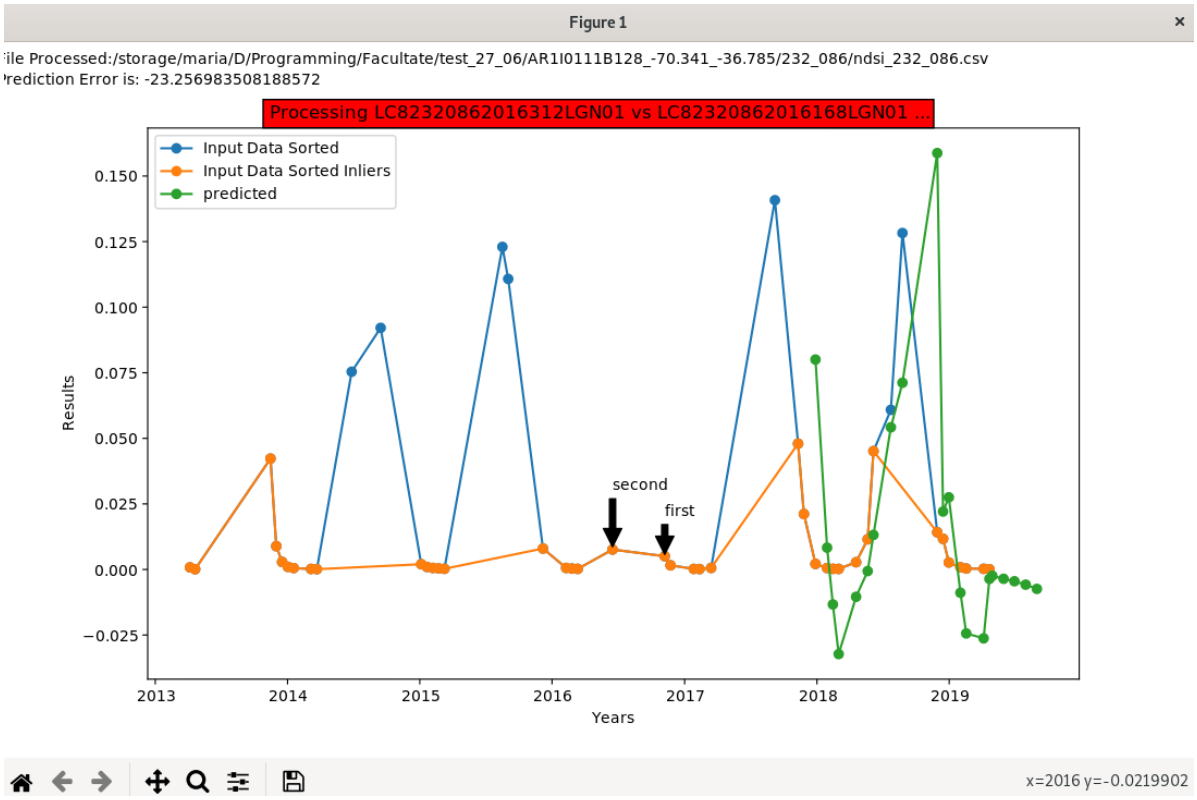

Figure 4.11: Picking two NDSI images to create their difference and movement images.

When the two items are **picked**, an **asynchronous process** calling the difference movement module with the name of the scenes, their paths and the output directory for image writing.

Difference The difference image is calculated by making a **subtraction** between the two numpy images, therefore highlighting whether snow has appeared or disappeared in comparison to the first picture.

Movement The movement image is implemented by using the **Optical flow algorithm**, which is the pattern of apparent motion of image objects between two consecutive frames caused by the movement of object. It is a **2D vector field** where each vector is a displacement vector showing the movement of points from first frame to second. [55] We use the algorithm in our image viewing because it works with assuming that:

- The **intensities** of the pixels do not change by much [55];
- The **neighbors** of the pixels show the same movement pattern [55].

Figure 4.12 shows the results of processing the first and second picked images from the plot.

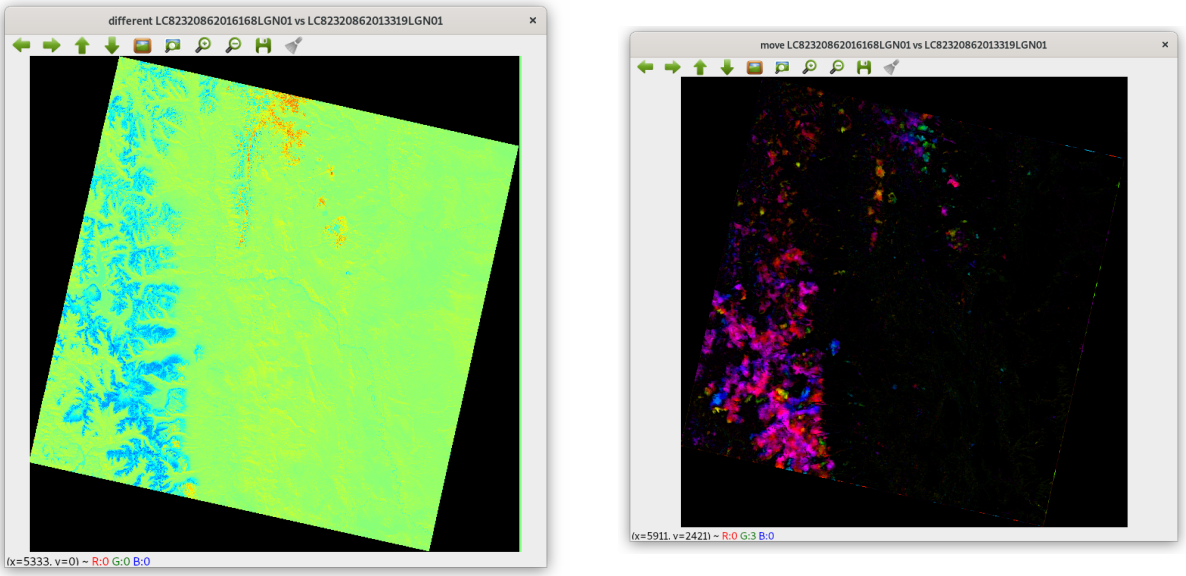

Figure 4.12: The difference and movement images for the selected points on the plot.

The results can be **saved** to disk by pressing the save button in the OpenCV GUI.

Chapter 5

Performance Evaluation

The first test area is located in Las Heras Department, Provincia Mendoza, Argentina, on the glacier with id AR1J11132208 (-32.272, -69.606), on path 233, row 82. The location has **50** pictures dating from 2013 to present, with 8 outliers. The estimation on already existing data set had an error of **21.50%**, while the extra 10 future estimates keep their direction of approaching 0.

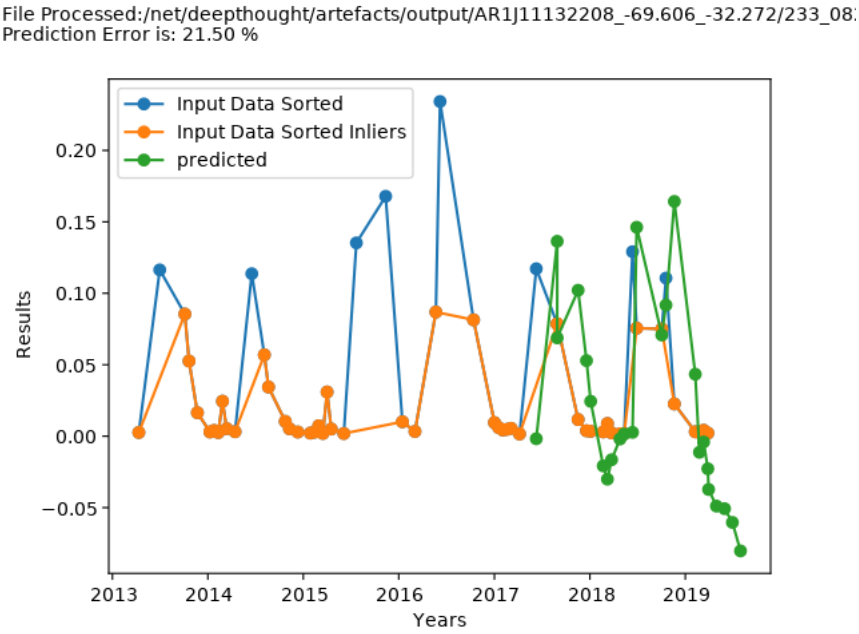

Figure 5.1: Las Heras Department, Provincia Mendoza, Argentina (AR1J11132208, 233, 82).

The second test area has the same location, on the glacier with id AR1J11132208 (-32.272, -69.606), only on a different path and row: 232, 83. The location has **61** pictures

dating from 2013 to 2019, with only 5 outliers. The estimation on already existing data set had an error of **10.16%**; the future 10 estimates also keep their movement direction predicting further melting season keeping a form of the ones before.

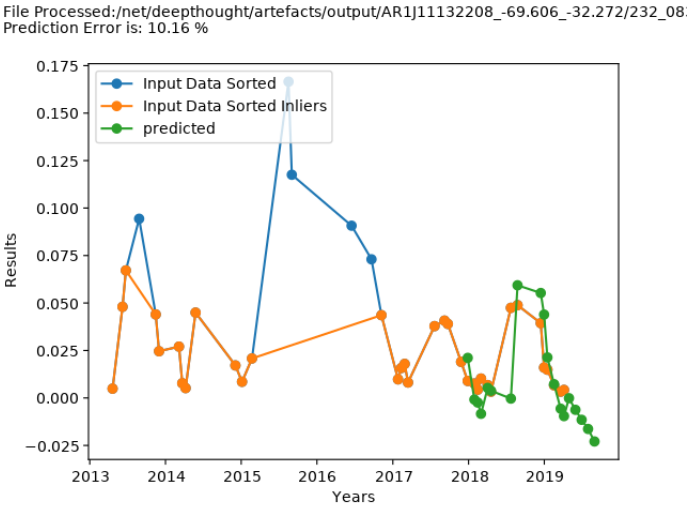

Figure 5.2: Las Heras Department, Provincia Mendoza, Argentina (AR1J11132208, 232, 83).

The third picked area is located in Panjshir, Afganistan, on glacier with id AF5Q112C0242 (35.184, 69.625) and path row 152, 36. The location has **48** pictures dating from 2013 to present, with only 5 outliers. The estimation on already existing data set had an error of **31.17%**; the future 10 estimates also keep their movement type close to the plot.

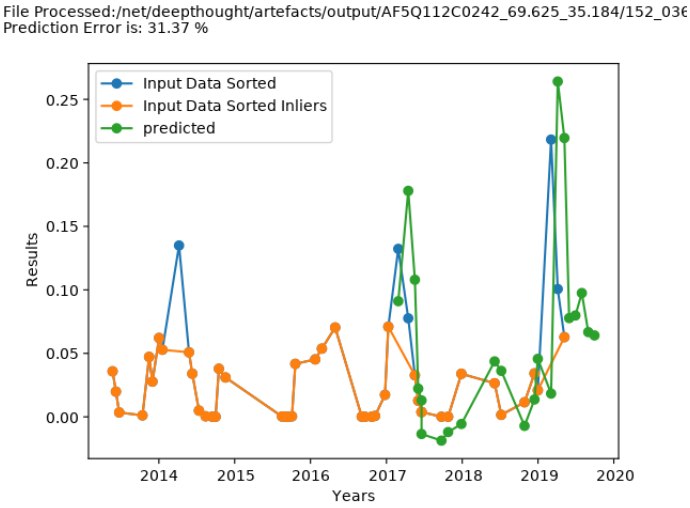

Figure 5.3: Panjshir, Afganistan (AF5Q112C0242, 152, 32).

The last tested area is located in Las Heras Department, Provincia Mendoza, Argentina, on glacier with id AR1J11132208 (-32.272, -69.606) and path row 232, 82. The location has **62** pictures dating from 2013 to 2019, with only 4 outliers. The estimation on already existing data set had an error of **4.30%**; the future 10 estimates also keep their movement falling.

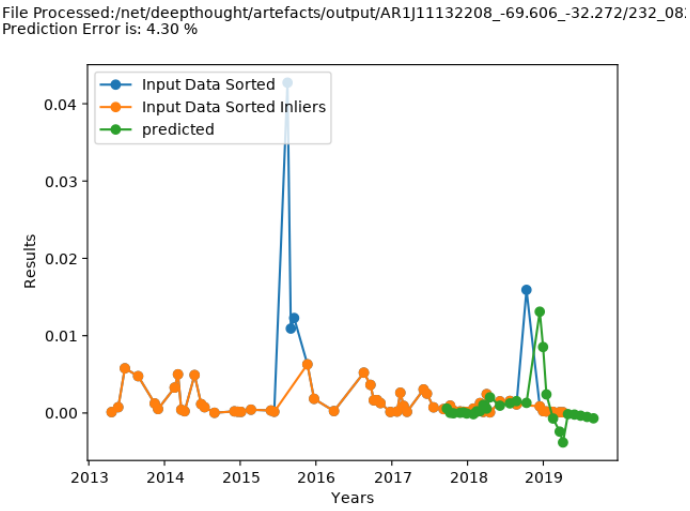

Figure 5.4: Las Heras Department, Provincia Mendoza, Argentina (AR1J11132208, 232, 82).

Chapter 6

Conclusions

Our goal was to predict glacier evolution by tracking snow coverage using satellite provided imagery. We successfully achieved this prediction with a mean error of 20%. On top of that, we also implemented the **visual tracking of glacier evolution** by aligning the images (with around **98%** accuracy rate between images and approximately **85%** one for successful alignments in a directory) which allowed us to compare the differences between images and generate motion estimation.

In conclusion, our goal was successfully achieved, even allowing us to add some **extra features** like **motion estimation** for visual data interpretation, and an **interactive plot** to display it.

From our point of view, the generated data could be useful for detecting climate change conditions, given that glaciers are the first indicatives. By doing that, we proved that this is a **valid approach** to **predict snow coverage variation**. Climate scientists could use this data in order to analyze the parameters extracted from the images in order to get a better understanding current climate change conditions in any part of the globe, and give us insights on what changes need to be made so the situation does not get out of control.

6.1 Future Development

We have created the application with two main features: NDSI ratio evolution and glacier movement estimation.

- NDSI ratio evolution prediction
- Glacier movement estimation.

The prediction could be improved by

- using other algorithms, to see which is the most appropriate to use. Some examples are the **Bayesian prediction**, which might be tested in order to testify which has better results in our case.
- testing other parameters for the ORB, Harris, RANSAC and ARIMA algorithms, since is possible that we find a combination which would create better estimations with smaller error values. In order to try parameter change the most efficient way, we can use **twiddle** [60] which allow us to **train the parameters programmatically**, thus raising the changes to find better results by much.

The motion estimation can be improved implementing the creation of a predicted image, rather than the snow ratio as a number, by updating the Optical Flow algorithm. This could be achievable by applying the movement vectors of the previous image to the next image, which would create the next move. This approach is similar with the one used on **motion-based video compression** [59]

The solutions applied in the thesis are programmatic, implemented with classic widely used algorithms in order to validate the technique. Modern algorithms like convolutional neural networks could be trained to predict the images rather than the numbers. The only problem is that we need glaciers which have valid, uncorrupted files, and more than the number we usually have (about 40). The training in such cases needs to be done with large amounts of data, which is higlhy intense and time comsuming.

Bibliography

[1] America's Climate Choices: Panel on Advancing the Science of Climate Change; National Research Council (2010). Advancing the Science of Climate Change. Washington, D.C.: The National Academies Press. ISBN 978-0-309-14588-6. "(p1) ... there is a strong, credible body of evidence, based on multiple lines of research, documenting that climate is changing and that these changes are in large part caused by human activities. While much remains to be learned, the core phenomenon, scientific questions, and hypotheses have been examined thoroughly and have stood firm in the face of serious scientific debate and careful evaluation of alternative explanations. Some scientific conclusions or theories have been so thoroughly examined and tested, and supported by so many independent observations and results, that their likelihood of subsequently being found to be wrong is vanishingly small. Such conclusions and theories are then regarded as settled facts. This is the case for the conclusions that the Earth system is warming and that much of this warming is very likely due to human activities."

[2] Intergovernmental Panel on Climate Change Fourth Assessment Report, "What is the Greenhouse Effect?" FAQ 1.3 – AR4 WGI Chapter 1: Historical Overview of Climate Change Science, IIPCC Fourth Assessment Report, Chapter 1, page 115

[3] AR4 SYR Synthesis Report Summary for Policymakers – 2 Causes of change.

[4] `https://www.un.org/sustainabledevelopment/climate-change/` (08/06/2019)

[5] `https://www.iflscience.com/environment/new-report-warns-high-likelihood-of-human-` (08/06/2019)

[6] Seiz, G.; N. Foppa (2007). The activities of the World Glacier Monitoring Service (WGMS) (PDF) (Report). Archived from the original (PDF) on 25 March 2009. Retrieved 21 June 2009.

[7] https://nsidc.org/cryosphere/glaciers/gallery/retreating.html(08/06/2019)

[8] https://en.wikipedia.org/wiki/Satellite_imagery accessed on (08/06/2019)

[9] https://en.wikipedia.org/wiki/Landsat_8 (08/06/2019)

[10] http://www.fis.uni-bonn.de/en/recherchetools/infobox/professionals/resolution/radiometric-resolution (08/06/2019)

[11] https://en.m.wikipedia.org/wiki/Landsat_8 (08/06/2019)

[12] Spectral resolution deailed https://www.geoimage.com.au/SWIR%20Series/resolution (08/06/2019)

[13] https://www.esri.com/arcgis-blog/products/product/imagery/band-combinations-for-landsat-8/ (08/06/2019)

[14] https://landsat.gsfc.nasa.gov/landsat-data-continuity-mission/ (09/06/2019)

[15] https://landsat.gsfc.nasa.gov/operational-land-imager-oli/ (09/06/2019)

[16] https://www.usgs.gov/centers/eros/science/usgs-eros-archive-landsat-archives-land qt-science_center_objects=0#qt-science_center_objects (09/06/2019)

[17] Thermal Infrared Sensor https://landsat.gsfc.nasa.gov/thermal-infrared-sensor-tirs/ (09/06/2019)

[18] https://www.gim-international.com/content/article/landsat-the-cornerstone-of-global-land-imaging (09/06/2019)

[19] https://landsat.gsfc.nasa.gov/the-worldwide-reference-system/ (09/06/2019)

[20] https://www.usgs.gov/land-resources/nli/landsat/landsat-collection-1?qt-science_support_page_related_con=1#qt-science_support_page_related_con (09/06/2019)

[21] https://www.usgs.gov/media/images/landsat-8-band-designations (09/06/2019)

[22] https://nsidc.org/data/glacier_inventory/browse.html (09/06/2019)

[23] https://www.projectorcentral.com/All-About-Bit-Depth.htm (09/06/2019)

[24] Green band from the Landsat Collection 1 Tier 1 Archive.

[25] Infrared band from the Landsat Collection 1 Tier 1 Archive.

[26] https://landsat.gsfc.nasa.gov/landsat-8/landsat-8-bands/ (10/06/2019)

[27] Normalized-difference snow index general information. https://link.springer.com/referenceworkentry/10.1007%2F978-90-481-2642-2_376 (10/06/2019)

[28] Normalized-difference snow index general information. https://eos.com/ndsi/ (10/06/2019)

[29] PyCharm cross-platform. https://en.wikipedia.org/wiki/PyCharm (10/06/2019)

[30] Python definition. https://en.wikipedia.org/wiki/Python_(programming_language) (10/06/2019)

[31] The Zen of Python Peters,Tim(19August2004)."PEP20TheZenofPython". PythonEnhancementProposals.PythonSoftwareFoundation. Retrieved24November2008. (10/06/2019)

[32] Definition of Git Scopatz,Anthony;Huff,KathrynD.(2015) .EffectiveComputationinPhysics.O'ReillyMedia,Inc.p.351. ISBN9781491901595.Archivedfromtheoriginalon7May2016. Retrieved20April2016. (10/06/2019)

[33] Linus Torvalds on Git. Torvalds,Linus(10June2007)."Re:fatal: seriousinflateinconsistency".git(Mailinglist). (10/06/2019)

[34] Linus Torvalds about Git goals. LinusTorvalds(3May2007) .Googletechtalk:LinusTorvaldsongit.Eventoccursat02:30. Archivedfromtheoriginalon28May2007 (10/06/2019)

[35] General information about Git https://en.wikipedia.org/wiki/Git (10/06/2019)

[36] Fast and scalable Git Torvalds,Linus(19October2006)."Re: VCScomparisontable".git(Mailinglist). (10/06/2019)

[37] Git large project handling. http://digitalvampire.org/blog/index.php/2006/11/16/oh-what-a-relief-it-is/ (10/06/2019)

[38] Gdal definition. https://gdal.org/ (10/06/2019)

[39] Gdal general information. https://en.wikipedia.org/wiki/GDAL (10/06/2019)

[40] NumPy definition https://www.numpy.org/ (10/06/2019)

[41] NumPy performance https://en.wikipedia.org/wiki/NumPy (10/06/2019)

[42] https://en.wikipedia.org/wiki/OpenCV (10/06/2019)

[43] https://gisgeography.com/landsat-file-naming-convention/ (11/06/2019)

[44] https://earthexplorer.usgs.gov/ (20/06/2019)

[45] https://gisgeography.com/usgs-earth-explorer-download-free-landsat-imagery/ (20/06/2019)

[46] https://landsat.usgs.gov/usgs-landsat-global-archive (20/06/2019)

[47] https://nsidc.org/data/glacier_inventory/ (20/06/2019)

[48] https://nsidc.org/data/glacier_inventory/query.html (21/06/2019)

[49] https://docs.opencv.org/3.0-beta/doc/py_tutorials/py_feature2d/py_fast/py_fast.html (21/06/2019)

[50] https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_feature2d/py_brief/py_brief.html (21/06/2019)

[51] https://docs.opencv.org/3.0-beta/doc/py_tutorials/py_feature2d/py_orb/py_orb.html (21/06/2019)

[52] https://docs.opencv.org/3.4.3/dc/d0d/tutorial_py_features_harris.html (21/06/2019)

[53] https://en.wikipedia.org/wiki?curid=1089270 (21/06/2019)

[54] https://en.wikipedia.org/wiki/Random_sample_consensus (21/06/2019)

[55] https://docs.opencv.org/3.3.1/d7/d8b/tutorial_py_lucas_kanade.html (21/06/2019)

[56] https://github.com/sat-utils/sat-search (22/06/2019)

[57] https://github.com/sat-utils/sat-search (22/06/2019)

[58] https://jiffyclub.github.io/snakeviz/ (28/06/2019)

[59] https://www.cv-foundation.org/openaccess/content_iccv_2015/papers/Walker_Dense_Optical_Flow_ICCV_2015_paper.pdf (23/06/2019)

[60] http://www.mastertheboss.com/jboss-server/jboss-script/twiddle-reference-guide?showall=1 (25/06/2019)

Appendix A

Glossary

A.1 Acronyms

CLI	Command Line Interface
OLI	Operational Land Imager
GDAL	Geospatial Data Abstraction Library
TIRS	Thermal Infrared Sensor
NDSI	Normalized-Difference Snow Index
NIR	Near Infrared
SWIR	Short Wavelength Infrared
NASA	National Aeronautics and Space Administration
USGS	United States Geologial Survey
WRS2	World Reference System-2
NSIDC	National Snow and Ice Data Center
IDE	Integrated Development Environment
MTL	Metadata File
GUI	Graphical User Interface
IT	Graphical User Interface
ORB	Oriented FAST and Rotated BRIEF)

Table A.1: Acronyms table