Bachelor Thesis

Deep-learning-based Lake Ice Detection using ESA Sentinel-1 SAR data

Pascal Imhof
May 31, 2019

Institute of Geodesy and Photogrammetry

ETH Zürich

Professorship:
Prof. Dr. Konrad Schindler

Supervisors:
Manu Tom
Dr. Emmanuel Baltsavias

ETH

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Eigenständigkeitserklärung

Die unterzeichnete Eigenständigkeitserklärung ist Bestandteil jeder während des Studiums verfassten Semester-, Bachelor- und Master-Arbeit oder anderen Abschlussarbeit (auch der jeweils elektronischen Version).

Die Dozentinnen und Dozenten können auch für andere bei ihnen verfasste schriftliche Arbeiten eine Eigenständigkeitserklärung verlangen.

Ich bestätige, die vorliegende Arbeit selbständig und in eigenen Worten verfasst zu haben. Davon ausgenommen sind sprachliche und inhaltliche Korrekturvorschläge durch die Betreuer und Betreuerinnen der Arbeit.

Titel der Arbeit (in Druckschrift):

Deep-learning-based Lake Ice Detection using ESA Sentinel 1 SAR data

Verfasst von (in Druckschrift):
Bei Gruppenarbeiten sind die Namen aller Verfasserinnen und Verfasser erforderlich.

Name(n): Imhof

Vorname(n): Pascal

Ich bestätige mit meiner Unterschrift:

- Ich habe keine im Merkblatt „Zitier-Knigge" beschriebene Form des Plagiats begangen.
- Ich habe alle Methoden, Daten und Arbeitsabläufe wahrheitsgetreu dokumentiert.
- Ich habe keine Daten manipuliert.
- Ich habe alle Personen erwähnt, welche die Arbeit wesentlich unterstützt haben.

Ich nehme zur Kenntnis, dass die Arbeit mit elektronischen Hilfsmitteln auf Plagiate überprüft werden kann.

Ort, Datum
Itingen, 31.5.2019

Unterschrift(en)
P.Imhof

Bei Gruppenarbeiten sind die Namen aller Verfasserinnen und Verfasser erforderlich. Durch die Unterschriften bürgen sie gemeinsam für den gesamten Inhalt dieser schriftlichen Arbeit.

Acknowledgements

I would like to thank the following people who supported me during my bachelor thesis:

- Prof. Dr. Konrad Schindler, for making this project possible
- Manu Tom, for his great support during each state of the thesis
- Dr. Emmanuel Baltsavias, for his support
- Silvan Leinss, for helping me with my questions in the field of radar
- Roman Imhof and Manuela Scheidegger for proof-reading

Abstract

The goal of this bachelor thesis is to train a deep neural network to detect lake ice on satellite radar images. The images used are Synthetic Aperture Radar (SAR) data of the Sentinel-1 satellite constellation of the Copernicus program of ESA which have been downloaded (pre-processed) from Google Earth Engine. The area of interest is the region of Sils in the canton Graubünden in Switzerland where the three alpine lakes, Lake Sils, Lake Silvaplana and Lake St. Moritz are located. Similar studies with optical satellite images and webcams have been done in the past. The ground truth was generated before by manual interpretation of webcam images. A Deeplab v3+ model was established and trained to detect lake ice/snow coverage and to distinguish it from water.

In summary, this thesis shows that it is possible to train the neural network for the detection of lake ice on the investigated lakes. The predicted precision is not yet at the desired level. It has been shown that there is a dependency of the results on the orbit (incidence angle) and possibly also on the different surface structures (smooth water, water with waves, snow-free ice, ice covered with snow, ice covered with wet snow). Due to the orbit dependence and the revisit time of the satellite the data sets are very small and a further in-depth analysis with additional data (from several winter periods) and more lakes is recommended.

Table of Contents

1 Introduction

The Global Climate Observing System (GCOS) was founded in 1992 by the World Meteorological Organization with the goal of providing accurate, sustained and freely accessible climate observation data for research and education. The GCOS defined Essential Climate Variables (ECVs) for the observation of the climate change. One of them is *lake ice* and it is accepted that analyzing this climate variable is significant for climate change research and global warming studies ("GCOS," 2019).

The goal of this bachelor thesis is to train a deep neural network to detect the lake ice using satellite radar imagery. The necessary images are freely available Synthetic Aperture Radar (SAR) data of the Sentinel-1 satellite constellation of the Copernicus program from the European Space Agency (ESA). The data have been downloaded (pre-processed) from the Google Earth Engine platform. Using Deeplab v3+ (Chen, Zhu, Papandreou, Schroff, & Adam, 2018), a model was trained to detect ice/snow coverage on lakes and to distinguish it from water. The used radar images were chosen from the period between 1st September 2016 to 31st May 2017. The used Sentinel-1 data are images of product level 1 in GRD full resolution mode. The area of interest was defined as the region of Sils in the canton Graubünden in Switzerland. In this region, there are the lakes, Sils, Silvaplana and St. Moritz, as shown in Fig. 1.

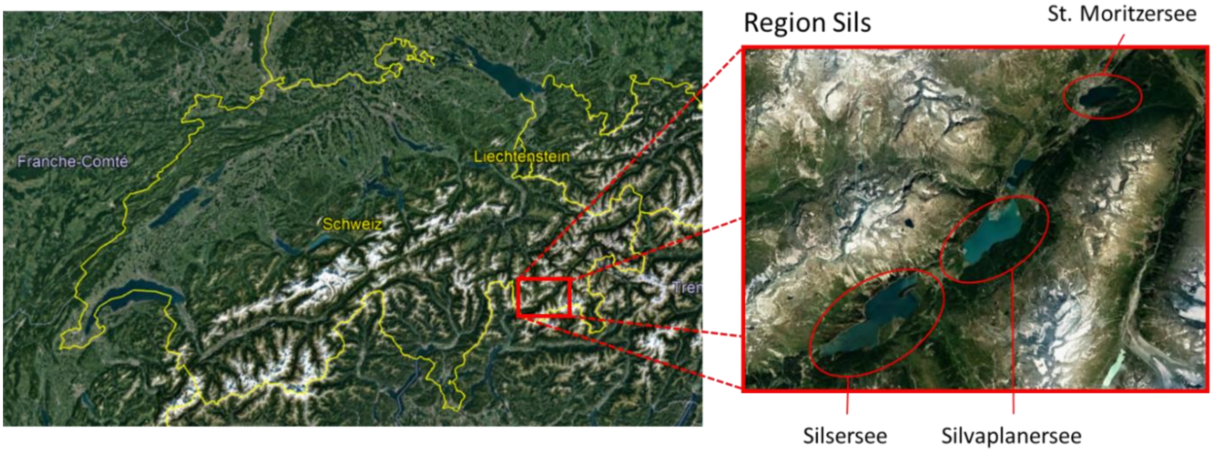


Figure 1. Map of Switzerland and detail sector map of the three lakes of interest, Lake Sils, Lake Silvaplana, and Lake St. Moritz (source: Google earth).

Tab. 1 summarizes some geographical parameters of the three lakes ("Alle Seen der Schweiz," 2019). The lakes are comparably small and are located in an Alpine environment. They are also known to freeze completely during the winter months every year.

Table 1. Key parameters of the lakes.

Name	Area (km^2)	Depth (m)	Altitude (m)
Lake Sils	4.1	71	1797
Lake Silvaplana	3.1	78	1791
Lake St. Moritz	0.78	44	1768

This thesis is motivated by a project titled 'Integrated lake ice monitoring and generation of sustainable, reliable, long time series to identify suitable sensors and processing methods for automatic ice monitoring on Swiss lakes', initiated by the Swiss Federal Office of Meteorology and Climatology (Meteo Swiss). One of the outcomes of the first phase of the project was a machine learning-based methodology using low resolution optical satellite images. The main problem with optical satellite images is the data loss due to cloud coverage. However, they could show that the algorithm produces consistent results when tested on data from multiple winters (Tom, Kälin, Sütterlin, Baltsavias, & Schindler, 2018). Second major outcome was a neural network-based methodology utilizing data from webcams. The overall per-pixel accuracies for the tested data set exceeded 95%. Though the approach generated excellent results, one practical problem with webcams is that it is very expensive to make this technology operational on a country- or world-scale. In addition, multiple webcams are needed to get full coverage of a lake, especially in the case of big lakes (Muyan et al., 2018).

This thesis targets to use radar data. Radar has the big advantage, that there is no interference by bad weather or clouds. With radar it is possible to look through the clouds and also to observe the lakes during the night (Albertz, 2009).

This thesis is structured as follows. Chapter 2 gives a short overview about related work, followed y chapter 3 which describes the most important theoretical principles. In chapter 4 the applied methodology is described. Chapter 5 summarizes the results of the different experiments, which are then discussed in chapter 6. The conclusion and suggestions for future work are described in chapter 7.

2 Related Work

Detecting lake ice with a neural network is not a new idea. There are already studies done which combine lake ice detection with machine learning and they proof that it is possible to precisely predict lake ice with a neural network. One recent publication used low-resolution optical satellite images (Tom et al., 2018). The authors worked with optical satellite images from MODIS and VIIRS and trained a neural network to detect lake ice coverage on lakes in Switzerland. The input images had a low resolution of 250m – 1000m. The authors established a simple classification approach which worked with high accuracy for the tested lakes. They also state that the methodology is generic and should be applicable to lakes in other regions as well. However, one big disadvantage of this technique is the dependency on good weather conditions without clouds.

A second recent publication used webcams to detect lake ice with a neural network (Muyan et al., 2018). Semantic segmentation maps could be obtained based on the application of a neural network on conventional RGB webcam images. A classification system with four labels (water, ice, snow, clutter) was introduced and segmentation accuracies of >95% have been reported. The authors found that ice was the most difficult to predict. Based on the simplicity of the system (webcams can be found meanwhile in a lot of places) this approach is believed to have a big potential for lake ice detection. However, cost considerations and the need to cover the entire lake area with webcams might be substantial challenges to apply this methodology in larger areas.

An interesting approach to detect lake ice by using SAR data was published in 2018 by Wang et.al. (J. Wang, Duguay, Clausi, Pinard, & Howell, 2018). They used dual polarization RADARSAT-2 imagery and introduced a region-based classification algorithm for detecting lake ice. With this approach they could predict frozen/non-frozen days with an overall accuracy of 90.4%.

A recent important development in regards of advancing deep learning was recently published by Chen et al who applied an encoder-decoder structure with Atrous separable convolution for semantic image segmentation (Chen et al., 2018). They achieved with an encoder-decoder structure and a spatial pyramid pooling module in combination with Atrous convolution a remarkable prediction performance of 89.0% on the PASCAL VOC 2012 data set and 82.1% on the Cityscape dataset without any postprocessing.

3 Theoretical Principles

The following chapter summarizes the relevant background information and theoretical principles which are the basis for the better understanding of this bachelor thesis. The chapter will cover the essential information about the ESA Sentinel-1 satellite program, some theoretical background about the applied SAR radar system, information about satellite data formats provided by ESA, information about the applied data pre-processing steps by Google Earth Engine, some basic theory about radar interaction with different surfaces, and finally a short introduction to deep learning principles.

3.1 ESA Sentinel-1 Satellite Program

3.1.1 Sentinel-1

This chapter is based on Sentinel-1 - Mission Summary (2019), Sentinel-1 - ESA EO Missions (2019), Sentinel-1 - Overview (2019) and Overview Copernicus (2019).

Sentinel-1 is the first satellite constellation of the Copernicus program of ESA started in 2014 ("Sentinel-1 - ESA EO Missions," 2019). The program plans to have at the end six satellite families in the orbit whereof three of them have been launched already. The goals of the program are to monitor and to map the surface of the world. The produced data can be used in for example for emergency mapping, iceberg monitoring, ice condition forecasting at sea, ship detection, climate change monitoring and many other applications. All data from the Copernicus program are free of charge and can be downloaded after a free registration at the ESA download portal (ESA Open Access Hub, 2019).

Today, there are two Sentinel-1 satellites (Sentinel-1A and Sentinel-1B) operational in space. Sentinel-1A was launched in April 2014, Sentinel-1B two years later in April 2016. They are both on the same sun-synchronous near polar orbit. They are displaced by 180° degrees. The lifetime of these two satellites is planned to be between 7 and 12 years. After this time, they shall be replaced by two new Sentinel-1 satellites (Sentinel-1C and Sentinel-1D). The satellites have a C-SAR system on board which is mounted at the right side of the satellite (referenced to flight direction). In contrast to the optical detection systems, a radar allows the monitoring through clouds and to scan the earth during night and bad weather. The Sentinel-1 satellite has a repeat cycle of 12 days at the equator, giving a cycle time of 6 days with two Sentinel-1 satellites. During these 12 days the satellite is orbiting the earth within 98.6 min on an altitude of 693 km. The inclination is 98.18°. The same point on earth is mapped within one repeat cycle several times and the geographical scanning pattern is shown in Fig. 2 (status: October 2016). The region of Sils (Graubünden, Switzerland) is visible on 4 different orbits (numbered 15, 66, 117, 168) during one repeat cycle. Sentinel-1A takes images of the region of Sils in a 4/3/4/1 cycle

- Day 4 from Orbit 15 (at about 5pm)
- Day 7 from Orbit 66 (at about 5 am)
- Day 11 from Orbit 117 (at about 5 pm)
- Day 12 from Orbit 168 (at about 5 am)

Fig. 2 shows the spatial revisit frequency of the two satellites and indicates also the pass direction (ascending, descending). As it is an ESA mission, the focus area is Europe. The red and green lines in Fig. 2 describe the orbits over the surface. During the period of interest (1.9.2016 – 31.5.2017), the second Sentinel-1 satellite (1B) was launched and doubled the number of available images of the region towards the end of the period (March 2017) giving images of the region on every 1-3 days.

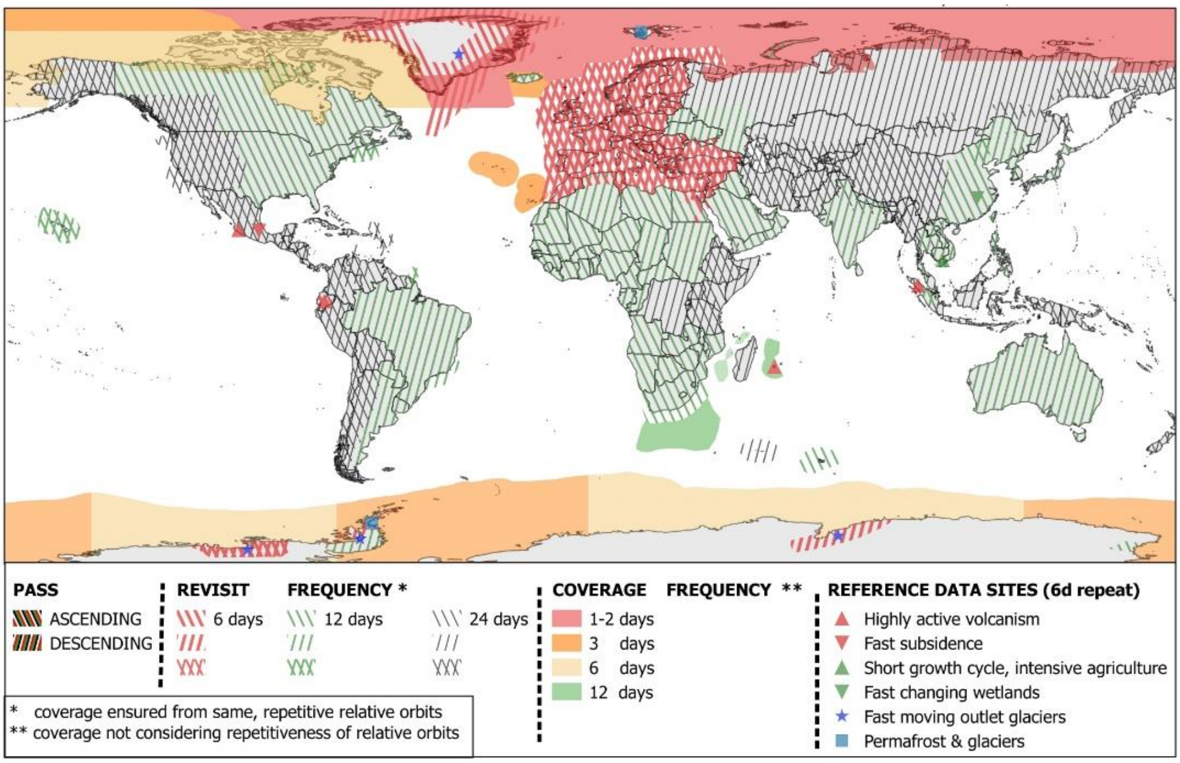

PASS
ASCENDING
DESCENDING
REVISIT
FREQUENCY *
6 days
12 days
24 days
COVERAGE
FREQUENCY **
1-2 days
3 days
6 days
12 days
REFERENCE DATA SITES (6d repeat)
Highly active volcanism
Fast subsidence
Short growth cycle, intensive agriculture
Fast changing wetlands
Fast moving outlet glaciers
Permafrost & glaciers
* coverage ensured from same, repetitive relative orbits
** coverage not considering repetitiveness of relative orbits

Figure. 2. Constellation Observation Scenario : Revisit & Coverage Frequency for 10/2016 (source: ("Sentinel-1- Observation Scenario," 2019).

3.1.2 Synthetic Aperture Radar (SAR)

This chapter is based on Moreira et al. (2013) if nothing else is stated. The Sentinel-1 satellites use SAR systems. Like a normal radar, this system sends out electromagnetic waves in the GHz range and detects the backscattered echoes of the surface. The big difference to a normal side looking radar is the higher azimuth resolution. A SAR-system can use different frequency bands. The decision which frequency band is used is determined by the observation goals. Each band interacts differently with the surface. Tab. 2 gives an overview about the different available frequency bands. Sentinel-1 operates in the C-frequency band. SAR sensors are using frequency modulated pulse waveforms. That means that the amplitude stays constant, but the frequency is varied. From the reflected signal the SAR sensor measures the amplitude and the phase. A SAR system changes in predefined cycles between transmitting and receiving mode. If the received signals are combined in the right way, it is possible to generate a virtual aperture which is much bigger (based on a given antenna size) and allows to build satellites with smaller antennae. Smaller antennae have also a bigger opening angle which allows to observe the ground on the surface for a longer time than with a normal side looking radar system (Albertz, 2009).

Table 2. Different frequency bands with frequency and wavelength (Moreira et al, 2013).

Frequency Band	Frequency [GHz]	Wavelength [cm]
Ka	40-25	0.75-1.2
Ku	17.6-12	1.7-2.5
X	12-7.5	2.5-4
C	7.5-3.75	4-8
S	3.75-2	8-15
L	2-1	15-30
P	0.5-0.25	60-120

As already mentioned, the big advantage of SAR compared to a normal radar system is the higher azimuth resolution. For a normal side looking radar system the azimuth resolution is calculated with the Eq. (*1)*.

$$\delta_a = \frac{\lambda}{d_a} * r \qquad (1)$$

δ_a = azimuth resolution

λ = wavelength

d_a = antenna length

r = range distance (distance between satellite and surface point)

The range distance is a measure of the distance between the satellite and the ground and is for a satellite system like Sentinel-1 as stated before in the range of 690 kilometers. For a good resolution (small value), the antenna length must therefore be big (at a given wavelength), which is technically and economically not feasible.

Moreira et. al. show (p. 10) that in contrast to a normal radar system, the azimuth resolution of a SAR system can be calculated as in Eq. (*2)*. The derivation of the equation can be found in (Moreira et al, 2013)

$$\delta_a = \frac{d_a}{2} \qquad (2)$$

δ_a: azimuth resolution

d_a: antenna length

Eq. (*2)* shows that the azimuth resolution in a SAR system depends only on the antenna length. The smaller the antenna, the better is the azimuth resolution.

3.1.3 Generation of an image out of raw SAR-Data

The following paragraph is based on Moreira et al. (2013, p.10-11). In contrast to visual sensor data, the raw SAR data are not interpretable directly. Fig. 3 summarizes the necessary transformation process schematically. There are two different, consecutive filter operations used, one in the range and the second one in the azimuth direction. This filter operations are done in the frequency domain and deliver an image which is compressed in the range and the azimuth direction and contains the information about the distance between the satellite and the ground points in the form of reflection intensity data. Five additional correction steps have to be added afterwards in the Google Earth Engine to get suitable, interpretable data (described in chapter 3.2).

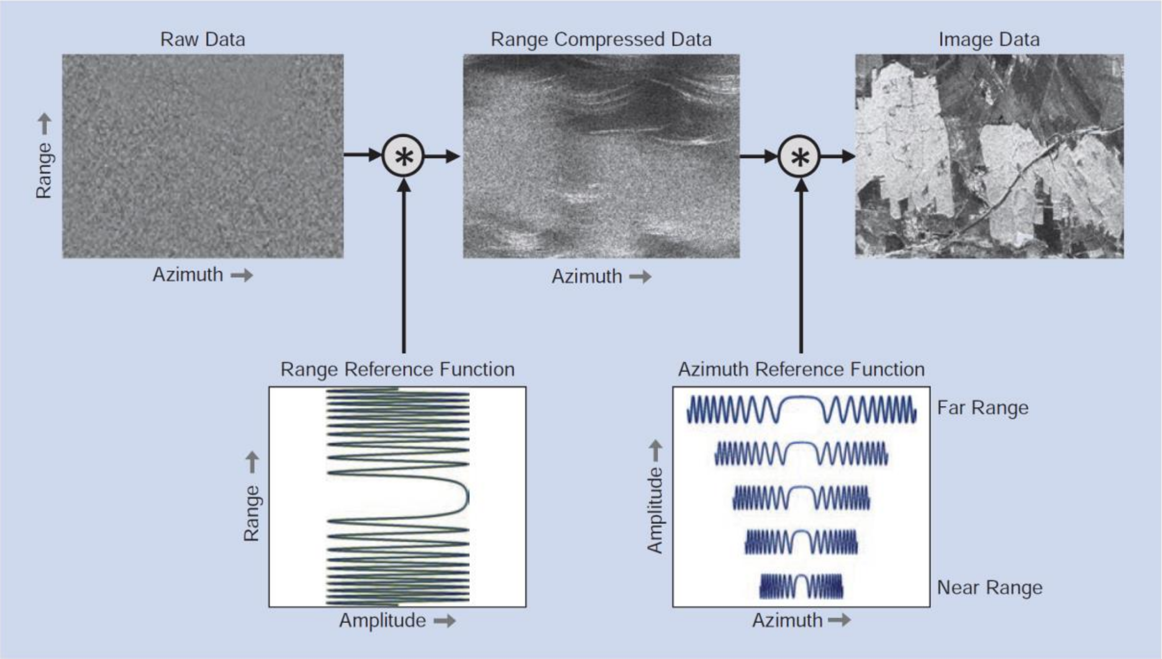


Figure 3. Schematic overview about radar signal processing (source: (Moreira et al, 2013), p. 11)).

3.1.4 Observation Scenario

This paragraph is based on Sentinel-1- Observation Scenario (2019) and Sentinel-1 - Instrument Payload (2019) if nothing else is quoted. There are four different observation modes used by the Sentinel-1 satellites. These modes differentiate specifically in their area coverage and the spatial resolution and are visualized in Fig. 4.

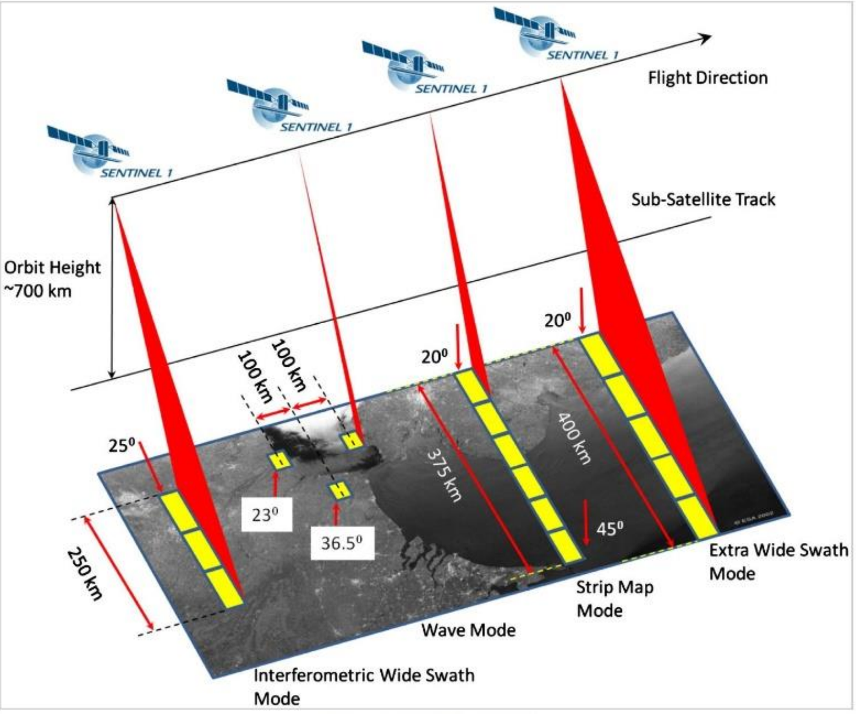


Figure. 4. Different observation modes of the Sentinel-1 satellites (source: ((Moreira et al, 2013), p.10)) .

These modes are:

- Interferometric Wide Swath Mode (IW)
    - This is the Sentinel-1 satellite default mode over landmass. In this mode the geometric resolution is 5 x 20 m. The swath of this mode contains 3 sub swaths. This mode is running with TOPSAR technique.
- Extra-Wide Swath Mode (EW)
    - This mode is used over maritime, ice and polar zones. The spatial resolution is 20 x 40 m. The swath of this mode contains 5 sub swaths. It is running with a TOPSAR technique.
- Strip Map Mode (SM)
    - One strip consists of six sub swaths. All six sub swaths together have a length of 375 km. The spatial resolution of this mode is 5 x 5 m. This mode is used for continuing the ERS and Envisat mission.
- Wave Mode (WV)
    - In this mode it is possible to determine sea wave height and wave length and it is used only over open seaside. In contrast to the three other modes, WV mode is a discontinuous observation which uses distinct, well separated spots of the size of 20 x 20 km, measured with different beam angle, to generate the desired information.

In addition to these four different observation modes the Sentinel-1 satellites can switch between four polarization modes which are defined by a two-letter code. The first letter represents the polarization of the outgoing signal, the second letter the polarization filter for the incoming signal. V represents vertical polarization, H the horizontal polarization. The four possible combinations are: VV, VH, HH and HV.

The penetration depth of the radar wave varies with the polarization chosen. Therefore, polarization may provide information on the topology and orientation of small elements that compose the observed surface (Albertz, 2009). Fig. 5 gives an overview on the different standard observation scenarios (combinations of observation mode and polarization mode) used over the different areas of the globe. Over the region of Sils the polarizations VV and VH are used in combination with the standard IW observation mode.

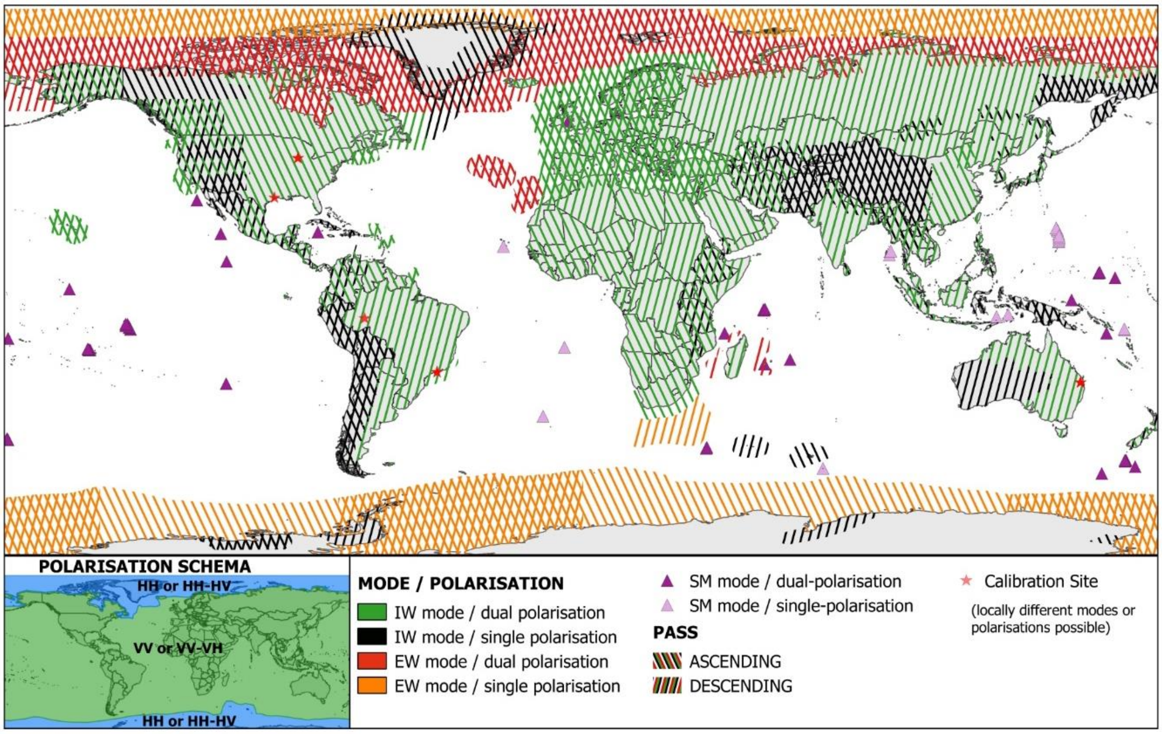

POLARISATION SCHEMA
HH or HH-HV
VV or VV-VH
HH or HH-HV
MODE / POLARISATION
🟩 IW mode / dual polarisation
⬛ IW mode / single polarisation
🟥 EW mode / dual polarisation
🟧 EW mode / single polarisation
▲ SM mode / dual-polarisation
▲ SM mode / single-polarisation
PASS
▨ ASCENDING
▨ DESCENDING
★ Calibration Site
(locally different modes or polarisations possible)

Figure 5. Constellation Observation Scenario : Mode – Polarisation – Observation for 10/2016 (source : ("Sentinel-1- Observation Scenario," 2019).

3.1.5 ESA SENTINEL-1 data products

This chapter is based on Sentinel-1 - Data Products (2019) if nothing else is quoted.

ESA offers the data form Sentinel-1 on www.scihub.copernicus.eu. This web portal offers different customized data product levels from which one can choose.

- Level 0
  - Raw SAR data. If you want to use these data, they have to be preprocessed.
- Level 1
  - This level is the most popular level. There are two product types available:
    - Single Look Complex (SLC): Each pixel is represented by a complex number which contains the whole phase and amplitude information. These images are georeferenced using the orbit information and the altitude of the satellite ("Sentinel-1 - Single Look Complex," 2019, p. 1).
    - Ground Range Detected (GRD): This product contains data projected to the ground using an earth ellipsoid. This product has no phase information anymore and has a nearly square spatial resolution. There are different resolutions available: Full Resolution (FR), High Resolution (HR) and Medium Resolution (MR).
- Level 2
  - This product contains the information about the Ocean Swell Spectra (OSW). Also included in this level is the Surface Radial Velocity (RVL).

A detailed overview on the different offerings can be found in Fig. 6.

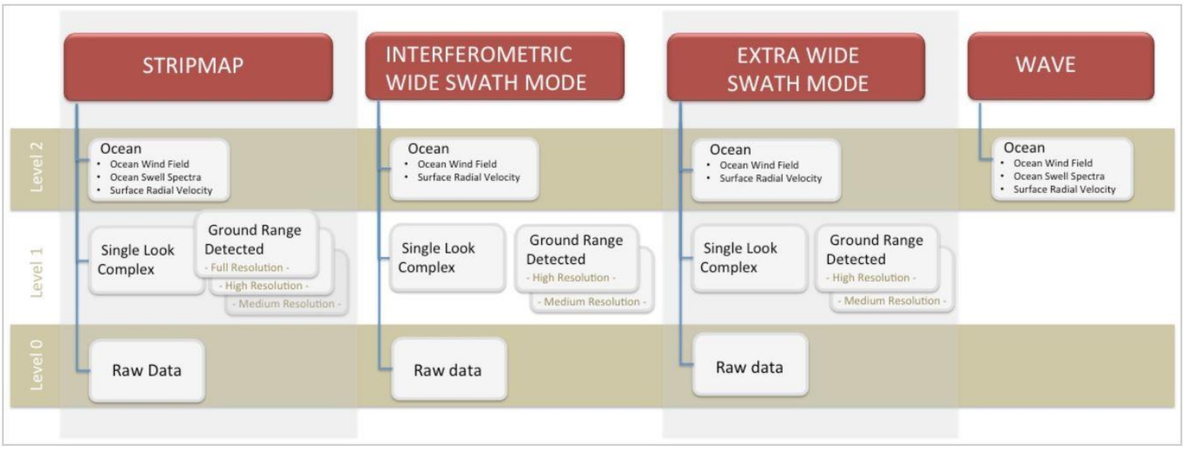


Figure 6. Overview about the different product levels of the different modes (source: ("Sentinel-1 - Data Products," 2019).

3.2 Google Earth Engine

Google Earth Engine is a planetary-scale platform for Earth science data & analysis. It stores and provides datasets of satellite missions, preprocesses them and makes them freely accessible for education and research. One of the slogans in the advertising video of Google Earth Engine summarizes it very well:

Letting scientist focus on data instead of downloading & managing data (Google Earth, 2017)

There is a web application with which some first, basic screening and analysis work can be directly done. For further in-depth analysis, it is possible to download the data via Google Drive which supports and provides GeoTiff and TFRecord formats.

For Sentinel-1 the image set is called 'COPERNICUS/S1_GRD'. This dataset contains Level-1 Ground Range Detected (GRD) images which are preprocessed from Google and provide backscatter coefficients in decibels. Google Earth Engine provides for Sentinel-1 data the following five preprocessing steps with the Sentinel-1 Toolbox from ESA (Google Earth Engine, 2019):

- Apply orbit file
  - Changes the orbit file with a corrected orbit file
- GRD border noise removal
  - Corrects the noise at the border of the images
- Thermal noise removal
  - Corrects the thermal noise between the sub-swaths
- Radiometric calibration
  - Calculates the backscatter intensity using the GRD metadata
- Terrain correction
  - Corrects the side looking effects using the SRTM 30 meters digital elevation model

The thermal noise removal, radiometric calibration and terrain correction are explained in more detail in the following sections.

3.2.1 Thermal noise reduction

This chapter is based on Albright (2002).

Thermal noise in satellite radar data originates mostly from the heat generated by the satellite itself. It can become a problem if the backscattered radar signal is of low intensity (‘dark image’) and this noise has to be removed to give reliable information. There are two different thermal noise corrections applicable.

- Empirical thermal noise removal
- Theoretical thermal noise removal

An empirical thermal noise correction is based on the assumption that highly reflective target areas, for example water surfaces without any disturbances, are not sending back any radar signals. Noise signals in these areas are then considered thermal noise and are deducted from the whole image. The advantage of this procedure is that it is very easy and produces comparable results. On the other hand, in most cases there are only a few suitable target areas available and in the end it remains empirically with a lot of uncertainties.

In contrast, a theoretical noise correction is using the whole set of metadata from the radar images. The advantage of this procedure is, that any image can be used. The results of this correction are also comparable with other images, which are corrected with this method. The theoretical correction is based on Eq. (*3*):

$$N_\sigma = n_r * 30 * \log\left(\frac{r_s}{r}\right) - 2 * G_r + 10 * \log(\sin(I)) \qquad (3)$$

- n_r = noise reference level
- r_s = slant range
- r = reference range
- G_r = Antenna pattern correction
- I = incidence angle

A theoretical thermal noise reduction has been applied by Google Earth Engine to all data used in this thesis. It is stated that the preprocessing tools of the Sentinel-1 Toolbox from ESA have been used.

3.2.2 Radiometric calibration

This chapter is based on the ESA Document Radiometric Calibration of S-1 Level-1 Products Generated by the S-1 IPF (Miranda, 2015).
The radiometric calibration transforms the radar reflectivity to physical units (for example decibel). The radar reflectivity contains a real and an imaginary part. The Level 1 products of Sentinel 1 contain Calibration Annotation Data Sets (CADS). Four Look Up Tables (LUT) can be found in these datasets. With one of them it is possible to transform the radar reflectivity A_{σ} into the radar cross-section σ^{0}. After this correction the area is normalized and is aligned with the ground range plane. In the CADS an average height is used on an ellipsoid earth model. With this ellipsoid at hand, the area normalization factor can be simplified to sin(α) for the radar cross-section. Eq. (*4)* is needed for the calculation of the radar cross-section σ^{0}:

$$\sigma^0 = \frac{DN^2}{A_{dn}^2 * K} * \frac{1}{G_{eap}^2} * \left(\frac{R}{R_{ref}}\right)^3 * \sin(\alpha) \qquad (4)$$

- $\frac{1}{G_{eap}^2}$ elevation atenna pattern (EPA) correction
- $\left(\frac{R}{R_{ref}}\right)^3$ range spreding loss (RSL) correction
- A_{dn} is the product final scaling from internal SLC to final SLC ord GRD
- α local incidence angle of the used earth model
- K calibration constant
- DN is the pixel amplitude directly taken from the measurement file

In Sentinel-1 data the EPA and the RSL corrections are by default already applied. With this Eq. 4 it simplifies to the Eq.(*5)*:

$$\sigma^0 = \frac{DN^2}{A_{dn}^2 * K} * \sin(\alpha) \qquad (5)$$

If the Look up Tables (LUT) are used Eq. (*5)* can be simplifeyd to the Eq. (*6)*:

$$\sigma^0 = \frac{DN^2}{A_{\sigma}^2} \qquad (6)$$

A radiometric calibration has been applied by Google Earth Engine to all data used in this thesis. It is stated that the preprocessing tools of the Sentinel-1 Toolbox from ESA have been used.

3.2.3 Terrain correction

The following section is based on Canada Natural Resources (2008).

The terrain correction removes distortion effects originating from the side looking geometry of the radar system. This is necessary because otherwise the radar images could not be used in combination with other georeferenced products as for example a shapefile. A short summary about three important distortion effects due to the side looking geometry of a SAR are listed below:

- Slant-range scale distortion:
  - This effect is visualized in Fig. 7 at the left. The distances on the ground between a-b and c-d have the same length. However, in the radar image the distance c-d will be longer than the distance a-b. The reason is that radar measures distances over time differences. The red lines in Fig. 7 are symbolizing the front of a radar wave, Δt_1 is smaller than Δt_2. That means that the distance for the radar wave traveling to a-b will be shorter and the wave will be reflected earlier at the surface leading to a shorter total time between transmitting and receiving the radar wave package, resulting in a shorter ab distance.
- Relief displacement
  - Foreshortening:
    This effect is caused by the surface of the earth. In Fig. 7 in the middle the effect is visualized. The distance a-b on the slope of the mountain will be measured as zero. The radar wave will be reflected at a and b at exactly the same time. If the slope of the mountain is not parallel to the radar wave front (as visualized between c and d, the distance will be measured as too short because of the slant-range distortion (see point above).
  - Layover:
    In a layover the radar wave front hits the top of the mountain first. The radar reflection at the top of the mountain will be reflected before the signal reaches the ground of the mountain. This means as shown in Fig. 7 at the right, that for point b, the horizontal distance to the satellite is shorter than the horizontal distance of point a. In the radar image the slope of the mountain lays above the bottom (point a) of the mountain.

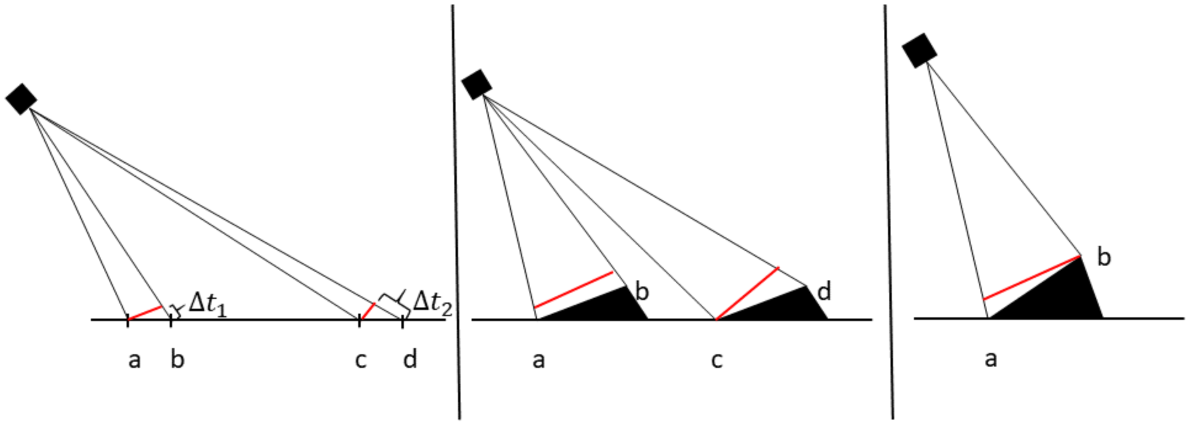


Figure 7. Left: Slant range scale distortion, middle: Foreshortening, right: Layover

All data used in this thesis have been terrain corrected by the algorithms of Google Earth Engine. It is stated that the preprocessing steps of the Sentinel-1 Toolbox from ESA have been used.

3.3 Radar Backscattering at different surfaces

This chapter is based on Albertz (2009) and SAR Instrument - Sentinel-1 SAR Technical Guide - Sentinel Online (2019) if nothing else is quoted.

The backscattering depends on a variety of factors which can be grouped into two main categories:

- Parameters of the emitting and detecting Sentinel-1 sensor
    - Wave length
        - The Sentinel-1 sensor works with a frequency of 5.405 GHz. This corresponds to a wavelength of 5.54 cm.
    - Incident angle
        - The incident angle varies between 20° and 46° degrees.
    - Polarization
        - For the region of Sils, Sentinel-1 polarizes the emitted and detected signals either in the VV or in the VH mode.

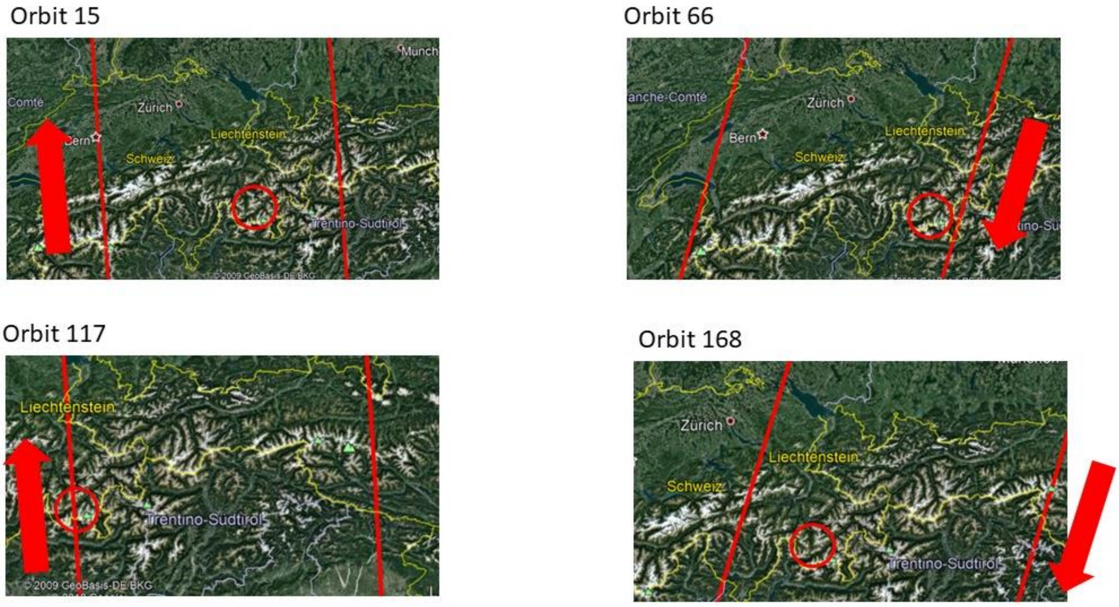
- Parameters of the surface
    - Geometrical factors of the surface
        - The roughness of the surface has a big influence on the backscatter of the radar signal.
        - The influence of the landscape topology has to be considered as well and has already been described in the chapter about terrain correction.

    - Physical factors of the surface
        - The permittivity of the material on the surface has a big influence on the backscattered signal and will be described in more detail in chapter 3.3.3

Three factors, which could have some importance for the results of this thesis, are discussed in more detail in the following three short sub chapters.

3.3.1 Incidence angle of the radar beam

As described before, Sentinel-1 orbits the earth within 12 days on a multitude of different orbits. 4 specific orbits (orbit numbers 15, 66, 117 and 168) are imaging the region of Sils (red circle on all four images in Fig. 8) on a regular basis. It is obvious that depending on the orbit and the flight direction (bold red arrow) of the satellite, the position of the Region of Sils in the measured area (between the two red lines) is always different relative to the sensor position of the satellite with orbit 66 and 117 being very close or at the edge of the observed region.



Orbit 15

Orbit 66

Orbit 117

Orbit 168

Figure 8. Different orbits and their surface covers, red arrow: flight direction of the satellite, red lines: view field of the satellite, red circle: region of Sils (source: Google Earth).

Fig. 9 shows a close up of orbit 117 in the region of Sils. The satellite moves from bottom to top and measures to the right. On the figure on the left there is an optical image of the region from Google Earth Pro. On the right side is the corresponding radar image of the same region which was downloaded from Google Earth Engine. It has to be checked if this measurement in orbit 117, which is clearly different to the orbits 15 and 168, has an influence on the recorded radar data and later on the Deep Learning experiments. A small incidence angle of the radar wave leads to more backscatter at a smooth surface (lake water) and could lead to a brighter tonality of the radar water image compared to an image taken from the same lake under a much bigger incidence angle.

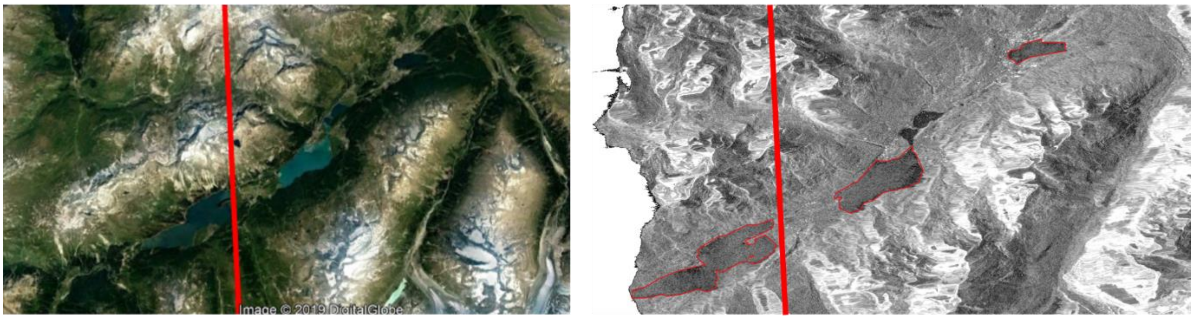


Figure 9. left orbit in Google earth, right: radar image of orbit 117 (source left: Google Earth, source right: Google Earth Engine).

3.3.2 Roughness of the surface

Fig. 10 shows two satellites on two different orbits imaging the same spot on a surface. The incident angle α of orbit 1 (red) is bigger than the incident angle β of orbit 2 (green). If the surface at the spot is now rough (right hand side) the two radar beams are backscattered in a completely different way leading to different images of the same spot.

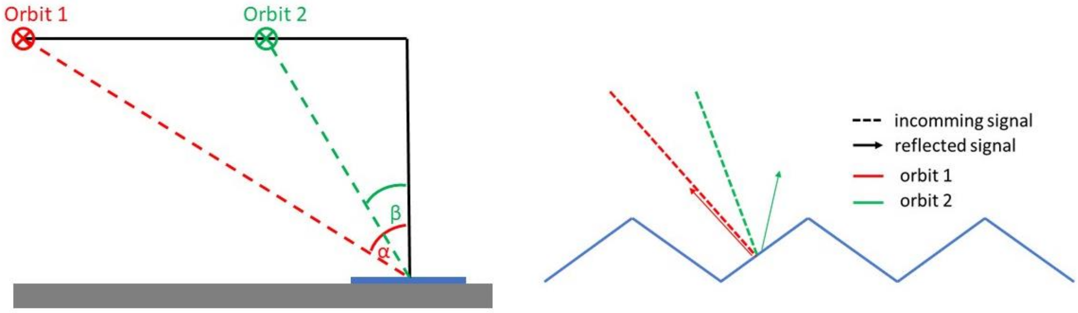


Figure 10. Orbits with different incident angle.

Roughness is a relative measure depending on wavelength and incidence angle. The ratio between the emitted wavelength of the satellite radar and the surface structure is important. If this ratio is small and the incidence angle is big, the radar signal obeys the law of reflection and nearly no signal is backscattered (the image appears dark like in the case of a smooth water surface of a lake). If the surface structure has dimensions comparable to the incidence wavelength. There is a lot of diffuse reflection in all directions and with this a higher amount of backscatter. Wind-stress and current are able to roughen the surface of a lake leading to much higher backscatter at the same spot on the surface (the tonality of the image is much brighter) (Albertz, 2009). In a region like the Engadin where down valley winds are frequently seen this influence has to be at least checked.

3.3.3 Permittivity of the surface material

This chapter is based on Schwaizer (2017).

Radar waves are electromagnetic waves. Electrical properties of surface materials determine therefore their interaction with electromagnetic waves. In this thesis, the interaction of the different water forms (ice, dry snow, wet snow, water) with radar microwaves might play an important role and have to be taken into account in the interpretation of the results. The backscatter signal will be influenced by the different water forms and with this, the input data for the neural networks.

The permittivity depends on several parameters:

- Physical state of the material
  - In this thesis the investigated material is water in its physical states snow, ice, and liquid water. Depending on the physical state the molecular interaction patterns between the single water molecules differ considerably. This leads to differences in the dipole characteristics of the molecules and therefore influences the interaction with electromagnetic waves.
- Temperature
  - With increasing temperature, the ability of the water molecules to move and absorb vibrational and rotational energy becomes higher. That leads to a different electrical behavior of the material and therefore changes the behavior with electromagnetic waves as well.
- Frequency of the used radar waves
  - Depending on the frequency of the radar waves the interaction with the molecules of the material is different.

Based on the listed dependencies above, it is obvious that the permittivity in combination with radar waves is a very complex topic. For simplicity, the following interaction schemes between the different physical states of water and the electromagnetic waves of a microwave radar beam are assumed:

- Water (smooth surface, no wind)
    - The used radar waves are reflected on the water surface and it behaves like a mirror. The backscattered signal is minimal and the image appears black.
- Dry snow / white ice
    - The used radar waves are reflected in a diffuse way and the backscatter signal is much bigger compared to a smooth water surface. The image is much brighter.
- Wet snow
    - The used radar waves interact with wet snow like they do with water and reduce the backscattered signal heavily. The image appears dark.
- Ice
    - Uncovered blank ice does not interact much with the used radar waves. They are going through the ice and interact with the water surface below the ice layer.

3.4 Deep Learning

The following part is based on the book Deep Learning with Python Chollet (2018).

Deep learning is as subclass of machine learning methods and based on artificial neural networks with the goal to predict/extract specific data features from raw data in an iterative process. The basic principle of deep learning is shown in Fig. 11 represented by a successive layer representation. There is an input X which is pushed through a consecutive number of different layers, generating on each layer weighting factors and probabilities and ending in each cycle with a prediction Y'. The computer compares this prediction Y' with the true target value Y and calculates the deviation in a so-called loss function. These loss scores enter into an optimizer which adapts the set of weight factors in order to improve the next optimization step of the prediction. The goal of the network is to minimize the differences between the input true targets Y and their prediction Y'.

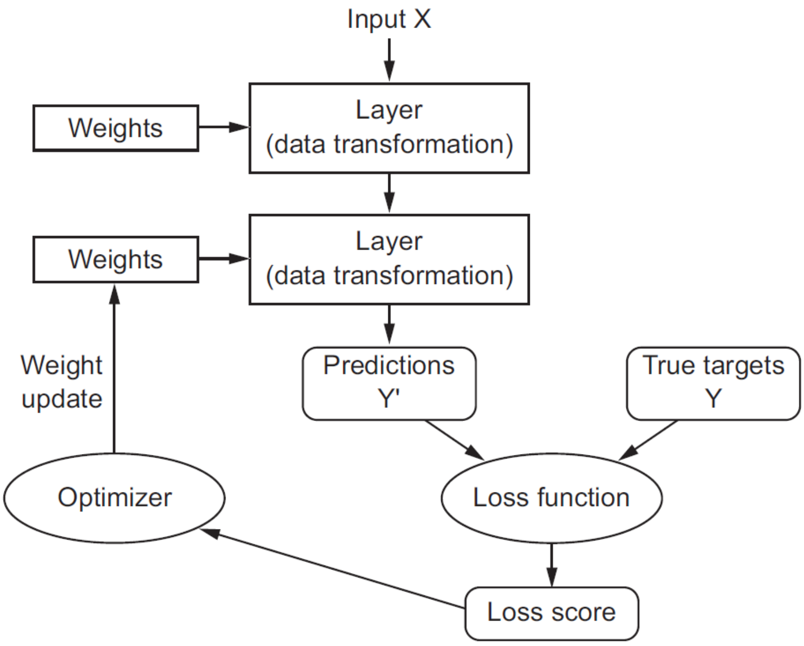


Figure 11. Model of a deep learning network (source: (Chollet, 2018), p. 11).

3.4.1 Convolutional Neural Networks (CNN)

This chapter is based on Zulkifli (2018).

A convolutional neural network is a sub class of deep neural networks and is most often used to analyze images. In Fig. 12 a normal network architecture of a regular convolutional network is shown schematically. It starts with the input image on the left side of the figure. This can be for example an RGB image (a bitmap image holding RGB color values in 3 image channels) or a gray scale image with a one-color information layer. If we start with a gray scale radar image from the area of Sils and pick a 32 x 32-pixel image and apply 32 filters in this first convolutional layer we are getting a feature map of that set of filters with the size of 32 x 32 x 32-pixels. Each filter is convolved across the width and height of the input image generating a 3-dimensional feature map represented by the first box in Fig. 12. Convolutional networks may then include pooling layers as well. A pooling layer reduces the image size by combining the outputs of neuron clusters at one layer into a single neuron in the next layer, increasing the depth of the network visualized by the second box in the figure below. This process is iteratively repeated until a fully connected layer is generated. In the fully connected layers all neurons are connected to each other and the process starts now to allocate the neurons with the highest values to the predefined labels, in our example ice and water.

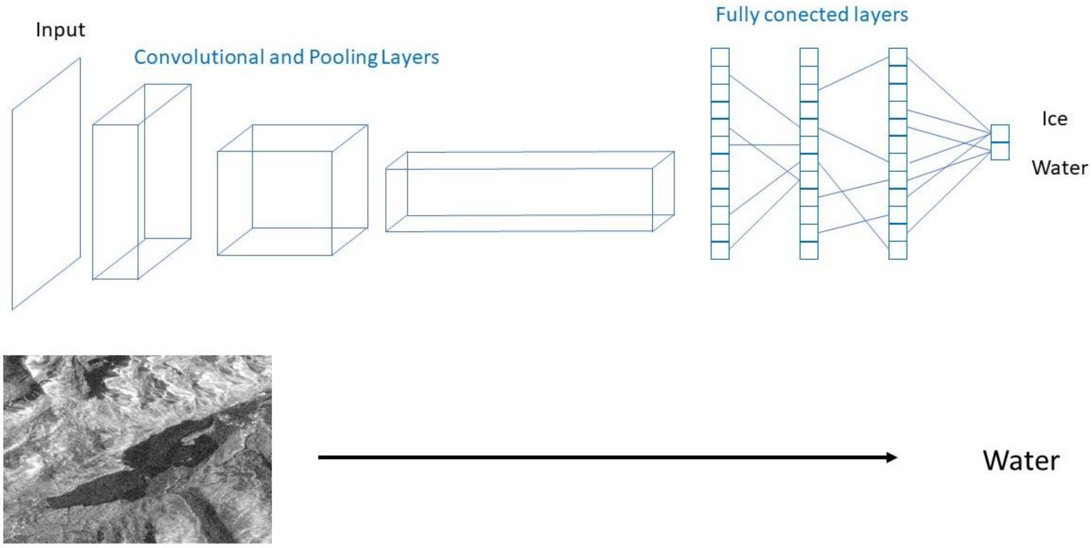


Figure 12. Normal convolutional network.

In networks which work like the network described above the result is a vector with probabilities for each label. The label with the biggest probability will determine the output, in our example either water or ice. However, there is no information anymore where this label is located in the input image and the generated information does only allow an image classification. After such a network analysis the computer can say what is on the image, but it has no idea where the things are.

If we not only want to know what is on the image but also where it is, we have to use semantic segmentation of neural networks. Such networks allocate to each pixel in the image an annotation to which label that pixel belongs to. With that change the computer knows what is and where it is on the image.

A segmentation network has normally three main parts (Fig. 13):

- Convolutional Layers
- Pooling Layers
- Transpose Convolutional Layers

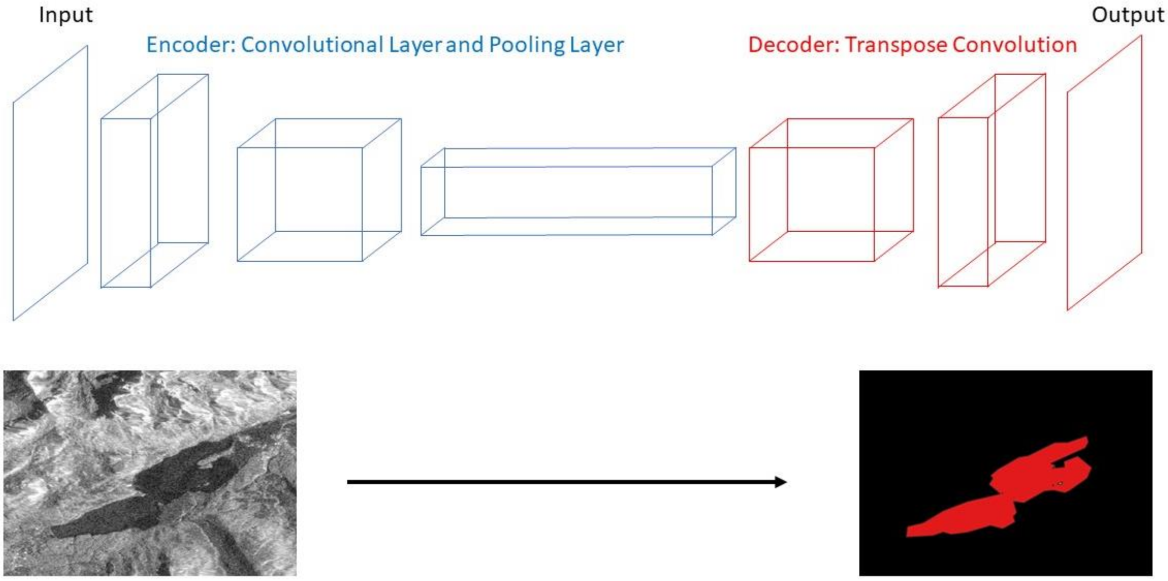

Input
Encoder: Convolutional Layer and Pooling Layer
Decoder: Transpose Convolution
Output

Figure 13. Flow scheme of a semantic segmentation model (top) and result (bottom) of the Lake Silvaplana indicating water in the lake area.

3.4.2 Architecture of Deeplab v3+

This chapter is based on Encoder-Decoder with Atrous Separable Convolution for Semantic Segmentation Chen et al. (2018) and Liang-Chieh et al. (2016/2019).

Deeplab v3+ is a state-of-the-art code for segmentation tasks with deep learning. In this thesis the newest version of Deeplab, Deeplab v3+, was used. Deeplab v3+ combines two techniques. On one side there is spatial pyramid pooling module, on the other side there is an encoder-decoder structure. Spatial pyramid pooling is able to work with multi scale contextual information and with images of different size. The encoder-decoder structure enables to catch the object boundaries. A modified Xception network (Xception 65) is used as backbone of Deeplab v3+ replacing all max pooling layers by depthwise separable convolutions with striding. Fig. 14 shows the architectural structure of the used network Xception 65:

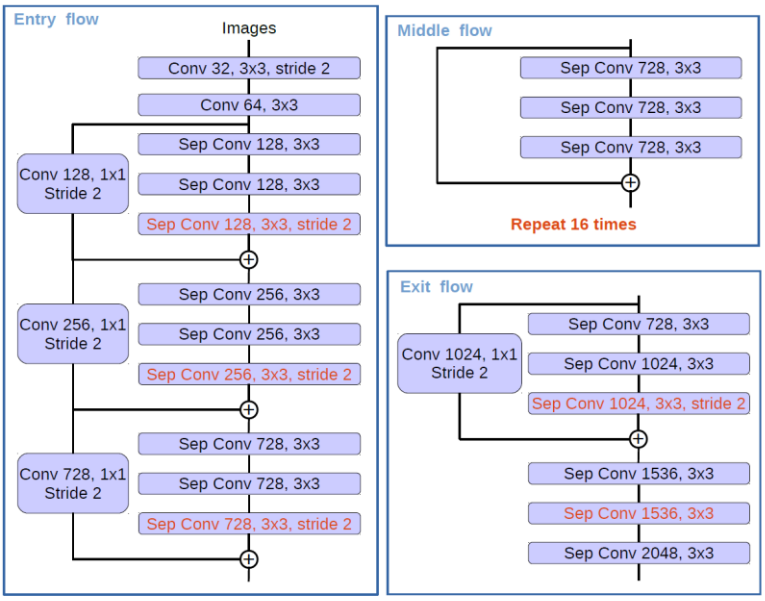


Figure 14. Architecture of Xception (source: (Chen et al., 2018), p. 7).

The following sub chapters will explain some of the relevant techniques, tools, and principles of Deeplab v3+ in more detail.

3.4.2.1 Encoder-decoder structure

This chapter is based on Chen et al. (2018).

The encoder-decoder structure enables to catch object boundaries in an image better and is based on a down-sampling / up-sampling process in a segmentation network as visualized in Fig. 15. It starts with the input image on the lower left side of the figure. In a consecutive down-sampling process (encoding), the size of the image is reduced with convolutional and pooling layers followed by an up-sampling process (decoding) based on stride transpose convolutions to get back an image with full input resolution.

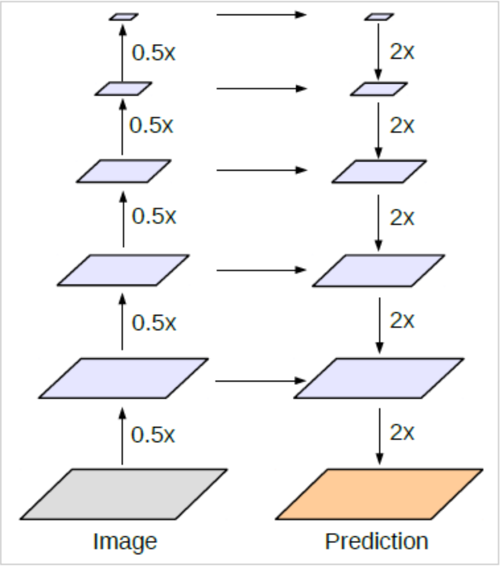


Figure 15 Encoder-Decoder structure (source : (Chen et al., 2018), p. 2).

3.4.2.2 Spatial Pyramid Pooling

This chapter is based on He, Zhang, Ren, & Sun (2015).

The problem in normal convolutional neuron networks is that they need images with fixed input size for the classifier and the fully connected layers. With spatial pyramid pooling it is possible for the network to deal with different input image sizes. The spatial pyramid pooling layer is added in most cases after the last convolutional layer. This layer has different bins with different spatial resolution (see Fig. 16, step 1), however their size is proportional to the input image size. In this way, the number of bins is fixed. Inside of these bins the information is pooled to a fixed size (Fig. 16, step 2). This fixed array can be used afterwards as input into fully-connected layers or a classifier (Fig. 16, step 3).

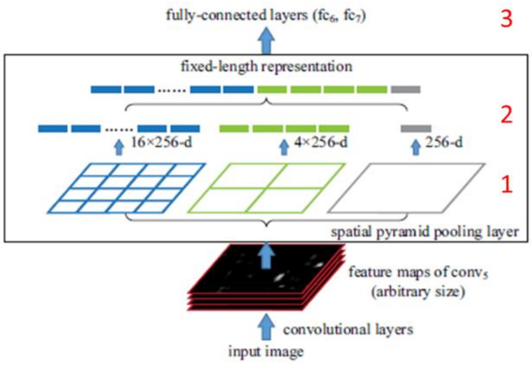
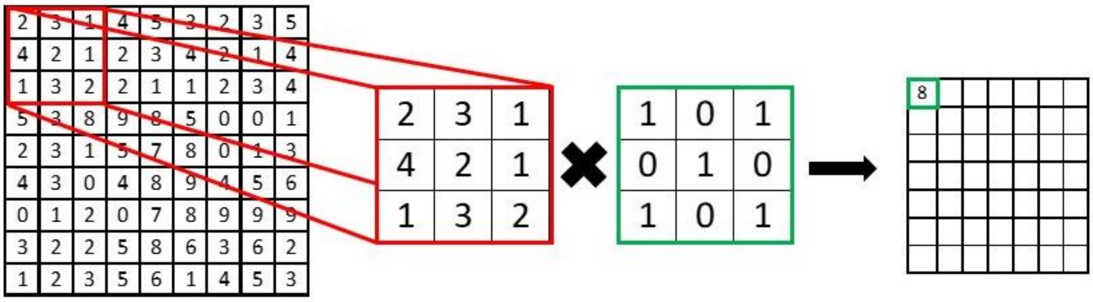

fully-connected layers (fc$_6$, fc$_7$)
3
fixed-length representation
2
16×256-d
4×256-d
256-d
1
spatial pyramid pooling layer
feature maps of conv$_5$
(arbitrary size)
convolutional layers
input image

Figure 16. Spatial Pyramid Pooling (source: (He et al., 2015), p. 3).

3.4.2.3 Convolution Layer

A convolution layer works as shown in Fig. 17. With these layers it is possible to detect structures in the image. For example, it is possible to detect edges which such a structure. On the left side is the input image. The numbers in the red framed grid represent the gray values. In a first step, a filter (marked green) will be overlaid over this input image. In the example shown below it is a 3 x 3 pixel filter with a binary 0/1 filtering. The filter value is multiplied with the corresponding pixel value and a mean value for the 3x3 matrix is evaluated.



2	3	1	4	5	3	2	3	5
4	2	1	2	3	4	2	1	4
1	3	2	2	1	1	2	3	4
5	3	8	9	8	5	0	0	1
2	3	1	5	7	8	0	1	3
4	3	0	4	8	9	4	5	6
0	1	2	0	7	8	9	9	9
3	2	2	5	8	6	3	6	2
1	2	3	5	6	1	4	5	3

2	3	1
4	2	1
1	3	2

1	0	1
0	1	0
1	0	1

Figure 17. Convolutional layer.

The filter is moving over the input image as shown in Fig. 18. In this example the kernel moves from one pixel to the next and always calculates the new value for the new pixel. The step size is equal to one. If it is done accordingly for the whole image the output layer will be smaller than the input layer along the following equation:

Size of the input layer minus the size of the filter plus one gives the size of the output image (in Fig. 17 it would be 9-3+1 = 7)

Because there is the same number of pixels in the x direction as in the y direction and the kernel is also rectangular the result applies for both axis. If that is not the case the calculation must be done for each axis separately.

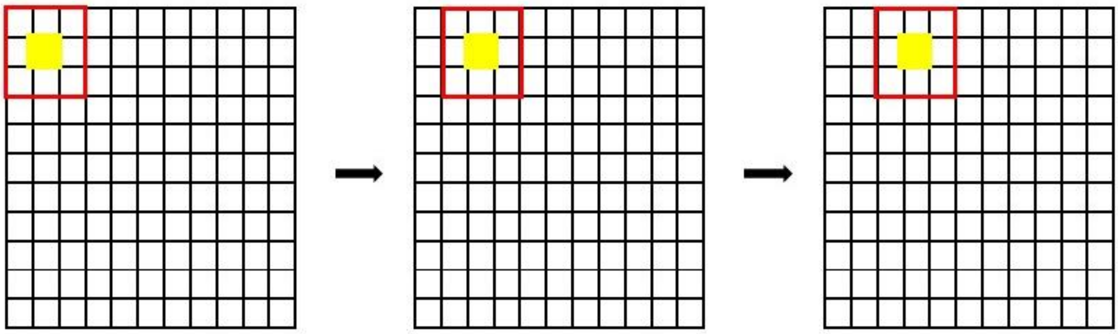


Figure 18. Sliding window.

3.4.2.4 Atrous Convolution

This chapter is based on (Chen et al., 2018).

An Atrous convolution filter is nearly the same as a convolutional filter. The calculations are the same. The only difference is the introduction of a specific factor which increases the field of view of an Atrous filter. Fig. 19 shows the differences if you apply filter factors 1, 2, or 3. For filter size 1, the Atrous filter is a normal convolutional filter. Applying a factor 2 or 3 increases the field of view as visualized in Fig. 19. The main advantage of such a filter is the possibility to incorporate the number of pixels of a greater area without the disadvantage of a big filter size.

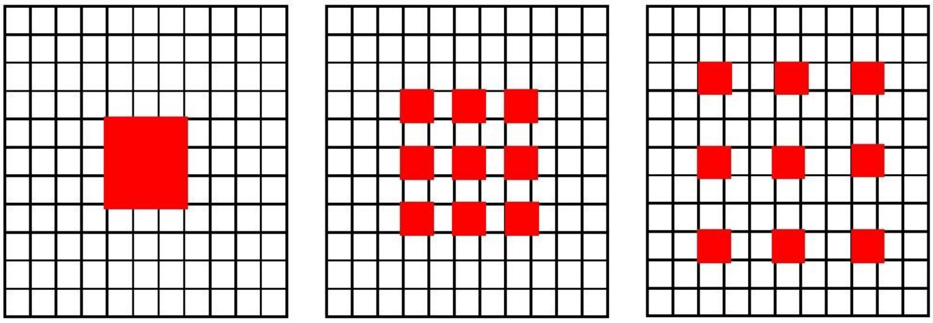


Figure 19. Schematic principal of an Atrous convolution with different filter sizes (left =1, middle =2, right = 3).

3.4.2.5 Pooling Layers

The pooling layers are mostly used to reduce the image size. In the example below (Fig. 20) there is a so called max-pooling layer described. Like a convolutional layer the pooling layer has a kernel size. In the example below the kernel size is 2 x 2 pixels with a strip size of two as well. The kernel determines the biggest value, which is inside of the kernel, and writes it into the resulting feature map.

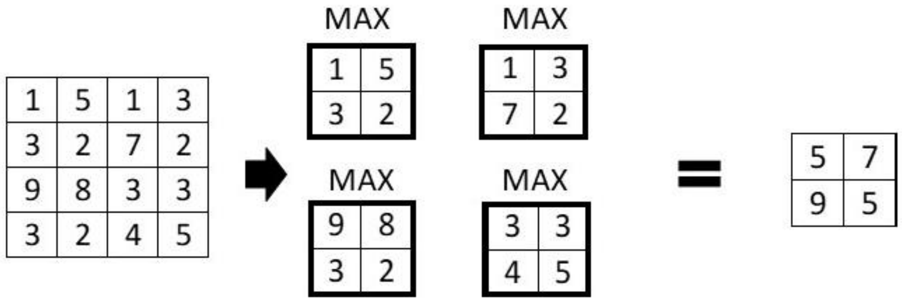


Figure 20 Pooling layer.

3.4.2.6 Depthwise separable convolutions

This chapter is based on C.-F. Wang (2018).

Because of the huge amount of multiplications in a convolutional filter step the needed calculation time can be very fast extremely high. A way to circumvent this problem is to apply depthwise separable convolutions which is explained with an example.

We start with an RGB image of the size 12 x 12 pixels and want to convolute with 256 filters of the size 5 x 5 pixels. There is no padding and the strip size is equals one. The output would have a size of 8x8x256 pixels (height, width, number of layers)

Let's have first a look at a normal convolution process. The process would result in 256 kernels of size 5x5x3 pixels. For one pixel in the output the kernel has to do 25 calculations per layer. Because there are 3 layers, this number must be multiplied with 3 resulting in 75 calculations for one output pixel. The output contains 8*8=64 pixels which gives a total of 8*8*75 = 4800 multiplications for one kernel totaling for 256 filter in 256*4800 = 1'228'800 calculations for this simple example.

If we use depthwise separable convolutions the number of calculations will be much smaller. In a first step, there will be three 5x5x1 kernels. Each of these filters will move over one layer in the input image. That makes a total of 3*5*5*8*8 = 4'800 multiplications. In a next step the pointwise convolution will be 256 times 1x1x3 kernels. Each of these kernels is moving 8x8 time, in total 256*1*1*3*8*8 = 49'152 calculations steps. Adding the two numbers results in a total of only 53'952 calculation steps if depthwise separable convolution is applied.

3.4.2.7 Transposed convolution

This chapter is based on Chen et al. (2018).

Deeplab v3+ up-samples bilinearly. Fig. 21 shows an up-sampling by the factor 2. In this case the raster size is doubled and the values from the original image are written into the raster as shown in Fig. 21.

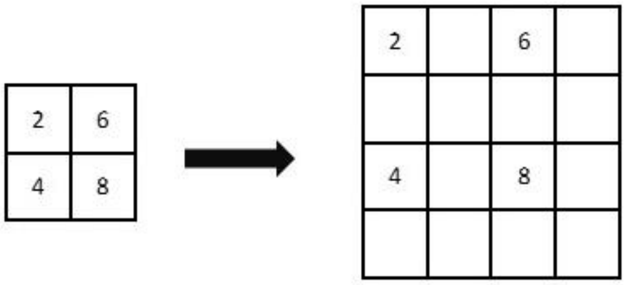


Figure 21 Up-sampling by factor 2.

The empty raster cells are filled up with a bilinear interpolation. For that the four nearest initial neighbor pixels are taken. In Fig. 22 this process is graphically shown illustrating the process for the black circle pixel (x,y). The pixel which has the shortest distance to the black point (yellow point) gets the highest weighting factor, the pixel which has the longest distance (red point) has the lowest weighting factor.

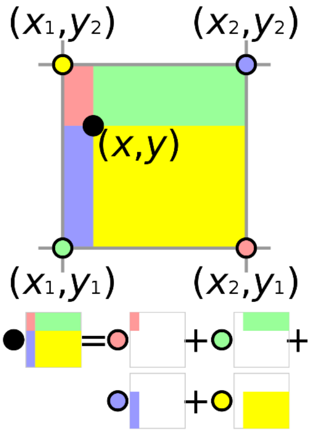


Figure 22. Schematic process description of a bilinear interpolation (source: (“Bilinear interpolation - Wikipedia,” 2019)).

3.4.2.8 Learning rate

This chapter is based on Zulkifli (2018).

The goal of a neural network is to minimize the difference between a prediction of a value and the true value itself via an iterative process along a pathway to a global minimum (Fig. 23). To achieve this, the network adapts the single step weights in the layers accordingly. Each step is symbolized in the figure with a graphic symbol, representing different learning rates. The squares are representing a high learning rate with the advantage of the possibility to converge fast in the minimum, however, with the disadvantage of having the possibility to overshoot. The network changes the parameters too fast and the global minimum cannot be found. On the other hand, symbolized with circles, there is a visualization of a process with a small learning rate with the disadvantage, that it can take very long to get to the global minimum and that there is the fair chance that the system finds only a local minimum. An optimization process has to be applied to find the appropriate learning rate for each system.

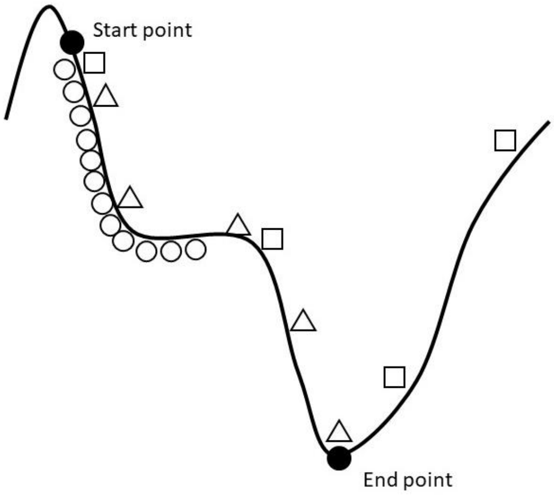


Figure 23. Schematic view of different learning rates in a neural network: squares: high learning rate, circles: low learning rate, triangle: optimal learning rate.

4 Methodology

In the following chapter the workflow of the thesis is described.

4.1 Data access

The goal of this bachelor thesis is to train a deep neural network to detect lake ice on satellite radar images. The necessary images are freely available SAR data of the Sentinel-1 satellite constellation of the Copernicus program of the European Space Agency (ESA). There are two possible ways to access the relevant data. The first one is to download the data directly from the ESA open access hub (ESA Open Access Hub, 2019). If this way is chosen, the whole pre-processing has still to be done. That includes for example the radiometric calibration and the terrain correction. The second possibility is to download the data from Google Earth Engine (Earth Engine Code Editor, 2019) where the pre-processing is already done. This saves time and gives the possibility to store all images in the Google Cloud and avoids the time-consuming work of pre-processing the data. The following pre-processing steps are already done by Google Earth Engine (Google Earth Engine, 2019):

- Apply orbit file
    - Corrects metadata of the orbit file
- GRD border noise removal
    - Corrects invalid data at the border
- Thermal noise removal
    - Corrects the thermal noise
- Radiometric calibration
    - Computes the backscatter intensity with the metadata
- Terrain correction
    - Corrects side looking effects by using the SRTM 30-meter digital elevation model

For the thesis, the data were downloaded as GeoTiff files from Google Earth Engine with the following data specifications:

- Polarization: VV and VH
- Ascending and descending orbits
- Level 1 Product
    - GRD
    - Geometric resolution at the equator is 10 x 10 meters per pixel, in the region of Sils the resolution is ca. 7 x 10 meters per pixel

The download is possible with a few lines of JavaScript in the code editor of the Google Earth Engine webpage. The only requirement is to have a Google account. One big disadvantage of this procedure is that every image must be manually downloaded to Google Drive (Google Drive, 2019). There is no possibility for a direct download to the own computer. As explained in chapter 3.1.4, there are different observation modes used by Sentinel-1 over different areas of the globe. In the area of interest (Lakes of St. Moritz, Silvaplana and Sils) there are mainly radar images in the Interferometric Wide Swath Mode (IW) available and they have been chosen as base data for this thesis. There are 238 images (from 119 days) available in the period of 1st September 2016 to 31st May 2017. 119 of them are with VV polarization. The other 119 images are with VH polarization. The images in the different polarization modes are captured at the same time. In Fig. 24 the two images taken on 1st September 2016 are shown for comparison. Based on the larger contrast in the VV images, most of the experiments of this thesis were run with the VV images.

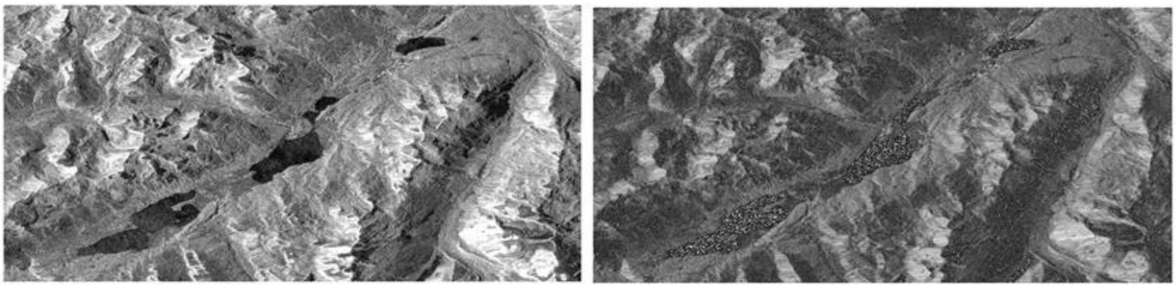


Figure 24. SAR Sentinel-1 images of the region of Sils (1.9.2016), left with VV, right with VH polarization.

4.2 Data pre-processing

Google Earth Engine is pre-processing the image data as described in chapter 3.2. However, the input files have to be processed further before they can be used in a neural network. Two additional processing steps have to be done with all data files:

- Data reduction: detection and identification of all lake areas in all images to reduce the total data size
- Log scaling: conversion of the negative to positive values which can be transferred into gray scale values

In a first step, the areas around the Lake Silvaplana, Lake Sils, and Lake St. Moritz have been cut out from the original SAR Sentinel-1 image from Google Earth Engine with the help of a self-written Python code (using freely available Python libraries) (Fig. 25).

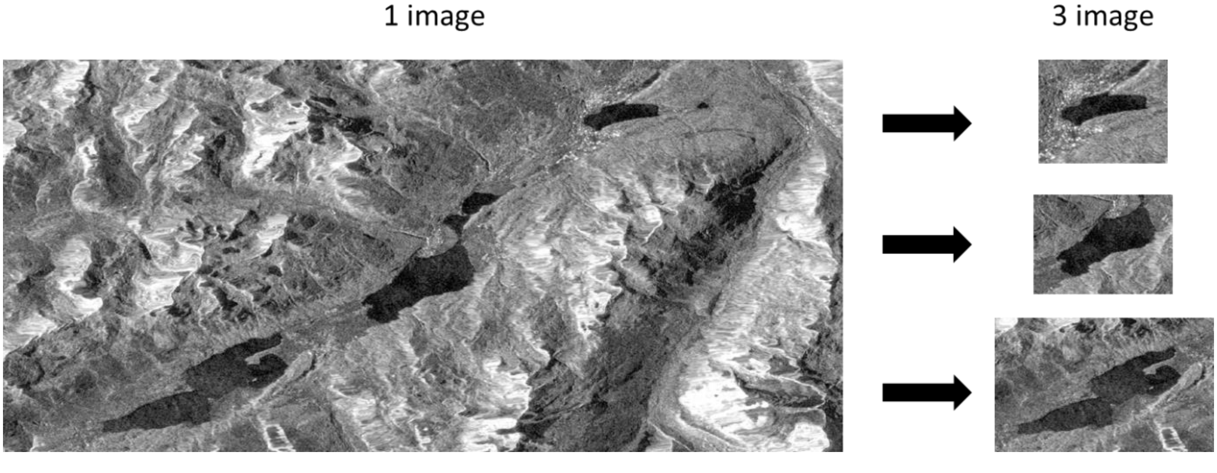


Figure 25. SAR Sentinel-1 image of the region of Sils. Data reduction step with the help of shapefiles of different lake areas.

Then, the code uses the respective shapefiles of the Lake St. Moritz, Lake Silvaplana, and Lake Sils to check if a single pixel lies within the lake boundaries or not. The used shapefiles are the same as the ones used in the publication ‘Lake ice detection in low-resolution optical satellite images’ (Tom et al., 2018).

The second processing step is needed because Google Earth Engine uses the following log scaling Eq. (7 *)* on all backscattering values:

$$10 * \log(x) \qquad (7)$$

According to Google, this log-scaling is done because the backscatter coefficients can vary by several orders of magnitude (Google Earth Engine, 2019). However, with this log-scaling factor, all values are negative (Fig. 26) and cannot be used in the neural network. The reason for this is that the input for Deeplab v3+ is a portable network graphic file (PNG). This format has values between 0 and 255 which correspond to 256 different gray shades between black and white. Negative input values are set per default to zero (Rander-Pehrson et al., 1999) resulting in the complete loss of the whole image information and the image would appear completely black.

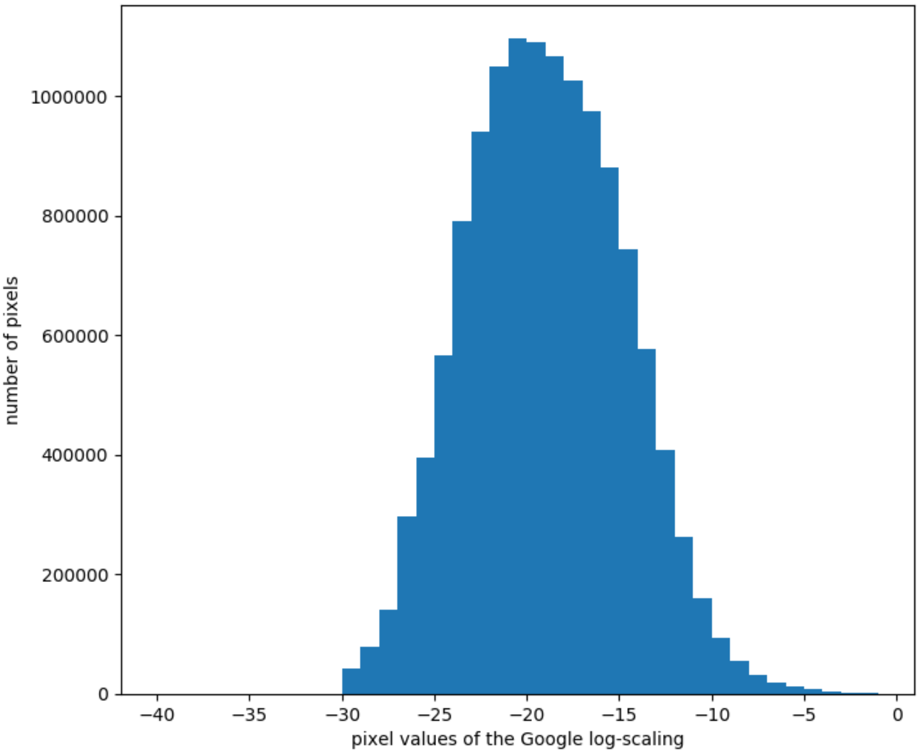


Figure 26. Distribution of SAR data from Google Earth Engine.

The log-scaling by Google is undone with the Eq. (8 *)*.

$$10^{\frac{x}{10}} \qquad (8)$$

Inspired by the recent deep learning study about deforestation using Sentinel-1 SAR data (Gümgümcü, 2018) the additional log scaling factor of Eq. (*8)* was applied on the backscatter values:

$$x^{\frac{8}{27}} \qquad (\,9\,)$$

With this log scaling Eq. (*9*) the distribution becomes more Gaussian compared to the unprocessed data and the values are getting positive between 0 and 1 (Fig. 27 left side). In a next step, the values are multiplied by 255 to adapt them to the .png format. The resulting distribution is shown in Fig. 27 on the right side.

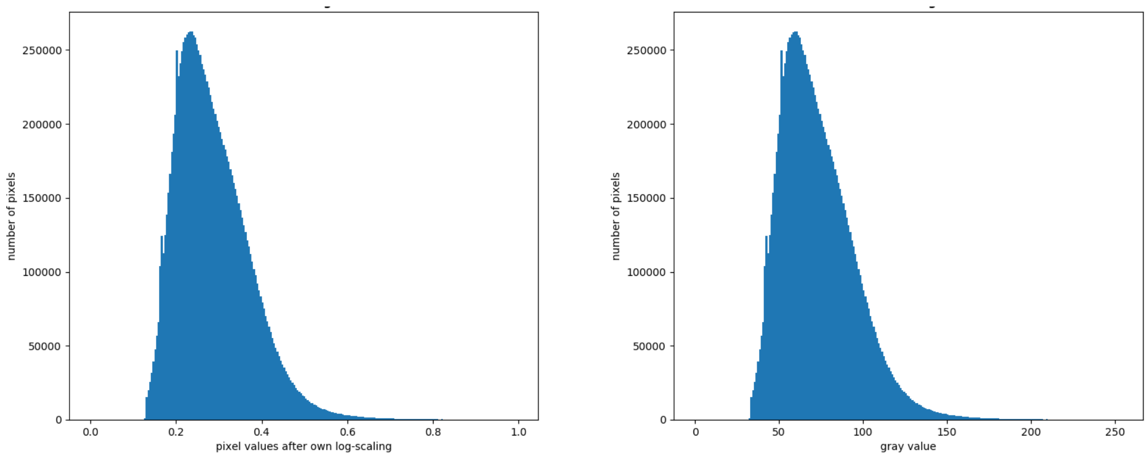


Figure 27. left: distribution after applying Eq. (*9*), right: distribution after multiplying the left distribution with 255.

At the end of this process, the Python code writes all pixels which are belonging to the lake areas with the image coordinates and the pixel value (backscatter amplitude) into a text file. All the pixels which are outside of the lakes are labeled by setting the pixel value to 255 (white). This process generated three sets of 238 input data text files (one set for each lake including 119 images in VV polarization and 119 images in VH polarization) with the log scaled backscatter information on a scale of 0-255.

If this text file information is back converted into an image, typical images as the one shown in Fig. 28 are generated.

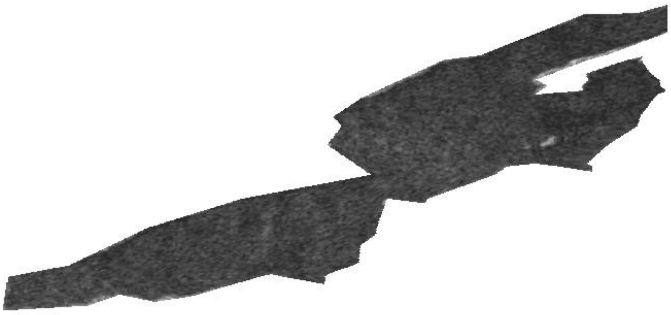


Figure 28. Result of the pre-processing steps for Lake Sils (with log-scaling).

4.3 Ground truth

The ground truth data set for this thesis from the region of Sils was established by the Photogrammetry and Remote Sensing Institute of the ETH Zurich. Webcam images were used to judge if the surface of the lake is frozen or not. Fig. 29 shows some examples of one webcam view (St. Moritz) at different dates.

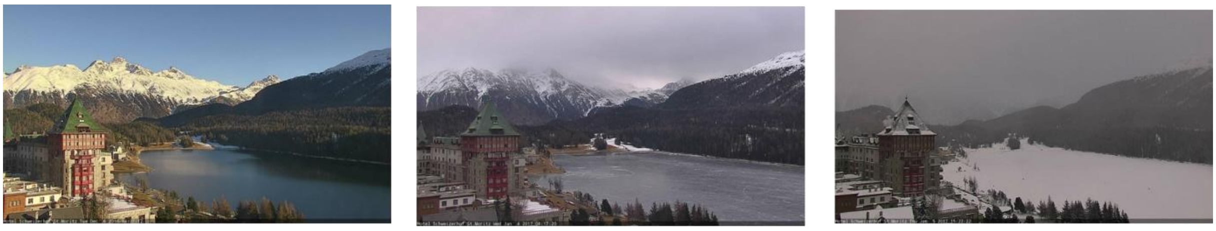


Figure 29. Different webcam images with different lake surface coverage; left: 6.12.2016, water, middle: 4.1.2017, ice, right: 5.1.2017, snow (Muyan, 2018)

The (daily) ground truth data of all three lakes are labeled as follows:

- s/i: snow or ice (90%-100%)
- ms/mi: more snow or more ice (at least more than 60%-90%)
- mw: more water (more than 60%-90%)
- w: water (90%-100%)

This information was used to generate ground truth for this thesis. Fig. 30 shows two examples of generated ground-truth images of Lake Sils. On the left side from September 2016 representing a water surface visualized in green. On the right side an image from February 2017 representing the completely frozen sea visualized in red. The background is set white. There is no precise ground truth available for the transition dates, when the lake is partly frozen.

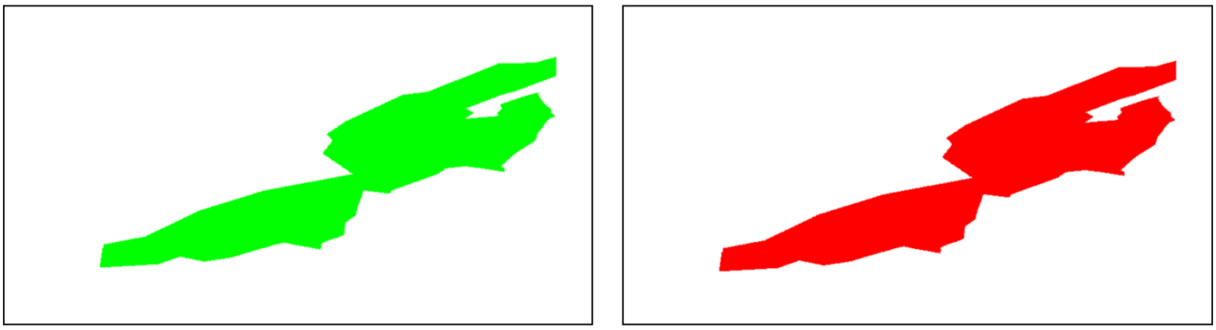


Figure 30. left: label example for water, right: label example for snow/ice (Lake Sils).

In Tab. 3, an overview about the number of water, snow/ice and transition days is given (period 1.9.2016 - 31.5.2017). All lakes are in the same geographic region and therefore the 'completely frozen days' do not vary too much.

Table 3. Water, snow/ice and transition days of Lake St. Moritz, Lake Silvaplana and Lake Sils.

Name	Water	Snow/ice	Transition
Lake St. Moritz	148 days	106 days	19 days
Lake Silvaplana	123 days	102 days	48 days
Lake Sils	136 days	100 days	37 days

4.4 Training and Validation Data

The text files described in chapter 4.2 are used to generate the training and validation datasets. A split of 80% training to 20% validation is applied. The allocation to the training set / validation set is done per month and visualized in an indicative way in Fig. 31.

If we assume that there are 10 images of the region Sils per month and we have 3 lakes, then there are 30 text files. These text files are sorted per date in a list. That means that there are 24 text files for the training dataset and 6 text files for the validation dataset. Every sixth text file is therefore a validation data set (green dots). This general approach ensures that there are files in the training dataset and in the validation dataset from the beginning (1.9.2016) to the end (31.5.2017) of the investigated time period.

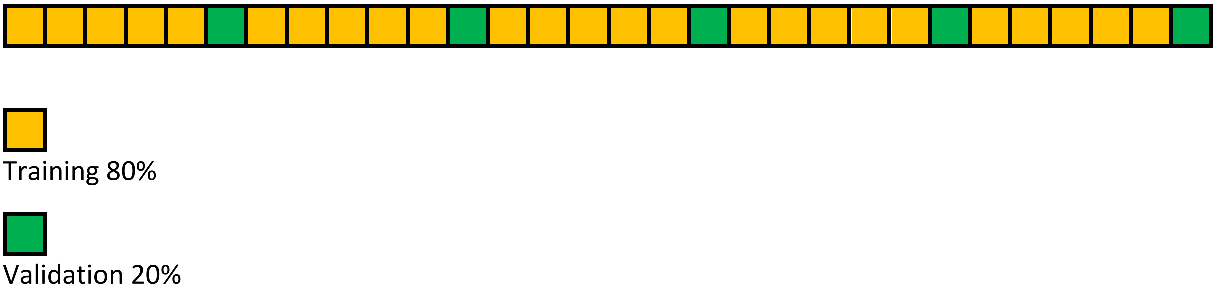


Figure 31. Visualization of the training/validation split.

Square patches are extracted from the image data and fed as input to the neural network. Patches with 50% overlap were used in training. However, validation was done with non-overlapping patches.

4.5 Setting up Deeplab v3+

The following Deeplab v3+ parameters have been set:

- Atrous convolution = 1,2,3
  - With this setting of Atrous convolution it is also possible to run the experiments with the smallest patch size of 16 x 16 pixels
- Outputstride = 2
  - The input images are already small
- Batch size = max 4
  - Limited by the available memory on the local computer

More information about the settings applied in Deeplab v3+ can be found in the Appendix.

4.6 Evaluation Metrics

A confusion matrix is used to evaluate the output of the network (Fig. 32). The ground truth and the prediction are differentiated along the three labels water, snow/ice, and background. A similar confusion matrix with the same indicators has been used in the master thesis 'Lake ice detection with webcams' (Muyan, 2018).

		predcition		
		water	**snow/ice**	**background**
ground truth	**water**	true water (TW)	false snow/ice (FS1)	false background (FB1)
	snow/ice	false water (FW1)	true snow/ice (TS)	false background (FB2)
	background	flase water (FW2)	false snow/ice (FS2)	true background (TB)

Figure 32. Confusion matrix.

With these basic output values of an experiment, the recall and the precision can be calculated for each label. The following Eq. (*10*) and Eq. (*11*) are an examples for the recall and the precision of the label 'water'. For the other two labels (snow/ice and background the calculation have to be done accordingly). The recall describes the ratio of correctly assigned pixels to the sum of ground truth pixels with the corresponding label. The precision describes the ratio of the correctly assigned pixels to the sum of the predicted pixels with a corresponding label. It shows the quality of the prediction of a specific label.

$$recall = \frac{TW}{TW + FS1 + FB1} \qquad (10)$$

$$precision = \frac{TW}{TW + FW1 + FW2} \qquad (11)$$

A combination of the recall and precision is the Intersection over Union (short IoU) for a specific label.

$$IoU = \frac{TW}{TW + FW1 + FW2 + FS1 + FB1} \qquad (12)$$

To get an overall measure of the quality of the prediction of the whole dataset the 'mean IoU' and the 'overall accuracy' are calculated as follows:

$$mean\ IoU = \frac{\sum_{i=1}^{N} IoU_i}{N} \qquad (13)$$

N = number of classes

$$overall\ accuracy = \frac{TW + TS + TB}{number\ of\ all\ pixels} \qquad (14)$$

5 Results

In this chapters the results of the different experiments are shown. Mean IoU and overall accuracy are reported in two different ways: including the background and without background. A pretrained Xception network (with cityscape dataset) was used for all experiments. (Based on the experience with an experiment without pretraining where the results were not satisfying still after 40 epochs)

The following list summarizes the varied parameters:

- Different orbit constellations
    - All orbits together
    - Orbits 15, 66 and 168 together
    - Only orbit 168
    - Only orbit 15
- Different combinations of polarization modes
- Different patch sizes
    - 64x64 pixels, 32x32 pixels, 16x16 pixels
- Different numbers of epochs
    - 10, 20, 30, 40
- Different learning rates
    - 0.01, 0.001, 0.0001, 0.00001

5.1 Orbit Experiments

5.1.1 All orbits together / variation of number of epochs

- Polarization: VV
- Learning rate: 0.0001
- Number of epochs: variable
- Patch size: 64 x 64 pixel
- Orbit: all together

Table 4. Result of experiment: all orbits together / variation of epochs.

number of epochs		10	20	30	40
Recall/ Precision/ IoU	water	99.5% / 65.5% / 65.3%	99.7% / 66.1% / 65.9%	97.9% / 67.9% / 66.9%	99.5% / 68.1% / 67.9%
	snow/ice	3.3% / 99.7% / 3.3%	5.4% / 99.8% / 5.4	11.4% / 99.7% / 11.4%	13.0% / 99.9% / 13.0%
	background	99.6% / 99.6% / 99.2%	99.9% / 99.9% / 99.8%	99.9% / 99.6% / 99.6%	99.9% / 99.9% / 99.9%
mean IoU without background		**34.3%**	**35.6%**	**39.2%**	**40.5%**
overall accuracy without background		**66.2%**	**67.0%**	**69.2%**	**69.7%**
mean IoU with background		55.9%	57.0%	59.3%	60.3%
overall accuracy with background		84.1%	95.1%	95.3%	95.5%

Result: The complete data set with all images from all orbits can be obviously not trained successfully. The results are not good. The increase in the number of epochs does not show a significant increase in 'mean IoU without background' and 'overall accuracy without background'.

Conclusion: Based on these results it was decided to split in a first step the orbits and investigate data sets of orbits 66/168 and 15/117 separately. This split allows to check if there is an influence of flight direction or day time (morning vs evening).

5.1.2 Ascending and descending orbits together

- Polarization: VV
- Learning rate: 0.0001
- Number of epochs: 30
- Patch size: 64 x 64 pixel
- Orbit: descending: 66/168 (5 am) together. ascending: 15/117 (5 pm) together

Table 5. Result of experiment: orbit combinations 66,168 and 15,117.

orbit combinations		66, 168	15, 117
Recall/ Precision/ IoU	water	33.6% / 82.0% / 31.3%	98.4% / 68.0% / 67.2%
	snow/ice	89.4% / 45.0% / 42.4%	39.6% / 96.5% / 39.1
	background	99.8% / 99.9% / 99.8%	99.4% / 99.9% / 99.8%
mean IoU without background		**37.0%**	**53.1%**
overall accuracy without background		**54.8%**	**68.7%**
mean IoU with background		58.0%	73.3%
overall accuracy with background		93.0%	95.8%

Result: Compared with experiment 5.1.1, the results are improving but are still not up to the mark.

Conclusion: Splitting the dataset based on orbits improves the results, even though the dataset size will decrease which could eventually result in overfitting.

5.1.3 All orbits separated

- Polarization: VV
- Learning rate: 0.0001
- Number of epochs: 30
- Patch size: 64 x 64 pixel
- Orbit: all orbits separated

Table 6. Result of experiment: all orbits separated.

orbit		15	66	117	168
Recall/ Precision/ IoU	water	93.9% / 77.8% / 74.0%	93.2% / 91.4% / 85.7%	77.7% / 83.1% / 67.1%	75.1% / 83.4% / 65.3%
	snow/ice	68.4% / 92.4% / 64.8%	89.7% / 93.2% / 84.2%	82.7% / 78.9% / 67.8%	82.3% / 74.5% / 64.3%
	background	99.9% / 99.8% / 99.8%	99.5% / 98.8% /98.2%	99.9% / 99.8% / 99.8%	99.9% / 99.8 % / 99.8%
mean IoU without background		**69.4%**	**84.9%**	**67.4%**	**64.8%**
overall accuracy without background		**83.0%**	**92.7%**	**81.1%**	**79.1%**
mean IoU with background		79.5%	89.3%	78.2%	76.4%
overall accuracy with background		97.2%	95.6%	96.9%	96.7%

Result: The results are much better than the two experiments before. It seems that the different orbits have quite an influence and that Deeplab v3+ learns more from the data if the images of the different orbits are separated. Doing this, the system gets a set of data where the incidence angle is always the same within the dataset and this influences the results of the experiments positively. The best results are achieved with orbit 66 with a 'mean IoU without background' of 84.9% and an 'overall accuracy without background' of 92.7%.

Conclusion: There is an influence of the incidence angle of the radar beam and the data set has to be split along the different orbits for all the following experiments.

5.2 Polarization Experiments

5.2.1 VH

- Polarization: VH
- Learning rate: 0.0001
- Number of epochs: 30
- Patch size: 64 x 64 pixel
- Orbit: all orbits separated

Table 7. Result of experiment: all orbits separated, VH polarization.

orbit		15	66	117	168
Recall/ Precision/ IoU	water	99.0% / 69.4% / 68.9%	98.6% / 77.7% / 76.9%	83.4% / 81.1% / 69.9	73.9% / 81.3% / 63.1%
	snow/ice	49.7% / 99.7% / 49.7	68.8% / 98.6% / 68.1%	79.1% / 83.0% / 68.1%	82.9% / 77.1% / 66.5%
	background	99.9% / 99.8% / 99.8%	99.9% / 99.9% / 99.8%	99.9% / 99.8% / 99.9%	99.9% / 99.8% / 99.8%
mean IoU without background		**59.3%**	**72.5%**	**69.0%**	**64.8%**
overall accuracy without background		**76.7%**	**85.0%**	**82.2%**	**79.2%**
mean IoU with background		72.8%	81.6%	79.2%	76.5%
overall accuracy with background		96.2%	97.5%	97.1%	96.6%

Result: These results have to be compared with the results of experiment 5.1.3. Interestingly the results for orbits 15 and 66 are significantly worse, whereas orbits 117 and 168 are comparable with the results obtained with VV polarization. This might be another hint that the interaction of a microwave radar beam with different surface structures and under different incidence angles is complex and will need a further thorough investigation.

Conclusion: VV polarization gives higher 'mean IoU without background' and 'overall accuracy without background' scores than VH polarization.

VV and VH combined

To get a complete image about the influence of the polarization, there was also an experiment done combining all VV and VH images in one data set like an RGB image. The first layer (R) contained all VV images, the second layer (G) all VH images, and the third layer (B) was set zero.

- Polarization: VV and VH
- Learning rate: 0.001
- Number of epochs: 30
- Patch size: 64 x 64 pixel
- Orbit: 66

Table 8. Result of experiment: VV and VH combined.

number of epochs		30
Recall/ Precision/ IoU	water	99.1% / 80.0% / 79.5%
	snow/ice	72.7% / 99.0% / 72.2%
	background	99.9% / 99.9% / 99.9%
mean IoU without background		**75.8%**
overall accuracy without background		**86.8%**
mean IoU with background		83.9%
overall accuracy with background		97.9%

Result: The combination of VV and VH polarized images does not improve the training results.

Conclusion: All further experiments have been done therefore using the data set of images with VV polarization.

5.3 Patch Size Experiments

For further optimization, orbit 66 has been chosen.

- Polarization: VV
- Learning rate: 0.0001
- Number of epochs: 30
- Patch size: variable
- Orbit: 66

Table 9. Result of experiment: different patch size.

patch size		128 x 128 pixel	64 x 64 pixel	32 x 32 pixel	16 x 16 pixel
Recall/ Precision/ IoU	water	failed	93.2% / 91.4% / 85.7%	98.6% / 71.1% / 70.5%	99.7% / 57.9% / 57.8%
	snow/ice	failed	89.7% / 93.2% / 84.2%	55.7% / 98.6% /55.2%	20.9% / 99.4% / 20.9%
	background	failed	99.5% / 98.8% /98.2%	99.9% / 99.8% / 99.8%	99.9% / 99.9% / 99.8%
mean IoU without background		**failed**	**84.9%**	**62.9%**	**39.3%**
overall accuracy without background		**failed**	**92.7%**	**78.7%**	**62.2%**
mean IoU with background		failed	89.3%	75.2%	59.5%
overall accuracy with background		failed	95.6%	96.4%	93.6%

Result: Bigger patch sizes seem to give better results. There is a trend for better 'mean IoU without background' and 'overall accuracy without background' scores going from 16 x 16 to 64 x 64 pixel patches. The experiment with a patch size of 128 x 128 pixels failed because one tensor became too big. There was no time left to mitigate this by searching another parameter setting within Deeplab v3+.

Conclusion: It seems that the network is able to learn better if the information is not reduced too much to small patches and that it can profit from bigger structures within bigger areas. For the further optimization, 64 x 64 patches have been chosen.

5.4 Epochs Experiments

- Polarization: VV
- Learning rate: 0.0001
- Number of epochs: variable
- Patch size: 64 x 64 pixel
- Orbit: 66

Table 10. Result of experiment: different number of epochs.

number of epochs		10	20	30	40
Recall/ Precision/ IoU	water	97.1% / 77.9% / 76.1%	98.5% / 72.4% / 71.6%	93.2% / 91.4% / 85.7%	81.4% / 96.1% / 78.8%
	snow/ice	69.3% / 97.1% / 68.0%	58.0% / 99.6% / 57.8%	89.7% / 93.2% / 84.2%	96.1% / 82.7% / 80.0%
	background	99.3% / 98.6% / 97.9%	99.5% / 98.5% / 98.1%	99.5% / 98.8% /98.2%	99.3% / 99.1% / 98.4%
mean IoU without background		**72.1%**	**64.7%**	**84.9%**	**79.4%**
overall accuracy without background		**85.2%**	**80.4%**	**92.7%**	**89.3%**
mean IoU with background		80.1%	75.8%	89.3%	85.8%
overall accuracy with background		91.8%	89.7%	95.6%	94.1%

Results: The initial experiment described in 5.1.1 was now repeated with orbit 66. There is no clear trend in the results, 30 epochs giving the best results, however the differences between the different numbers of epochs being not too big.

Conclusion: Further optimization would be needed to find out the optimum number of epochs. For this thesis, 30 epochs have been chosen.

5.5 Learning Rate Experiments

5.5.1 Different learning rates over 30 epochs

- Polarization: VV
- Learning rate: variable
- Number of epochs: 30
- Patch size: 64 x 64 pixel
- Orbit: 66

Table 11. Result of experiment: different learning rates (30 epochs).

learning rate		0.01	0.001	0.0001	0.00001
Recall/ Precision/ IoU	water	failed	93.2% / 91.7% / 86.0%	93.2% / 91.4% / 85.7%	82.1% / 55.8% / 49.7%
	snow/ice	failed	90.3% / 92.9% / 84.5%	89.7% / 93.2% / 84.2%	26.2% / 58.8% / 22.1%
	background	failed	99.9% / 99.9% / 99.9%	99.5% / 98.8% /98.2%	99.9% / 99.5% / 99.4%
mean IoU without background		**failed**	**85.2%**	**84.9%**	**35.9%**
overall accuracy without background		**failed**	**92.4%**	**92.7%**	**56.9%**
mean IoU with background		failed	90.1%	89.3%	57.1%
overall accuracy with background		failed	98.7%	95.6%	93.0%

Results: For 30 epochs of training there is no big difference between learning rate 0.001 and 0.0001.

Conclusion: Run another experiment with 40 epochs to check influence of learning rate variation.

5.5.2 Different learning rates over 40 epochs

- Polarization: VV
- Learning rate: variable
- Number of epochs: 40
- Patch size: 64 x 64 pixel
- Orbit: 66

Table 12. Result of experiment: different learning rates (40 epochs).

learning rate		0.01	0.001	0.0001	0.00001
Recall/ Precision/ IoU	water	failed	96.7% / 88.6% / 86.1%	81.4% / 96.1% / 78.8%	87.6% / 56.8% / 52.5%
	snow/ice	failed	86.1% / 96.5% / 83.5%	96.1% / 82.7% / 80.0%	25.8% / 61.7% / 22.2%
	background	failed	99.9% / 99.6% / 99.5%	99.3% / 99.1% / 98.4%	98.3% / 98.9% / 97.3%
mean IoU without background		**failed**	**84.8%**	**79.4%**	**37.4%**
overall accuracy without background		**failed**	**92.1%**	**89.3%**	**58.9%**
mean IoU with background		failed	89.7%	85.8%	57.3%
overall accuracy with background		failed	95.9%	94.1%	78.9%

Results: Compared to 5.5.1 there is no improvement and the tendency to favor 0.001 as learning rate is shown in this experiment as well.

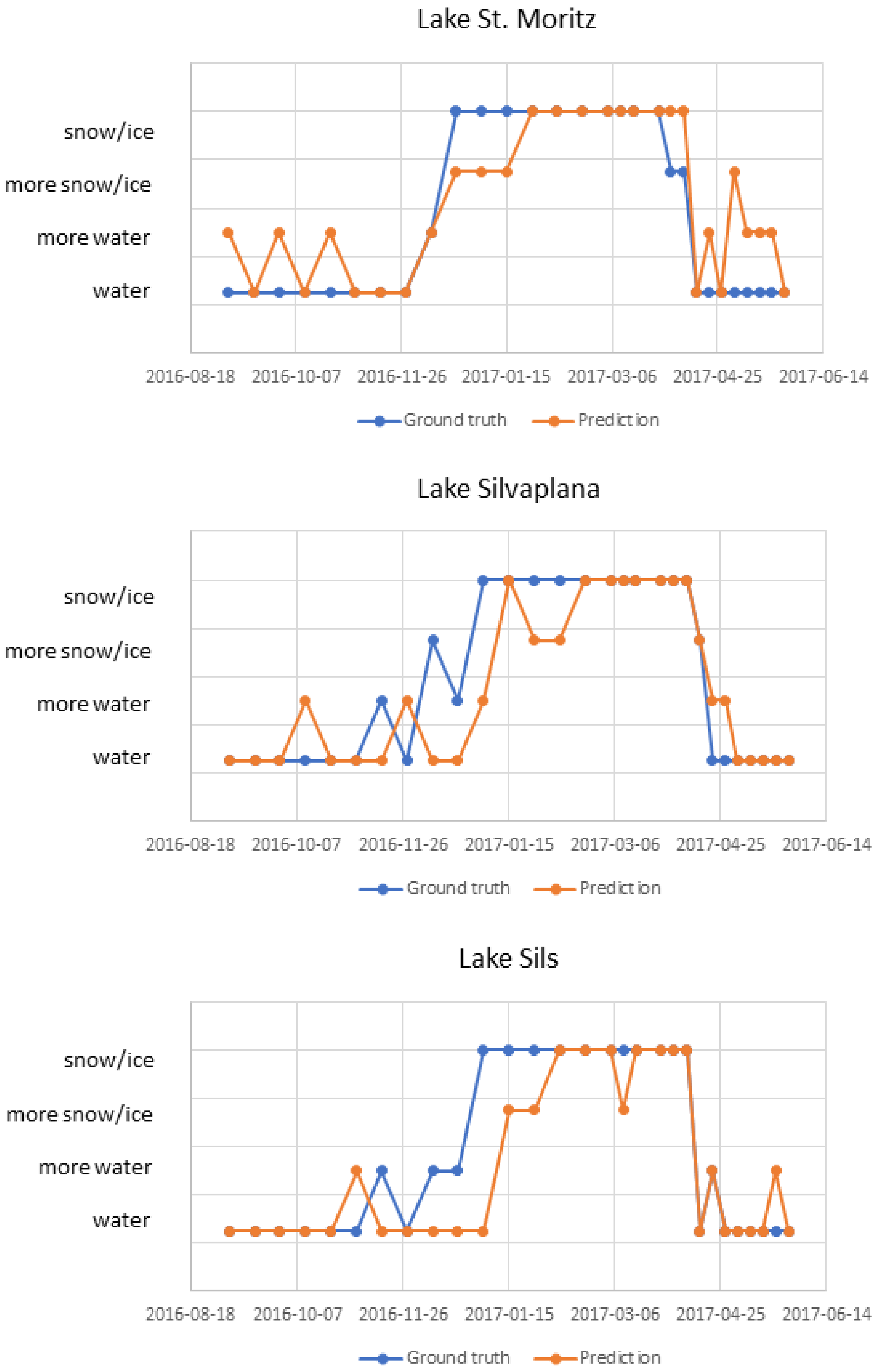
Conclusion: Continue to use 30 epochs with a learning rate of 0.001.

5.6 Time line experiments on orbit 15, 66, 117, 168

Based on the results of the preliminary optimization described in 5.1 - 5.5 a series of experiments was started with each lake/orbit combination resulting in a time line prediction. The following parameters have been applied:

- Polarization: VV
- Learning rate: 0.001
- Number of epochs: 30
- Patch size 64 x 64 pixel
- Orbit: all orbits separated

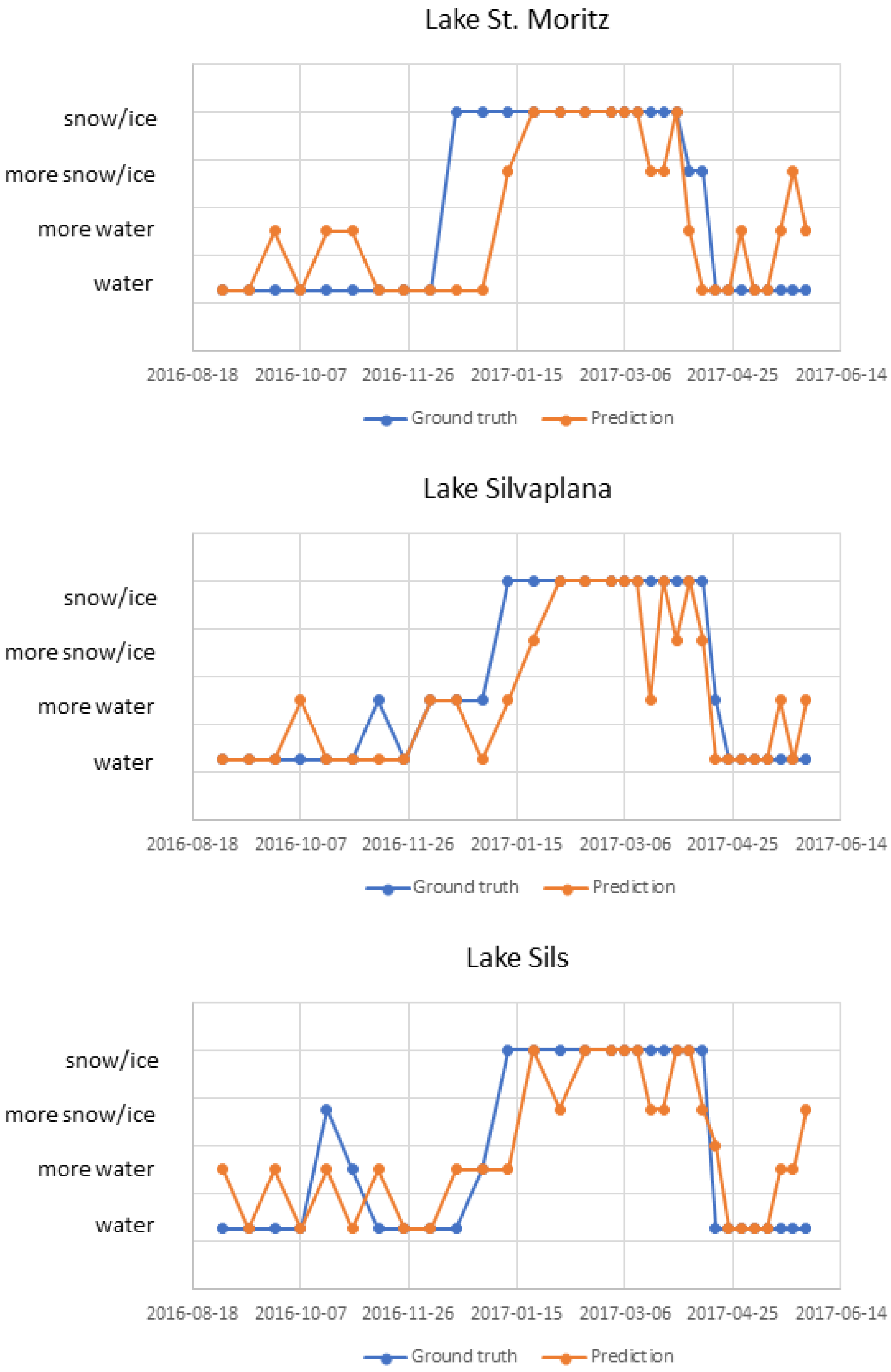
The evaluation of these experiments is different compared to the evaluation used in chapters 5.1-5.5. For this evaluation only the pixels which are allocated completely inside the lake area are considered. Each pixel can only be predicted either as snow/ice or as water. The number of all pixels assigned to the label water are summed up and compared to the total number of pixels within the lake boundaries. If this value is above 90%, the lake is labeled to be non-frozen (water, w). If this value is between 50-90%, the lake is assigned to the label 'more water' (mw). For the case that this value is between 10 and 50%, the label is set to 'more ice/snow' (mi) and for values below 10% to 'snow/ice' (s/i). Fig. 33 to Fig.36 show the results of the different orbits.


Lake St. Moritz

Lake Silvaplana

Lake Sils

Figure 33. Timeline orbit 66.

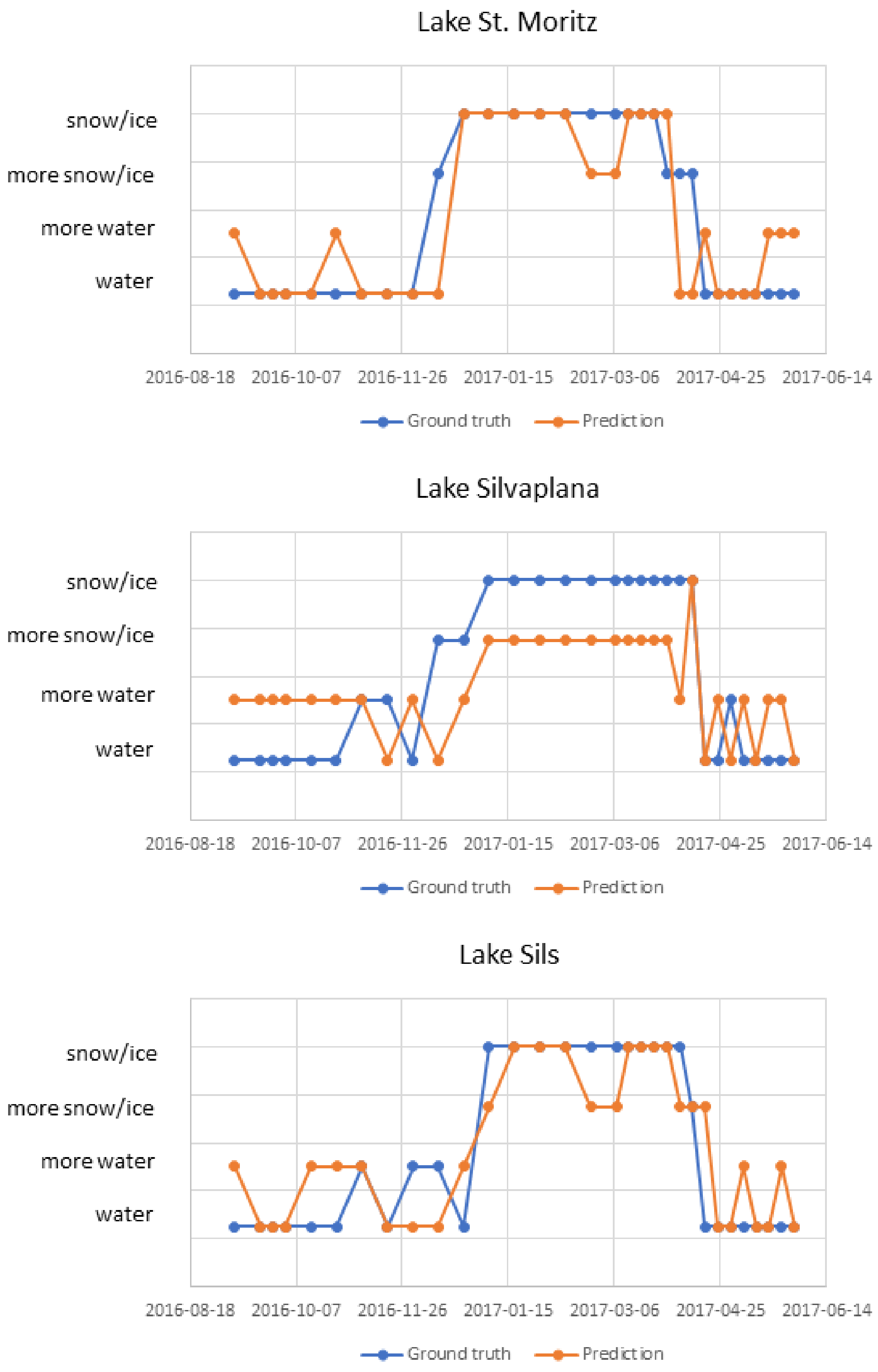


Lake St. Moritz

Lake Silvaplana

Lake Sils

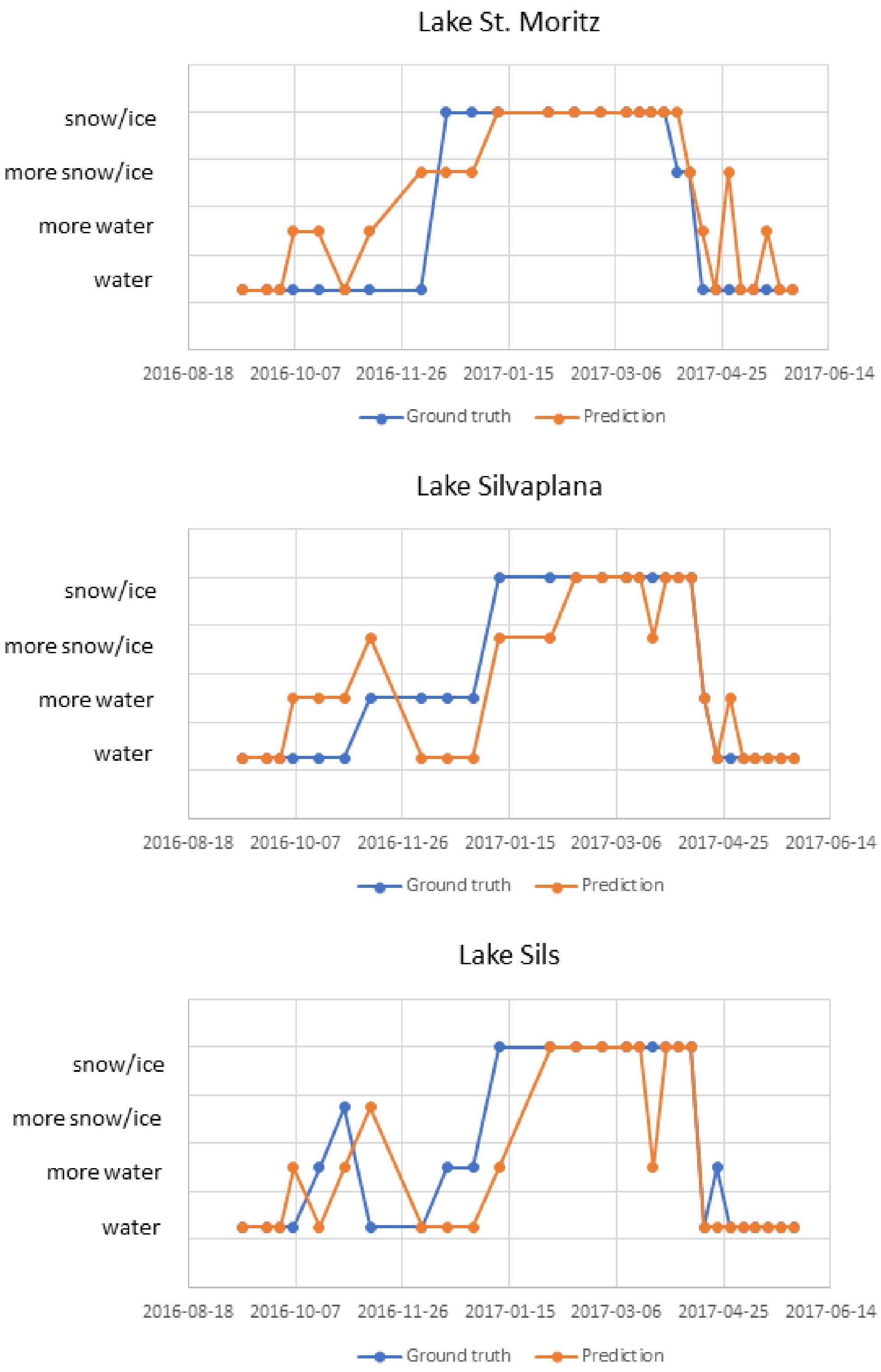
Figure 34. Timeline orbit 15


Lake St. Moritz

Lake Silvaplana

Lake Sils

Figure 35. Timeline orbit 117.



Lake St. Moritz

Lake Silvaplana

Lake Sils

Figure 36. Timeline orbit 168.

The failure rate for all orbits can be calculated by comparing prediction with ground truth (prediction ≠ ground truth) and is shown in Tab. 13:

Table 13. Failure rate of the different orbit timeline experiments (single lakes and overall)

Name	Orbit 15	Orbit 66	Orbit 117	Orbit 168
St. Moritzsee	45.2%	44.8%	38.7%	35.7%
Silvaplanersee	35.5%	34.5%	83.9%	39.3%
Silsersee	48.4%	31.0%	41.9%	32.1%
overall	43.0%	36.8%	54.8%	35.7%

The best prediction is achieved with orbit 66 and orbit 168. However, only 63% of the predicted values are correct.

6 Discussion

The results shown in chapter 5 indicate that it is possible to train a deep learning network with SAR data to detect lake ice, but the results are not perfect. Different factors which influence the radar images have an impact on the prediction of the network. The following factors are discussed:

- Incident angle
- Wind
- Blank ice
- Wet snow
- Deeplab v3+ parameters

6.1 Incident angle

The experiments in chapters 5.1 to 5.6 show that the different orbits have a significant influence on the results. The orbits differ in the incident angle and the day time when the images are taken. However, experiment 5.1.2 shows that the later has no influence on the result. The prediction quality is about the same, independent if the image was taken at about 5 am or 5 pm. Therefore, it is fair to assume that the observed dependency of the results is due to the incident angle and that this factor influences the result of the prediction.

Fig. 37 shows the histograms of gray values (x-axis) of all pixels measured during all four orbit passages (VV polarization) in the discussed time range. The pixels have been allocated using ground truth values to water (blue) or snow/ice (orange). All four histograms of the four different orbits have a different distribution. There is no clear distinction between water and snow/ice. That means for Deeplab v3+ that it has to learn the differences between water and snow/ice using the information from bigger pixel structures within patches. Fig. 37 shows that the histograms of the orbits with a steeper incident angle on the region Sils are wider (orbit 66 and orbit 117). For the other two orbits (15 and 168) the histogram is slimmer but adding up a higher number of pixels per backscatter intensity. The presented results in chapter 5.1 imply the following prediction ranking.

Orbit 66 >> 15 = 117 > 168

In the timeline experiments the order of the ranking is slightly different. Orbit 66 and orbit 168 show good timeline experiment results summarized in Tab. 13. The reason for this different behavior of orbit 168 remains unclear. Further work has to be done on the influence of the incidence angle and it has to be analyzed in more detail.

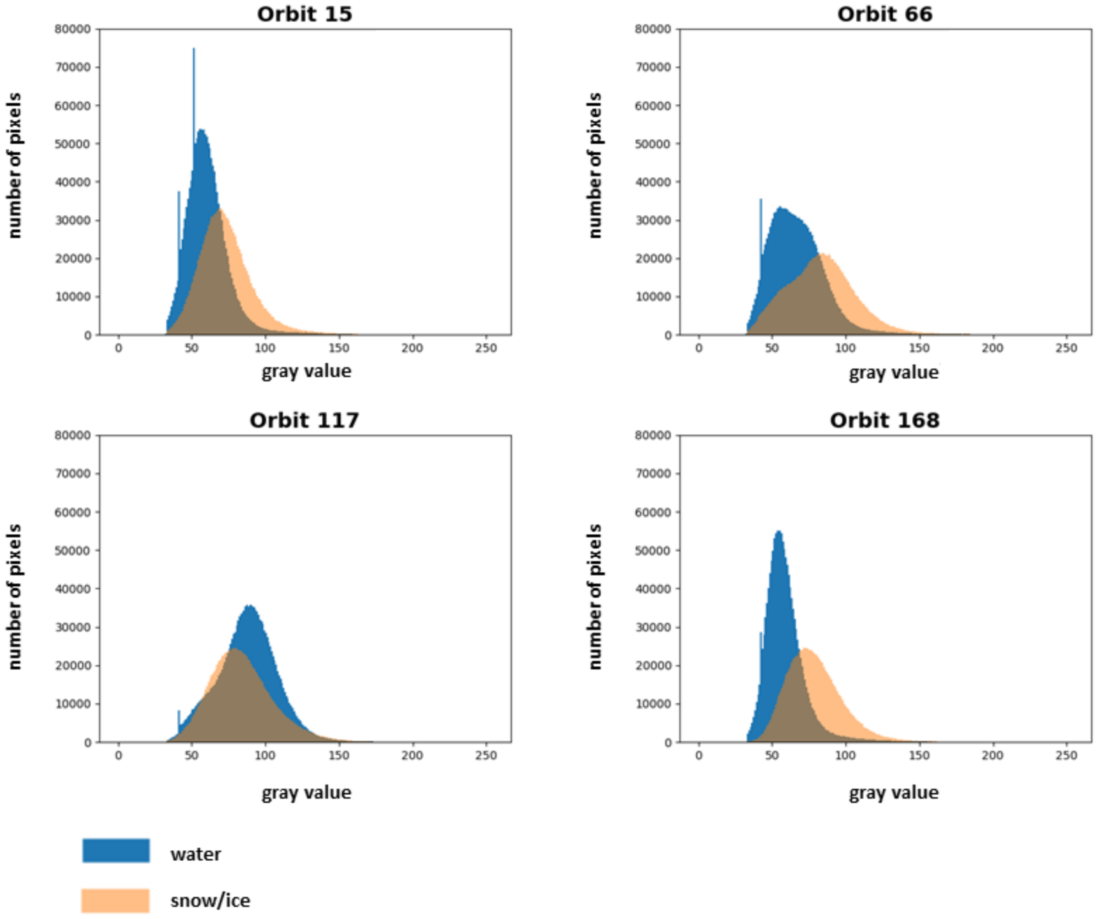


Figure. 37. Gray value distribution of the pixel backscatter values from the different orbits; blue: water, orange: snow/ice.

6.2 Wind

The time line experiments in chapter 5.6 show that it is difficult for Deeplab v3+ to predict water correctly. The reason for this issue could be wind. Fig. 38 shows the Lake Sils at two different days. At both days the surface of the Lake Sils is labeled ‘water’. On 24.4.2017, there is a lot of wind, on 7.10.2016 there is nearly no wind. The wind generates waves. The waves make the surface of the lake rougher. As described in chapter 3.3.2 there is a possible interaction of small waves (some centimeters) with the radar waves of the C band. The gray scale tonality in the image is significantly different. With a higher roughness more of the radar signal is backscattered to the satellite and the image appears brighter. It remains to be shown if a neural network is influenced by this difference.

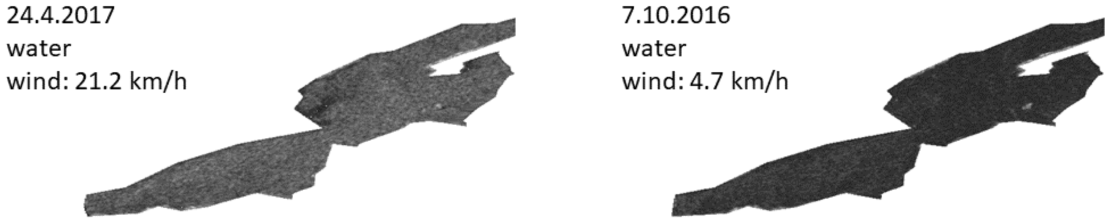


Figure 38. left: surface of Lake Sils on a windy day, right: surface of Lake Sils with nearly no wind.

To check this hypothesis, the wind data of the region of Sils have been studied. There is a weather check point in Segl-Maria, between the Lake Sils and the Lake Silvaplana. Meteo Swiss offers hourly mean values of the wind speed for research purposes. Fig. 39 shows the results of the timeline experiment of orbit 66 (blue: ground truth, orange: prediction) and compares it with the wind speed (gray: top peaks in the range of 25 km/h). However, no clear correlation can be found between wrong predictions during the non-frozen time and high wind speed. A deeper analysis of this factor should be done in the future.

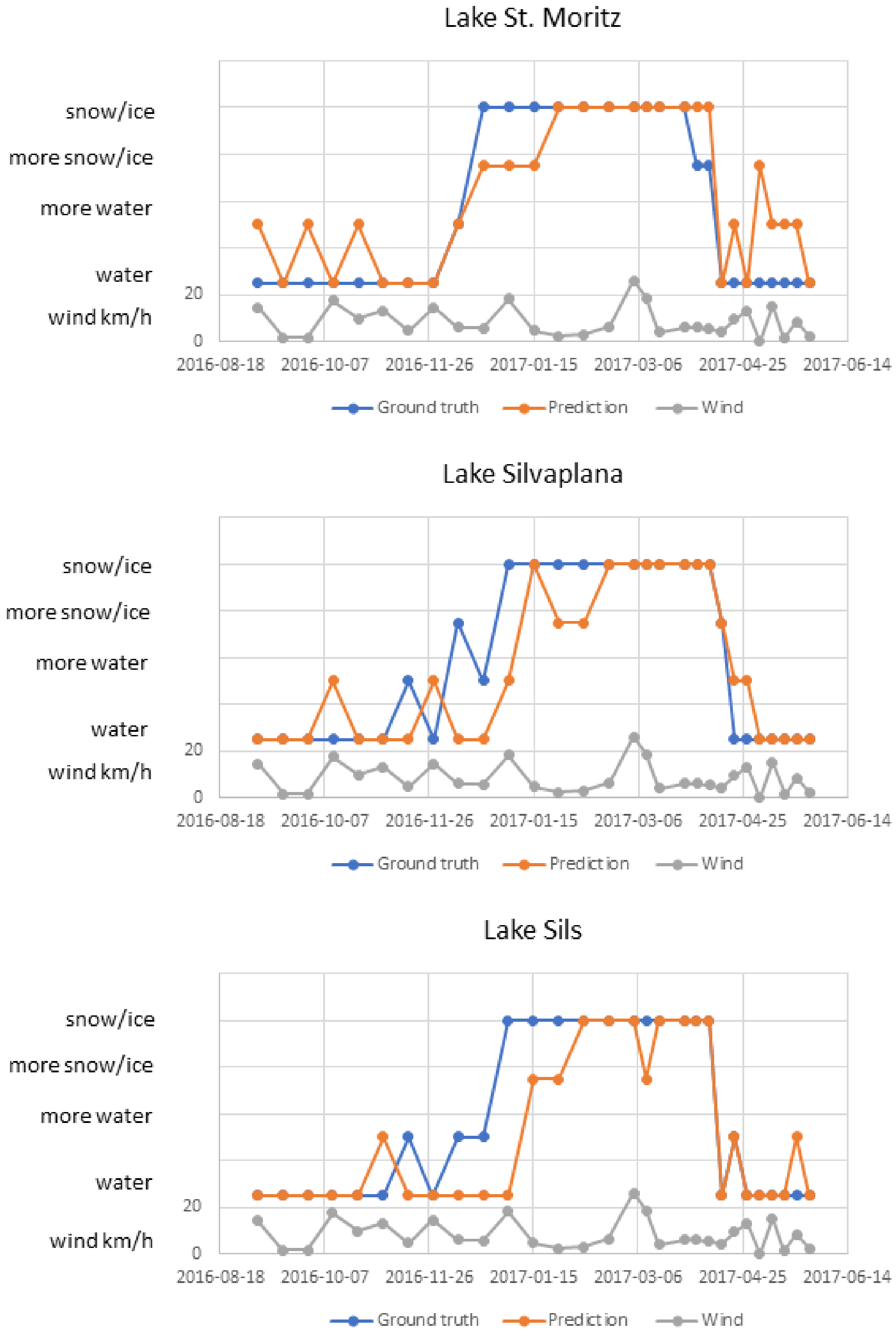


Figure 39. Timeline of the result of orbit 66 data with wind speed (hourly mean wind at timepoint when image was taken).

6.3 Snow-free ice

The timeline experiments in chapter 5.6 show that the network predicts snow/ice in nearly all cases with a delay of some days. A reason for this effect could be the missing precipitation in December 2016 and January 2017. Fig. 40 shows the precipitation for the whole investigation period between 1.10.2016 to 30.4.2017 based on data from Meteo Swiss. The winter was very dry and there was nearly no precipitation in December and the first half of January. On the other side the temperatures have been low and the surface of the lakes froze during this period completely, resulting in a snow-free ice layer with no/minimal snow coverage. As described in chapter 3.3.3 radar waves pass the blank ice layers and are reflect back from the water surface giving probably raise to the wrong prediction of Deeplab v3+. It would be interesting to follow up on this hypothesis and investigate other winter data sets when the freezing period is accompanied by strong precipitation. However, it has to be also considered that the incident angle plays a major role in the radar response and probably superimposes on this effect.

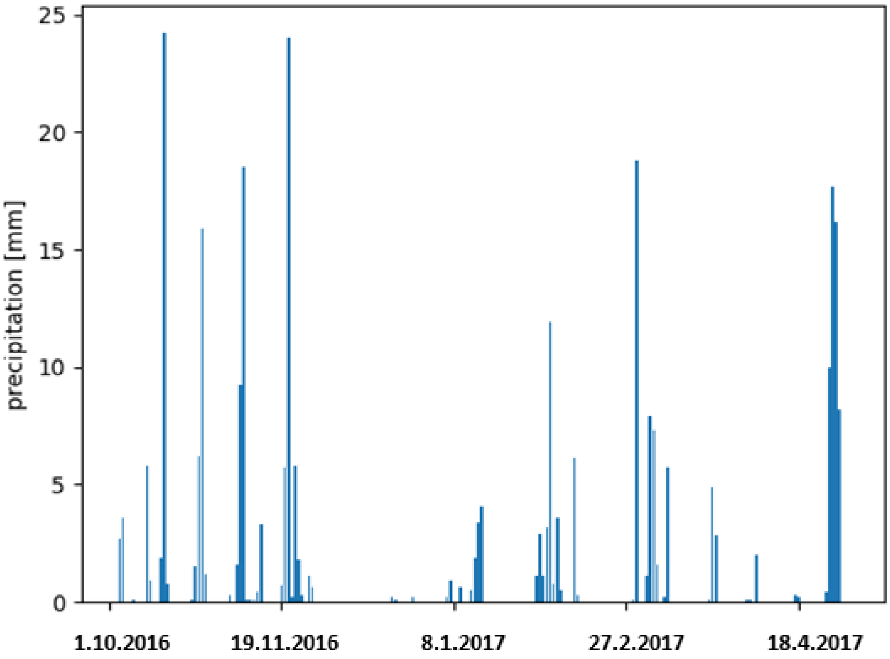


Figure 40. Precipitation in Segl Maria 1.10.2016 – 30.4.2017 (based on data from Meteo Swiss).

6.4 Wet snow

The timeline experiments in chapter 5.6 are also showing that the network has additional prediction problems during the frozen and the thawing period. This could be probably explained by wet snow on the ice and would be more pronounced in the orbit data measured in the afternoon (15 and 117) which is indeed the case. However, based on the small number of data points available it is difficult to check this statement and it would have to be investigated with a bigger data set in more detail.

6.5 Deeplab v3+

The whole set of experiments done in this thesis was run with the same Deeplab v3+ configuration varying only the parameters ‘number of epochs’ and ‘learning rate’. For the future it would be interesting to vary the Deeplab v3+ configuration parameters and to possibly achieve better results with a different setup.

Deeplab v3+ is developed for RGB images (3 layers). However, all experiments in this thesis (expect the experiment in chapter 0) are run with gray value images (1 layer). Deeplab v3+ was able to make predictions out of this one-layer images however there was no investigation if this data setup influenced the outcome of the results. For the future it would be helpful to understand this influence and to explore if the adaptation of the Deeplab v3+ parameter setting could result in better predictions.

It would be also interesting to set up a smaller deep learning network than Deeplab v3+ and to check if such a smaller network would be able to detect lake ice in SAR data as well or even better.

7 Conclusion and future work

The goal of this thesis was to check if it is possible to detect lake ice based on radar data with deep learning algorithms. Lake ice has been defined by the World Meteorological Organization in the framework of the Global Climate Observing System (GCOS) as an important climate variable for the observation of climate change.

In summary, this thesis has shown that it is possible to detect lake ice on three alpine lakes in Switzerland based on the available radar data set from the Sentinel-1 satellite constellation of the Copernicus program of ESA. The precision of the prediction is still not at the desired level and the results achieved with optical images and webcams are better. Therefore, it is recommended to do further in-depth investigations to clarify the potential of radar data for the prediction of lake ice. One important reason for this lack of the precision of the prediction is the complex interaction mechanism of radar microwaves with the different surface states of water.

It has been shown in this thesis that there is a dependency of the results on the orbit and with this on the position of the satellite relative to the lakes. The incidence angle of the radar beam determines the intensity of the backscattered signal and with this the quality and homogeneity of the input data. Therefore, all attempts to use the whole dataset (4 orbits) were failing and good results could only be achieved by using the limited number of images of orbit 66 and partly orbit 168. With this, the repeat cycle of images is reduced up to 12 days (mainly in 2016) which is not what is needed to determine like ice and freezing/thawing days precisely, as asked for by GCOS. As there are now two Sentinel-1 satellites operable (also seen in data of early 2017), it would be worthwhile to investigate this dependency on the incidence angle with more recent data.

Another important factor is the interaction of the radar waves with the different surface structures and the different surface states (smooth water, water with waves, blank ice, ice covered with snow, ice covered with wet snow). This thesis has shown that there might be some effects, however, the limited time has not allowed to dig deeper and investigate these parameters and how they influence the results in detail.

To further evaluate the potential of radar data to detect lake ice precisely the following topics should be considered:

- More experiments with the existing data
    - Study the orbit dependency in more detail
    - Study the surface interaction in more detail
    - Vary Deeplab v3+ parameters
    - Apply smaller networks
- Get more data and apply the model as well to other lakes to check the applicability
- Multitemporal analysis

Doing all of this will reveal if radar data will become a valuable tool to detect lake ice in the future.

8 Appendix

Deeplab v3+ is freely available on GitHub. The software was installed on a local computer with a GeForce GTX 1080 Ti GPU and on a server (Leonhard) with 16 Nvidia GTX 1080 GPU's. For each training only one GPU is used. The training time is for most of the experiments between 6 and 8 hours.

There is no specific instruction from the developers of Deeplab v3+ on how to adapt the code to a new dataset. Therefore two blogs (Sahu, 2018) ("DeepLabv3+ on your own dataset", 2019) have been used to adapt Deeplab v3+ to our dataset.

Lake ice detection was modeled as a 3 class (water, snow/ice, background) problem. The folder structure is very critical for the implementation of Deeplab v3+ and has to be followed.

For the 'sar ice' dataset the following folder structure is used:

- Dataset
    - JEPGImages
    - SegmentationClass
    - SegmentationClassRaw
    - ImageSets
    - Tfrecord

JEPGImages: contains the original patches from the training and validation dataset

SegmentationClass: contains the ground truth in RGB

SegmentationClassRaw: contains the ground truth in as a color indexed format

ImageSets: contains three text files,

- train.txt contains the name of the patches which are needed for the training,
- val.txt contains the name of the patches which are needed for validation,
- trainval.txt contains all the patch names which are needed for training and validation

TFrecord: contains the howl data from the folder dataset in a tfrecord format, this format is better for giving the data to the network.

It is necessary that all patches are named with the same name in the different subfolders of the folder dataset. For the training Deeplab v3+ needs the images out of the folder JEPGImages and SegementationClassRaw. The images in the folder SegementationClassRaw are generated from the images from the folder SegementationClass. Normally that happens with the file remove_gt_colormap.py. However, it does not work for the 'sar ice' dataset. Therefore the program of (Sahu, 2018) is used. This program converts the ground truth from a color image to a grayscale image. For each color in the ground truth a grayscale value is set. Deeplab v3+ needs the labels in an increasing order starting from zero for the background. It is not allowed that there is a gap between two grayscale values.

Additionally, the colormap for the dataset 'sar ice' must be added in the file get_dataset_colormap.py. In addition, in vis.py on line 83 the name of the colormap must be added. If these changes are not made there is the possibility that a colormap of Cityscape or Pascal is used (it belongs to the default parameters).

9 References

9.1 Bibliography

Albertz, J. (2009). *Einführung in die Fernerkundung* (4th ed.). WBG.

Albright, W. (2002, August 5). *Sar_noise*. Retrieved from http://www2.gi.alaska.edu/~rgens/teaching/asf_seminar/2002/sar_noise_floor.pdf

Alle Seen der Schweiz: Die Seeplattform. (2019, May 12). Retrieved May 12, 2019, from schweizersee.ch website: https://www.schweizersee.ch/

Bilinear interpolation - Wikipedia. (2019, May 28). Retrieved May 28, 2019, from https://en.wikipedia.org/wiki/Bilinear_interpolation

Canada Natural Resources. (2008, January 29). Radar Image Distortions. Retrieved May 12, 2019, from https://www.nrcan.gc.ca/earth-sciences/geomatics/satellite-imagery-air-photos/satellite-imagery-products/educational-resources/9325

Chen, L.-C., Zhu, Y., Papandreou, G., Schroff, F., & Adam, H. (2018). *Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation*. 801–818. Retrieved from http://openaccess.thecvf.com/content_ECCV_2018/html/Liang-Chieh_Chen_Encoder-Decoder_with_Atrous_ECCV_2018_paper.html

Chollet, F. (2018). *Deep learning with Python*. Shelter Island, New York: Manning Publications Co.

DeepLabv3+ on your own dataset. (2019, May 9). Retrieved May 9, 2019, from http://hellodfan.com/2018/07/06/DeepLabv3-with-own-dataset/

Earth Engine Code Editor. (2019, May 31). Earth Engine Code Editor. Retrieved May 31, 2019, from https://code.earthengine.google.com/

ESA Open Access Hub. (2019, May 31). Open Access Hub. Retrieved May 31, 2019, from https://scihub.copernicus.eu/

GCOS. (2019, May 12). Retrieved May 12, 2019, from https://gcos.wmo.int/

Google Drive. (2019, May 31). Google Drive – Cloud-Speicherplatz und Datensicherung für Fotos, Dokumente und mehr. Retrieved May 31, 2019, from https://www.google.com/drive/

Google Earth. (2017). *Meet Earth Engine*. Retrieved from https://www.youtube.com/watch?v=gKGOeTFHnKY

Google Earth Engine. (2019, May 11). Sentinel-1 Algorithms Google Earth Engine API. Retrieved May 11, 2019, from Google Developers website: https://developers.google.com/earth-engine/sentinel1

Gümgümcü, H. (2018, June 21). *Detecting deforestation using Sentinel-1 SAR data and deep learning*.

He, K., Zhang, X., Ren, S., & Sun, J. (2015). Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *37*(9), 1904–1916. https://doi.org/10.1109/TPAMI.2015.2389824

Liang-Chieh et al, C. (2019). *Models and examples built with TensorFlow. Contribute to tensorflow/models development by creating an account on GitHub* [Python]. Retrieved from https://github.com/tensorflow/models (Original work published 2016)

Miranda, N. (2015, May 21). *Radiometric Calibration of S-1 Level-1 Products Generated by the S-1 IPF*.

Moreira et al, A. (2013). *A Tutorial on Synthetic Aperture Radar*. 6–43.

Muyan, X. (2018, July). *Lake Ice Detection using Webcam Images*.

Muyan, X., Rothermel, M., Tom, M., Galliani, S., Baltsavias, E., & Schindler, K. (2018). *LAKE ICE MONITORING WITH WEBCAMS*. ISPRS Annals of the Photogrammetry.

Overview Copernicus. (2019, May 12). Retrieved May 12, 2019, from European Space Agency website: http://www.esa.int/Our_Activities/Observing_the_Earth/Copernicus/Overview4

Rander-Pehrson et al, G. (1999, July 14). PNG Specification: Data Representation. Retrieved May 8, 2019, from http://www.libpng.org/pub/png/spec/1.2/PNG-DataRep.html

Sahu, B. (2018, September 24). How to use DeepLab in TensorFlow for object segmentation using Deep Learning. Retrieved May 9, 2019, from freeCodeCamp.org website: https://medium.freecodecamp.org/how-to-use-deeplab-in-tensorflow-for-object-segmentation-using-deep-learning-a5777290ab6b

SAR Instrument - Sentinel-1 SAR Technical Guide - Sentinel Online. (2019, May 11). Retrieved May 11, 2019, from https://sentinel.esa.int/web/sentinel/technical-guides/sentinel-1-sar/sar-instrument

Schwaizer, G. (2017, July 7). *SAR / Optical Applications to Ice and Snow*. Retrieved from https://earth.esa.int/documents/10174/3166008/ESA_Training_Vilnius_07072017_SAR_Optical_Snow_Ice_Lecture_Schwaizer.pdf

Sentinel-1 - Data Products. (2019, May 11). Retrieved May 11, 2019, from https://sentinel.esa.int/web/sentinel/missions/sentinel-1/data-products

Sentinel-1 - ESA EO Missions. (2019, May 11). Retrieved May 11, 2019, from https://earth.esa.int/web/guest/missions/esa-operational-eo-missions/sentinel-1

Sentinel-1 - Instrument Payload. (2019, May 11). Retrieved May 11, 2019, from https://sentinel.esa.int/web/sentinel/missions/sentinel-1/instrument-payload

Sentinel-1 - Mission Summary. (2019, May 11). Retrieved May 11, 2019, from https://sentinel.esa.int/web/sentinel/missions/sentinel-1/overview/mission-summary

Sentinel-1 - Overview. (2019, May 12). Retrieved May 12, 2019, from https://sentinel.esa.int/web/sentinel/missions/sentinel-1/overview

Sentinel-1 - Single Look Complex. (2019, May 11). Retrieved May 11, 2019, from https://sentinel.esa.int/web/sentinel/technical-guides/sentinel-1-sar/products-algorithms/level-1-algorithms/single-look-complex

Sentinel-1- Observation Scenario. (2019, May 11). Retrieved May 11, 2019, from https://sentinel.esa.int/web/sentinel/missions/sentinel-1/observation-scenario

Tom, M., Kälin, U., Sütterlin, M., Baltsavias, E., & Schindler, K. (2018). *LAKE ICE DETECTION IN LOW-RESOLUTION OPTICAL SATELLITE IMAGES*. ISPRS Annals of the Photogrammetry.

Wang, C.-F. (2018, August 14). A Basic Introduction to Separable Convolutions. Retrieved May 15, 2019, from Towards Data Science website: https://towardsdatascience.com/a-basic-introduction-to-separable-convolutions-b99ec3102728

Wang, J., Duguay, C., Clausi, D., Pinard, V., & Howell, S. (2018). Semi-Automated Classification of Lake Ice Cover Using Dual Polarization RADARSAT-2 Imagery. *Remote Sensing*, *10*(11), 1727. https://doi.org/10.3390/rs10111727

Zulkifli, H. (2018, January 21). Understanding Learning Rates and How It Improves Performance in Deep Learning. Retrieved May 15, 2019, from Towards Data Science website: https://towardsdatascience.com/understanding-learning-rates-and-how-it-improves-performance-in-deep-learning-d0d4059c1c10

9.2 List of Figures

9.3 List of Tables