Landslide detection and mapping using Synthetic Aperture Radar amplitude imagery

M. J. Machielse

TU Delft

Deltares

Landslide detection and mapping using Synthetic Aperture Radar amplitude imagery

by

M. J. Machielse

to obtain the degree of
Master of Science in Civil Engineering
Specialisation: Geoscience and Remote Sensing

at the Delft University of Technology,
to be defended publicly on Monday November 1st, 2021 at 11:00 AM.

Student number: 4491033

Project duration: Thesis committee:	February 15, 2021 – November 1, 2021	
	Dr. R. C. Lindenbergh,	TU Delft Geoscience and Remote Sensing,
	Dr. T. A. Bogaard,	TU Delft Water Management,
	Dr. Ir. F. J. van Leijen,	TU Delft Geoscience and Remote Sensing,
	Ph.D. Ir. F. S. Tehrani,	Deltares,
	Ph.D. Ir. G. Donchyts,	Deltares,
	MSc. Ir. G. Santinelli ,	Deltares

An electronic version of this thesis is available at http://repository.tudelft.nl/.

TU Delft

Deltares

ABSTRACT

There is a growing demand for detailed and accurate landslide maps and inventories worldwide. The mapping of landslides is essential for emergency response, disaster mitigation and a better understanding of landslides. Currently, it is still difficult to detect the timing and extent of landslides accurately through satellite imagery. Most often optical satellite imagery is used, but this is limited since it requires daylight as well as cloud-free conditions in order to observe anything on the Earth's surface. Synthetic Aperture Radar (SAR) amplitude imagery shows great potential since it can overcome these disadvantages, the speckle and geometric effects make it difficult to detect landslides. Both SAR amplitude images without temporal averaging and with temporal averaging, within a time window of one month, were tested. This study introduces the use of SAR amplitude images by two Deep Learning (DL) models, a U-Net and a Pix2Pix conditional Generative Adversarial Network (cGAN), to detect and map landslides. These two models are trained and tested on several regions in South-East Asia where landslides have occurred recently. The two models were not able to detect and map the landslides based on the SAR amplitude imagery, as shown by the mean Intersection over Union (mIoU) values which are smaller than 0.01 and the inaccurately predicted images. Speckle reduction by temporal averaging of pre- and post-event images and consequently taking the log-based amplitude ratio, A_{ratio}, is recommended when using SAR amplitude imagery. Additionally, reduction of geometric effects by averaging images of descending and ascending order together is recommended. Despite the application of these measures, the two DL models used in this study don't have a successful outcome. In fact, a positive A_{ratio} returns landslide scars better than the models. The landslide detection becomes more effective when temporal averaging is done over more time, although this is impractical for timely landslide detection. In general optical imagery is recommended over SAR amplitude imagery, if available within three months after the landslide event. Otherwise, SAR amplitude imagery can be used, by using the aforementioned averaging and a threshold method on positive A_{ratio}.

Keywords: Synthetic Aperture Radar, landslide detection, U-Net, conditional Generative Adversarial Network.

ACKNOWLEDGEMENTS

I want to thank Deltares for giving me the opportunity to do my master thesis within the Data Science team. Although due to the Covid-19 pandemic I could not visit Deltares often, I felt part of the team by the regular meetings and fun events.

I want to express my thanks to my supervisors, who've helped me by our discussions, their feedback and their ideas. First of all thank you Faraz, for coming up with such an interesting idea to use conditional Generative Adversarial Networks for landslide detection and giving me the knowledge to continue your idea into reality. Thank you Genna, for the countless times you've helped me with programming and for introducing me to the world of TPU, TFRecords and Kaggle. Thank you Giorgio, for helping me find my place in Deltares and showing me the world of Deltares outside of my thesis. Also thank you for your advise on my thesis writing.

I am also very grateful for my supervisors from the TU Delft, who have guided me with their sharp observations and feedback. Thank you Roderik en Thom for our meetings, your feedback and advice. Also big thanks to Freek van Leijen for wanting to be my third supervisor.

Finally thanks to my friends and family who are always there for support. Writing a thesis is not always easy, especially in a pandemic when you are stuck at home, so thank you for keeping me sane.

Enjoy reading!

Marja Machielse
Delft, October 2021

CONTENTS

1 INTRODUCTION 1
1.1 Motivation . . . 1
1.2 Literature Review . . . 1
1.3 Goal of this study . . . 3
1.3.1 Hypothesis . . . 3
1.3.2 Restrictions . . . 4
1.3.3 Content . . . 4
2 THEORY 5
2.1 Resolution . . . 5
2.1.1 Spatial resolution . . . 5
2.1.2 Spectral resolution . . . 5
2.1.3 Temporal resolution . . . 5
2.1.4 Radiometric resolution . . . 6
2.2 Synthetic Aperture Radar . . . 6
2.2.1 Look angle and direction . . . 7
2.2.2 Polarization . . . 8
2.2.3 Speckle . . . 9
2.2.4 Amplitude . . . 9
2.2.5 Interferometric Synthetic Aperture Radar . . . 10
2.2.6 Amplitude vs. coherence based . . . 10
2.3 Machine learning methods . . . 10
2.3.1 Semantic segmentation . . . 11
2.4 Convolutional Neural Network . . . 11
2.4.1 Convolutional layers and max pooling . . . 12
2.4.2 Dropout . . . 13
2.4.3 Batch normalization . . . 13
2.4.4 Activation layers . . . 13
2.4.5 Loss functions . . . 14
2.5 Tuning of a Convolutional Neural Network . . . 16
2.5.1 Hyper-parameters . . . 16
2.5.2 Well generalized, over- and under-fitting . . . 17
2.5.3 Learning curve . . . 17
2.6 U-Net . . . 18
2.6.1 U-Net for landslide segmentation . . . 19
2.7 Generative Adversarial Networks . . . 19
2.7.1 Pix2Pix conditional Generative Adversarial Network . . . 20
2.7.2 Generative Adversarial Networks versus Convolutional Neural Networks . . . 22
2.7.3 Learning curve of Generative Adversarial Networks . . . 23
2.7.4 Generative Adversarial Networks for landslide segmentation . . . 24
2.8 Class imbalance . . . 24
2.8.1 Data augmentation . . . 24
2.9 Validation . . . 25
2.9.1 Training, testing and validation data sets . . . 25
2.9.2 Validation metrics . . . 25
3 METHODOLOGY 29
3.1 Ground truth data . . . 30
3.1.1 Landslides . . . 30
3.1.2 Landslide databases . . . 31
3.1.3 Digital Elevation Model satellite imagery . . . 35
3.1.4 Copernicus Sentinel-1 satellite imagery . . . 35

3.2 Representations . . . 37
3.2.1 Temporal averaging . . . 39
3.2.2 Use of masks . . . 40
3.2.3 A_{ratio} representation . . . 41
3.2.4 Ratioslope representation . . . 42
3.2.5 Ampslope representation . . . 42
3.3 Production of input images . . . 43
3.3.1 Independent (source) and dependent (target) images . . . 43
3.3.2 Rasterizing images into tile-sized images . . . 44
3.3.3 Image scaling . . . 45
3.3.4 Data augmentation . . . 47
3.4 Model setup . . . 47
3.4.1 Hyperparameter optimization . . . 47
3.4.2 U-Net . . . 48
3.4.3 Pix2Pix conditional Generative Adversarial Network . . . 49
3.5 Model evaluation . . . 50
3.5.1 Training, testing and validation data set ratio . . . 51
3.5.2 Landslide images and non-landslide images . . . 52
3.6 Software . . . 52
3.6.1 Processing Unit . . . 53
3.6.2 Optimal usage . . . 53
4 RESULTS 55
4.1 Human interpretation . . . 55
4.2 Tests with A_{ratio} representation . . . 61
4.3 Tests with ratioslope representation . . . 66
4.4 Tests with ampslope representation . . . 68
4.5 Discussion . . . 74
5 CONCLUSION 77
5.1 Recommendations . . . 78
A 87
A.1 Satellite images . . . 87
B 91
B.1 Definition of large landslides . . . 91
B.2 Additional tests with ampslope representation, featuring large landslides only . . . 91
B.3 Additional tests with ampslope representation, featuring all landslides 93

LIST OF FIGURES

Figure 2.1 Microwave emission, scattering and back-scattering of SAR. Source: [58]. 6
Figure 2.2 *left*: flight direction (A), nadir (B), swath (C), range(D), azimuth (E). *right*: incidence angle (A), look angle(B), slant range distance(C), ground range distance(D). Source: [4]. . . . 7
Figure 2.3 Schematic diagram of linear horizontal, linear vertical, right circular and left circularly polarized signals. The y-axis (vertical axis) is the electric field, the magnetic field is perpendicular to it on the x-axis (horizontal axis). Source: [79]. 8
Figure 2.4 Images of a parking lot and a palace (left) and the speckled SAR versions of those (right). Images obtained from [27]. . . . 9
Figure 2.5 left to right; object detection, semantic segmentation, instance segmentation. Source: [Gkioxari et al.]. 11
Figure 2.6 Max pooling with a 2*2 window. Source: [39]. 13
Figure 2.7 Learning curve showing under fit. Source: [8]. 18
Figure 2.8 Learning curve showing good fit. Source: [8]. 18
Figure 2.9 Learning curve showing over fit. Source: [8]. 18
Figure 2.10 U-Net architecture as described by [71]. 18
Figure 2.11 Scheme of the U-Net [71]. The source image is used as input, the output is a masked image. The mask gives the segmented landslide. 19
Figure 2.12 Results from landslide segmentation using a U-Net by [53]. a) and b) show incorrect extractions, c) is an correct extraction. The input images where optical images with red,green,blue and digital surface model, slope and aspect band. The overall accuracy of the method was 0.93. 19
Figure 2.13 Scheme of a conditional Generative Adversarial Network. The generator receives additional information y and/or random noise z. This than generates real and fake images and gives these to the discriminator. The discriminator also receives information x, which is a label referring to whether an image is real or fake. The output of the discriminator is whether the received images are real or fake. 20
Figure 2.14 Scheme of the composite model, with a source image as input which is translated to an output image. The output image is generated by the generator, both a fake and a real output image is made. Then the discriminator has to say which is true and which is fake. Through this process both generator and discriminator learn. The goal is to have a generator which generates the accurate output corresponding to the input image. 21
Figure 2.15 Plot of three source images, generated images and true target images on the first, second and last row respectively. 22
Figure 2.16 Learning curve of a Generative Adversarial Network (GAN) model showing a normal/expected loss. The "d-real" and "d-fake" plotted lines describe the discriminator losses with regards to real and fake targets respectively. The "gen" plotted line describes the generator loss. Source : [10]. 23
Figure 2.17 Intersection over Union. Source: [72]. 27

Figure 2.18 F1 score/dice coefficient is 2 * the area of overlap divided by the total number of pixels in both images [82]. 27
Figure 3.1 Workflow of the making of two representations, with regards to temporal averaging and masking. The first two representations are similar, so these are visualized in the same figure. Their difference lies in that one is masked based upon a slope image, while the other is not masked but has this slope image as an additional band. 29
Figure 3.2 Workflow of the third method to produce the input images. S-1 refers to the Copernicus Sentinel-1 satellite. t2 and t1 refer to the time of the image, where time 2 is later than time 1. 29
Figure 3.3 Workflow of the production of the training, testing and validation data set containing source and target images which will be used in the DL models. 30
Figure 3.4 The landslide database in Hiroshima, Japan. The red polygons (in the zoomed in figure these appear red, in the larger figure these are more black) are landslides. 32
Figure 3.5 The landslide database in Hokkaido, Japan. The smaller figures on the right are zoomed in figures from the larger figure. The red, dark red and orange polygons (in the zoomed in figure these appear red, in the larger figure these are more black) are landslides. The bright red polygons are landslides caused by the event described in this section, the dark red landslides are reactivated by this same event and the orange landslides are old landslides which are still visible after the event. 33
Figure 3.6 The landslide database in Lombok, Indonesia. The figure shows landslide polygons, and is zoomed in in two smaller figures on the bottom. The orange landslides are triggered by the event of 5 August 2018. The red polygons are landslides triggered by the subsequent event of 19 August 2018. 34
Figure 3.7 The landslide database in Tsangpo gorge, China. The pink polygons are landslides, the figure on the right is a zoomed in version. 35
Figure 3.8 Percentage of amount of pixels with value smaller than or equal to -30 decibel (dB) per SAR amplitude image. This is calculated over a random subset of approximately 3000 images from the training set. These images are smaller images made from the larger satellite images, which is detailed further in section 3.3. It can be concluded that almost all images include a percentage of smaller than -30 dB pixels, but that most of these percentages are smaller than 10 percent. 37
Figure 3.9 An area in the region of the Hiroshima Prefacture, Japan is selected, visible in the figure by the red circle. This area inside the circle will be used as example area to compare the different representations. 38
Figure 3.10 Images of the Copernicus Sentinel-2 (S-2) (optical) satellite, visualized in the red, green and blue band. The left image is the last available S-2 image before the landslide event, the right the first available S-2 image after the event. 38
Figure 3.11 Images of the Copernicus Sentinel-1 (S-1) SAR satellite, with amplitude dB values. The images are of descending direction. The left image is the last available S-1 image before the landslide event, the right the first available S-1 image after the event. 39

Figure 3.12 Mask image of landslide scars according to the (ground truth) landslide database. The white pixels represent landslide, the black pixels represent the background. 39
Figure 3.13 A slope image of the Digital Elevation Model (DEM) (left) and the corresponding mask (right), based on a degree larger than 0.5 degrees. 41
Figure 3.14 The A_{ratio} representations, with on the left temporal averaging done over multiple years and on the right after one month. This image is made in QGIS, these images are styled to a color scale to enhance the landslide detection. 42
Figure 3.15 On the left the A_{ratio} image is visualized, the same as in figure 3.14, but without the mask and in the original color. On the right the slope band is added, which results in the two-banded ratioslope image. 42
Figure 3.16 The ampslope images, where the left is of ascending order and the right of descending order. 43
Figure 3.17 Landslide polygons (left) and binary image of polygons (right) of the area of Tsangpo Gorge, China. 44
Figure 3.18 Tiling of geometry containing landslide polygons (in red) in the area of Hokkaido, Japan. 45
Figure 3.19 Source and target images of the three representations. The top row displays the A_{ratio} source image and the target image. The second row displays the ratioslope representation. The first two images show the two bands, the A_{ratio} and the slope band respectively. The third image shows the (ratioslope) source image, which consists of these two bands. The third row displays the ampslope representation. The first three images show the three bands, the SAR amplitude band pre- and post-image and the slope image respectively. The fourth image shows the (ampslope) source image, which consists of these three bands. 47
Figure 3.20 The input image and corresponding segmentation mask of the Oxford Pets dataset, on which the U-Net is trained originally. Source: [15]. 48
Figure 3.21 Scheme of the U-Net,with a source image as input and target images as output. The input image varies among the various tests, the first figure shows the A_{ratio}, the second the ratioslope and the third the ampslope image. The target image is a binary mask showing whether a pixel is a (part of) a landslide or not. The output image is generated by the U-Net, the goal is to produce the accurate output corresponding to the input image. 49
Figure 3.22 Scheme of the composite model,with a source image as input which is translated to an output image. The input image varies among the various tests, the target image is a binary mask showing whether a pixel is a (part of) a landslide or not. The output image is generated by the generator, both a fake and a real output image is made. Then the discriminator has to say which is true and which is fake. Through this process both generator and discriminator learn. The goal is to have a generator which generates the accurate output corresponding to the input image. 50
Figure 4.1 The target image shows the landslide scars. This image is a mask, which represents the landslide scars as white pixels and the background in black pixels. 56

Figure 4.2 This figure shows Copernicus Sentinel-2 (S-2) optical satellite images before (left) and after (right) a landslide event. The optical images are single images before and after the landslide, the least cloudy image within a month's range from the event is selected. . . . 56
Figure 4.3 This figure shows images before (left) and after (right) a landslide event. The top row shows images which have not been temporally averaged, the pre image is the last available image before the event and the post one is the last available image after the event. The bottom row shows images which have been temporally averaged over the course of one month. 57
Figure 4.4 This figure shows the post-event images, which have been temporally averaged over a month. The top image is averaged over images of ascending, the second of descending. The images of ascending and descending during a month post-event have been averaged together, resulting in the third image. . . . 58
Figure 4.5 This figure shows the log-based amplitude ratio, the A_{ratio}, for three images. The top image has not undergone any temporal averaging, the A_{ratio} is taken from a single pre- and post-event image. The second image shows an A_{ratio} where the pre- and post-event images are temporally averaged over the course of one month. The third images shows an A_{ratio} where the pre- and post-images are temporally averaged over the course of two years. The images are visualized with a color range. . . . 59
Figure 4.6 Masked form of A_{ratio} images, where a threshold of A_{ratio} bigger than 2.5 is taken. TLR: masked A_{ratio} without temporal averaging, with temporal averaging of one month, with temporal averaging of two years, ground truth target image. . 60
Figure 4.7 The ampslope representation (left) and ratioslope representation (right). . . . 60
Figure 4.8 Tile-sized images, with the shape and size of which the images are trained in the U-Net and cGAN. RTL: The A_{ratio} without temporal averaging, A_{ratio} with temporal averaging of one month, the ampslope, the ratioslope and the target image. . . . 61
Figure 4.9 Learning curves showing the loss for the A_{ratio} representation from the cGAN. The 'gen' line is the loss of the generator, the 'd-real' is the loss of the discriminator with regards to the real targets and the 'd-fake' is the loss of the discriminator with regards to the generated fake targets. TLR: The graphs are made with different batch sizes: 64,32,16 and 8 respectively. The x-axis shows the amount of tests, e.g. the amount of epochs * the amount of batches per epoch. The amount of tests is therefore higher with a smaller batch size. The y-axis shows the loss value. . . . 63
Figure 4.10 Learning curves showing the mean Intersection over Union (mIoU) for the A_{ratio} representation from the cGAN. TLR: batch sizes of 64,32,16 and 8 respectively. . . . 64
Figure 4.11 Learning curve showing the mean Intersection over Union (mIoU) for the A_{ratio} representation from the cGAN. The Intersection over Union (IoU) values which contribute to the mIoU are only calculated on landslide images. The batch size for this test was 64. . . . 64

Figure 4.12 Learning curve showing the loss and the Mean Intersection over Union (mIoU) for the A_{ratio} representation from the U-Net. The batch size for this test was 64. 65
Figure 4.13 Source, target and generated target images for the A_{ratio} representation produced by the cGAN. The batch size of this test was 8. 65
Figure 4.14 True target and predicted target from testing data set of the A_{ratio} representations, with batch size 64. 66
Figure 4.15 Source, target and generated target images for the ratioslope representations. The generated target images only contained 'not a number' (not a number (NaN)) values, therefore these are not visualized. The batch size of this test was 64. 67
Figure 4.16 Learning curves showing the loss for the ampslope representation featuring large landslides only from the cGAN. The 'gen' line is the loss of the generator, the 'd-real' shows the loss of the discriminator with regards to the real targets and the 'd-fake' shows the loss of the discriminator with regards to the generated fake targets. The top figure shows the learning curve for the data set of ascending direction, the middle for that of descending direction, and the last a combination of the two. The batch size of these models are 64. The x-axis shows the amount of tests, e.g. the amount of epochs * the amount of batches per epoch. The y-axis shows the loss value. 69
Figure 4.17 Learning curves showing the mean Intersection over Union mIoU for the ampslope representation featuring large landslides only from the cGAN. The top figure shows the learning curve for the data set of ascending direction, the middle for that of descending direction, and the last a combination of the two. 70
Figure 4.18 Learning curves showing the loss and mean Intersection over Union mIoU for the ampslope representation featuring large landslides only from the U-Net. The top figure shows the learning curve for the data set of ascending direction, the middle for that of descending direction, and the last a combination of the two. 71
Figure 4.19 Source, target and generated target images for the ampslope representation featuring large landslides only of ascending direction from the cGAN, with batch size 32. 72
Figure 4.20 True target and predicted target from testing data set of the ampslope representation featuring large landslides only of ascending direction from the cGAN,with batch size 32. 73
Figure B.1 The minimum sizes of the landslides defined as 'large' for each region, TLR: Hiroshima, Japan, Hokkaido, Japan, Lombok, Indonesia and Tsangpo Gorge, China. 91
Figure B.2 Source, target and generated target images for the ampslope representation featuring large landslides only of descending direction from the cGAN, with batch size 32. 92
Figure B.3 Source, target and generated target images for the ampslope representation featuring large landslides only of both ascending and descending direction combined from the cGAN, with batch size 32. 92
Figure B.4 True target and predicted target images for the ampslope representations featuring large landslides only from the cGAN, of descending direction with batch size 32. 93

Figure B.5 True target and predicted target images for the ampslope representations featuring large landslides only from the cGAN, of both ascending and descending direction combined with batch size 32. 93
Figure B.6 Learning curves showing the loss for the ampslope representation featuring all landslides from the cGAN. The 'gen' line is the loss of the generator, the 'd-real' shows the loss of the discriminator with regards to the real targets and the 'd-fake' shows the loss of the discriminator with regards to the generated fake targets. The top figure shows the learning curve for the data set of ascending orbit, the middle for that of descending orbit, and the last a combination of the two. The batch size of these models are 64. The x-axis shows the amount of iterations, e.g. the amount of epochs * the amount of batches per epoch. The y-axis shows the loss value. 94
Figure B.7 Learning curves showing the mean Intersection over Union mIoU for the ampslope representation featuring all landslides from the cGAN. The top figure shows the learning curve for the data set of ascending orbit, the middle for that of descending orbit, and the last a combination of the two. With batch size 64. 95
Figure B.8 Learning curves showing the loss and mean Intersection over Union mIoU for the ampslope representation featuring all landslides from the U-Net. The top figure shows the learning curve for the data set of ascending orbit, the middle for that of descending orbit, and the last a combination of the two. With batch size 64. 96
Figure B.9 Source, target and generated target images for the ampslope representation featuring all landslides of ascending direction from the cGAN, with batch size 64. 97
Figure B.10 Source, target and generated target images for the ampslope representation featuring all landslides of descending direction from the **Cgan!** (Cgan!), with batch size 64. 98
Figure B.11 Source, target and generated target images for the ampslope representation featuring all landslides with a combination of ascending and descending direction from the cGAN,with batch size 64. 99
Figure B.12 True target and predicted target of the testing data set of the ampslope representation featuring all landslides from the cGAN, of ascending direction with batch size 64. 99
Figure B.13 True target and predicted target of the testing data set of the ampslope representation featuring all landslides from the cGAN, of descending direction with batch size 64.. 100
Figure B.14 True target and predicted target of the testing data set of the ampslope representation featuring all landslides from the cGAN, with a combination of descending and ascending direction with batch size 64. 101

LIST OF TABLES

Table 3.1 The time range of the landslide events per region considered in this study. 31
Table 3.2 Amount of satellite images used for each region. 36
Table 3.3 Amount of images with and without landslides pixels for each region and in total. 45
Table 3.4 Minimum and maximum pixel values of the ampslope source (with bands SAR and DEM), A_{ratio} source and target images. The values to which these minima and maxima are scaled are specified in the last two columns. The minima and maxima are calculated from the training set only. *The A_{ratio} with mask and without have roughly the same minima and maxima values, the values displayed here have been rounded to two digits. 46
Table 3.5 Amount of images for the training, testing and validation sets of the ampslope images. The training set is both made for 'large' landslides and all landslides, but the testing and validation set are with all landslides only. Ascending and descending refer to the orbit direction, combined is the combination of both the ascending and descending sets. Both ascending, descending and combined sets are used. 51
Table 3.6 Amount of images for the training, testing and validation sets of the A_{ratio} and ratioslope representations. 51
Table A.1 S-1 SAR images used for Hiroshima region 87
Table A.2 S-1 SAR images used for Hokkaido region 88
Table A.3 S-1 SAR images used for Lombok region 88
Table A.4 S-1 SAR images used for Tsangpo Gorge region 89

ACRONYMS

DEM Digital Elevation Model . . . ix
SAR Synthetic Aperture Radar . . . i
GEE Google Earth Engine . . . 4
DL Deep Learning . . . i
CML Conventional Machine Learning . . . 3
S-1 Copernicus Sentinel-1 . . . viii
S-2 Copernicus Sentinel-2 . . . viii
CNN Convolutional Neural Network . . . 3
VV vertical transmit / vertical receiver . . . 8
HH horizontal transmit / horizontal receiver . . . 8
VH vertical transmit / horizontal receiver . . . 8
HV horizontal transmit / vertical receiver . . . 8
GCS Google Cloud Services . . . 53
InSAR Interferometric Synthetic Aperture Radar . . . 2
IW Interferometric Wide Swath . . . 35
GRD Ground Range Detected . . . 4
SRTM Shuttle Radar Topography Mission . . . 35
dB decibel . . . viii
ReLU Rectified Linear Unit . . . 14
PolSAR Polarimetric Synthetic Aperture Radar . . . 4
ANN Artificial Neural Network . . . 11
SVM Support Vector Machine . . . 11
RF Random Forest . . . 11
VV + VH vertical transmit/horizontal receiver . . . 8
HH + HV horizontal transmit/vertical receiver . . . 8
GSI Geospatial Information Authority of Japan . . . 31
GAN Generative Adversarial Network . . . vii
cGAN conditional Generative Adversarial Network . . . i
ee Earth Engine . . . 53
CPU Central Processing Unit . . . 53
GPU Graphics Processing Unit . . . 53
TPU Tensor Processing Unit . . . 53
NaN not a number . . . xi
IoU Intersection over Union . . . x
mIoU mean Intersection over Union . . . i
TP True Positive . . . 25
FP False Positive . . . 25
TN True Negative . . . 25
FN False Negative . . . 26
TPR True Positive Rate . . . 26
FPR False Positive Rate . . . 26

1 INTRODUCTION

This chapter introduces the topic of landslide detection and mapping. First the motivation for this topic will be explained. Then a literature review will be done, leading to the topic and scope, as well as the restrictions of this research.

1.1 MOTIVATION

Landslides occur frequently worldwide and cause death and severe damage to natural and human infrastructure [21]. Aside from this initial damage, landslides have a direct long-term economic and social impact [32]. The World Bank has identified a total land area of 3.7 million square kilometers under risk of landslides, from which 820 thousand square kilometers are high risk zones [17]. This affects 300 million people. For example, the 2018 Hokkaido Eastern Iburi earthquake triggered many landslides, which resulted in 31 severe injuries and 42 deaths. Many houses were destroyed and land was left barren [43; 36]. Natural disasters occur more often due to climate change and global warming. Thus landslides also increase in occurrence and pose an increasing danger to people and property [11].

There is a growing demand for detailed and accurate landslide maps and inventories worldwide. The mapping of landslides is important for emergency response, disaster mitigation and better understanding of landslides [28], [24]. Previously this mapping was done manually, only that is a laborious and sometimes impossible task since landslides may occur in remote regions. Many response efforts are impeded by a lack of information on the condition or location of damaged areas [49; 70]. In addition to rapid response it is important to create accurate landslide inventories to improve our understanding of where landslides occur. Additionally detailed and accurate landslide maps and inventories are crucial data for landslide prediction, which is currently still limited in terms of data amount.

Landslide detection and mapping is a problem in both time and space. Landslide detection is needed to find the time of occurrence of the landslide, mapping is needed to find the extent ('space') of the landslide. Currently it is still difficult to detect the time of occurrence and extent of landslides accurately. Therefore a landslide detection and mapping algorithm should be developed, which has the potential of worldwide generalization. The input data should be freely available satellite data, such that the algorithm is broadly usable.

1.2 LITERATURE REVIEW

Optical satellite imagery is often used to detect landslides (e.g. [14], [56], [49]). Optical data is limited since it requires daylight as well as cloud-free conditions in order to observe anything on the Earth's surface. It is therefore not always possible to detect landslides with optical imagery. For example, following the 2015 Mw 7.8 Gorkha earthquake, which triggered more than 25,000 landslides, cloud cover prevented the use of optical imagery for landslide mapping for more than a week [70].

Robinson et al. [70] demonstrated that this delay would have been much longer had the monsoon occurred during Nepal's monsoon season.

In comparison to the abundance of papers on landslide detection using deep learning on optical imagery, limited papers have explored the use of deep learning on Synthetic Aperture Radar (SAR) imagery. The most important advantage of SAR imagery opposed to optical imagery is that it penetrates clouds and operates day and night. Thus it has more coverage both spatially and temporally than optical imagery. Especially in cases where bad weather prevents the use of optical imagery, which is often the case for landslides since they are typically triggered by heavy rainfall and/or earthquakes.

SAR properties can be used in different ways to detect landslides, methods have been based on SAR amplitude ([44], [28]), coherence ([13]), polarimetry ([61],[66]) or some combination of the afore mentioned ([36]). Many studies focus on Interferometric Synthetic Aperture Radar (InSAR), which measures the radar phase change between two acquisitions in order to quantify ground surface deformation [28]. Changes in amplitude and coherence occur after an event (such as landslide) has happened. In densely vegetated regions, where landslides often occur, the coherence is always low. Therefore coherence-based methods do not work well in these cases, but amplitude-based methods possibly do [36]. Furthermore the processing of SAR amplitude imagery is far less time consuming and complicated than that of coherence and polarimetric imagery.

However, the processing in getting valuable information from SAR imagery is generally more complicated than for optical imagery. SAR satellite imagery is limited by geometric effects, such as foreshortening, layover and shadow in mountainous area [43]. Since landslides generally occur in hilly and mountainous areas, this is a problem to be considered. Furthermore SAR suffers from speckle and is therefore difficult to interpret.

To deal with these effects, SAR images are often averaged over time. For example, numerous papers [28; 44] use aggregation of images before a landslide event and images after a landslide event over time. This results in composite images or stacked images, e.g. the result of a certain averaging metric (e.g. the median) over the various images. Then the difference between these two images enables change detection, which may be a sign of landslide occurrence. Handwerger et al. [28] found that increasing the number of SAR images used in pre- and post-event stacks helps to reduce speckle in the SAR data. This increase in the number of images comes with an increase in the amount of time used to average over. Handwerger et al. [28] found that the best landslide detection results were obtained when the number of images in the post-event stack were increased over a period covering two years. This would imply that a landslide can only be detected two years after it has occurred. Temporal averaging, especially of the post event images, of more than two or three months is impractical and not desirable. Furthermore if more than two or three months of averaging post-event is required, the advantage of SAR over optical imagery is not relevant anymore. Hence it is crucial to find whether temporal averaging is necessary for landslide detection and if it is, over what time span.

Landslide detection studies are generally tested on a single landslide event and using a single radar sensor. For example, [44; 58; 66; 40; 3; 86] tested their methods using ALOS-2 imagery of the 2018 Hokkaido earthquake. Unfortunately ALOS-2 is not freely available and is collected infrequently worldwide. Little studies have used freely available SAR data from the Copernicus Sentinel 1 (S-1) satellites [28; 12]. Few studies, such as [62], used a global selection of landslides. The S-1 data is generally of coarser spatial resolution than the ALOS-2. This study will focus on S-1

data, since this is publicly available and free satellite data. This study will focus on different kinds of landslides, as the goal is to produce an algorithm which can detect landslides worldwide. Several regions which have undergone landslides are considered, which results in different kinds of landslides.

Multiple studies, e.g. [67; 68], have shown that Deep Learning (DL) methods give better results for landslide detection and -mapping than Conventional Machine Learning (CML) methods. Furthermore the amount of feature engineering is greatly reduced, since DL methods take the raw image as training data while CML needs labeled segments [68]. A Convolutional Neural Network (CNN) is a popular DL method which is especially useful for image processing. A CNN is the best choice for change detection due to its feature extraction capability when it has sufficient training samples according to [76]. CNNs incorporate feature engineering in one framework, therefore manual feature extraction is not necessary. This is since they are able to extract features and create informative representations of time series automatically. By using advanced DL methods such as a U-Net and a conditional Generative Adversarial Network (cGAN), the speckle and other effects which cause reduced quality of the SAR amplitude data might be overcome while using limited to no temporal averaging.

1.3 GOAL OF THIS STUDY

The goal of this study is to develop a landslide detection and mapping algorithm, which can be generalized to worldwide application. The input data will be based on freely available S-1 SAR amplitude data and on publicly available software, such that this method is accessible to a broader audience. The necessity of temporal averaging of SAR amplitude images is researched by using both temporal averaged and single images. These images are trained in a DL method, by which the speckle and other effects which cause reduced quality of the SAR amplitude data might be overcome while using limited to no temporal averaging.

This leads to the following research questions of this study, namely:

- How do landslides appear in SAR amplitude imagery?
- How can landslides be represented by a combination of SAR amplitude imagery and other (satellite) data?
- How can landslides be detected from suitable SAR amplitude image representation ?
- How can success of landslide detection and mapping be evaluated?

Which lead to the main research question:
How can landslide detection and mapping be done on SAR amplitude satellite imagery?

1.3.1 Hypothesis

In this study the hypothesis is established in two statements: the null hypothesis and the alternative hypothesis. The null hypothesis states there is no effect, e.g. that landslide detection and mapping cannot be done by SAR amplitude imagery via the method proposed in this study. The alternative states that there is an effect, e.g. that landslide detection and mapping can be build on SAR amplitude imagery via one

of the methods proposed in this study.

In more details, the hypothesis is that by using a state-of-the-art DL method the speckle of SAR amplitude data can be overcome, such that landslides can be detected and segmented. In this way it can be seen to what degree denoising in the form of temporal averaging is necessary with respect to landslide detection. This hypothesis can either be rejected, e.g. there is insufficient evidence to suggest that the null is false. Or that the null is rejected and the alternative is found to be true.

1.3.2 Restrictions

The focus will be on landslides which are produced by large events such as extreme rainfall and earthquakes. The type of landslides and their sizes will not be researched due to time constraints. The software used in this paper will be restricted to Google Earth Engine (GEE)and Python. The satellite imagery will be restricted to Copernicus Sentinel 1 (S-1) Ground Range Detected (GRD) product, which is freely available in GEE. Therefore the use of InSAR and Polarimetric Synthetic Aperture Radar (PolSAR) will not be used, as this is less available and requires other software to process.

1.3.3 Content

This study will further entail a theory chapter where theory about the landslide databases, SAR satellite data, resolution and CML and DL models will be further explained. A methodology chapter will follow where the method and workflow will be stated. Subsequently a result chapter will follow, which will visualize and explain the results. Additionally, the discussion and recommendations following this results will be stated. Finally the conclusion will follow, in which the research question will be answered and the study will be concluded.
e

2 THEORY

This chapter gives background information and theory about multiple topics which are touched upon in this research. Firstly in section 2.1 various forms of resolution will be discussed. Then in section 2.2 Synthetic Aperture Radar (SAR) will be explained. Section 2.3 compares various machine learning methods of Conventional Machine Learning (CML) and Deep Learning (DL). Then a Convolutional Neural Network (CNN), a type of DL method, will be explained in section 2.4 and tuning of this network in section 2.5. Then the U-Net, a type of CNN, will be explained in section 2.6. Additionally the conditional Generative Adversarial Network (cGAN) will be explained in section 2.7. Class imbalance in a data set, and how that can be handled, will be discussed in section 2.8. Finally validation methods used in this study will be discussed in section 2.9.

2.1 RESOLUTION

The following four resolutions are the main types of resolution in the field of remote sensing. Satellites are designed with a specific resolution. There is a trade-off amongst these resolutions, as it nearly impossible to have a high resolution for each of the four types. When using satellite data, it is important to realize the resolution of that data and its consequence on the result of your study.

2.1.1 Spatial resolution

Spatial resolution is the size of the smallest feature that can be detected by a satellite sensor. For example the Copernicus Sentinel-1 (S-1) SAR satellite has a spatial resolution of approximately 20 meters [ESA]. This means that one pixel represents an area of 20 by 20 meters on the ground surface.

2.1.2 Spectral resolution

Spectral resolution is the ability to measure specific wavelengths of the electromagnetic spectrum. For example a satellite sensor which can measure eight different wavelengths has a higher spectral resolution than one which can measure three. Generally optical satellite sensors, also referred to as multi-spectral, have higher spectral resolution than radar satellites. This is since optical satellite sensor generally can measure multiple wavelengths of the electromagnetic spectrum while radar satellites can only measure a single microwave frequency.

2.1.3 Temporal resolution

Temporal resolution refers to the time between images. Specifically it is the time between images of the same geographical area or how often data of the same area is collected. This is also known as revisit time.

2.1.4 Radiometric resolution

Radiometric resolution refers to how finely that received radiance is divided amongst each band. The sensor can distinguish and record a greater range of intensities of radiation when it has a higher radiometric resolution. This resolution is typically expressed in bits for each band, it is possible that a satellite has a different radiometric resolution among its different bands.

2.2 SYNTHETIC APERTURE RADAR

Synthetic Aperture Radar (SAR) is an active image radar, meaning it emits microwaves and receives those microwaves after they are reflected by the Earth's surface. The radar synthesizes small antennas mounted on a platform, usually a satellite or an aircraft, to realize large virtual antennas. High-resolution radar images are generated thanks to these large antennas. Microwaves are able to penetrate clouds and do not need light, therefore it is possible to observe the Earth's surface regardless the presence/absence of sunlight and clouds [20].

Scattering happens when an emitted microwave from the antenna enters a conductor or dielectric, inducing a current an re-emitting the microwave from the induced current. If this induced current is re-emitted in the opposite direction of the emitted microwave, it is called back-scattering. This back-scattering can be read from the radar, since it is reaching the radar. Scattering in other directions cannot be measured by the radar. The scattering intensity depends on the frequency, and electric characteristics of the scatterer. Properties of the scatterer, e.g. a tree, a building, or a landslide, can therefore be measured by the scattering intensity. Urban areas generally have high scattering intensity, while vegetated areas have low scattering intensity. This happens since in trees and grass hardly any current is induced, while in cars and buildings there is material (e.g. metal) in which a current is easily induced. This process is visualized in figure 2.1.

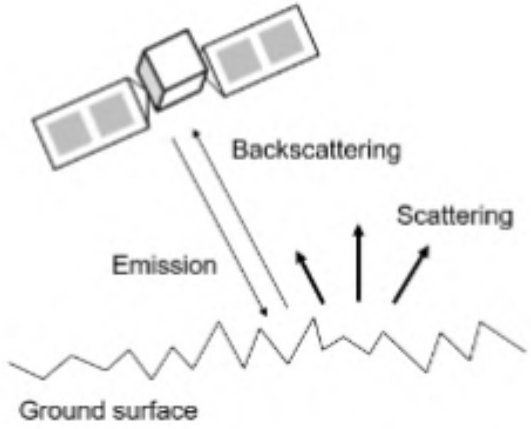

Figure 2.1: Microwave emission, scattering and back-scattering of SAR. Source: [58].

The SAR data used in this study is from the Copernicus Sentinel-1 (S-1) satellite constellation. The S-1 constellation currently consists of two satellites, S-1A and S-1B, launched in March 2014 and March 2016, respectively. The S-1 satellites carry a C-band radar sensor with wavelength of 5.6 cm. In comparison, there exist also L- and X-band radar sensors, with 24 cm and 3 cm wavelengths, respectively. Wavelength determines how the radar signal interacts with the surface and how far the

signal can penetrate into the surface. L-band radar penetrates better through dense vegetation than C-band, and thus generally produces better results in vegetated regions [86; 44].

Both satellites have a minimum 12-day revisit time for a given area. Using data from both satellites provides a minimum 6-day revisit time. However, the European Space Agency (ESA) does not acquire all S-1 images on every flight path and also does not always acquire both asc and desc data, therefore this revisit time is not guaranteed. [28].

The Sentinel-1 Ground Range Detected (GRD) product gives the calibrated backscattering coefficient σ_0. The sigma nought σ_0 is a normalized dimensionless measure for the intensity of a pixel. This sigma nought depends significantly on polarization, incidence angle and wavelength of the emitted microwave [23].

2.2.1 Look angle and direction

Look angle and look direction are the most important factors that affect the use of SAR imagery for change detection [2]. The direction of the SAR satellite may be either of descending or ascending order. This means the satellite is moving in an orbit with descending or ascending direction, which causes a certain look direction and angle. The nadir is directly beneath the platform. The radar beam is transmitted perpendicular to the flight direction along the swath. The swath runs from nadir (near range) to far range, and is the across-track dimension. The azimuth is the along-track dimension parallel to the flight direction, and therefore perpendicular to the swath. The look angle is the angle at which the radar 'looks' at the surface. The incidence angle is the angle between the radar beam and the ground surface. The incidence angle increases when moving from near to far range. This is visualized in figure 2.2.

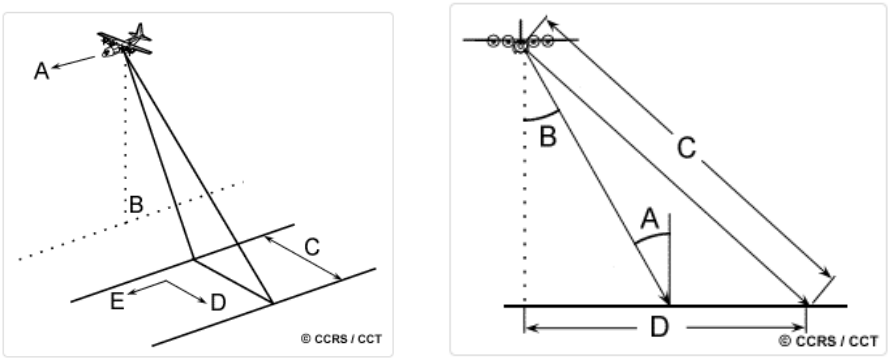

Figure 2.2: *left*: flight direction (A), nadir (B), swath (C), range(D), azimuth (E). *right*: incidence angle (A), look angle(B), slant range distance(C), ground range distance(D). Source: [4].

As previously mentioned, landslides may change the degree of slope and aspect of the topography of the affected area. This change can cause a change in the incidence angle of SAR. Therefore the imagery must be consistent in look angle and look direction in order to compare them and use them for change detection [78]. Combining ascending and descending geometry SAR data is a single stack improves landslide detection, compared to the use of ascending or descending images individually [28]. It also helps reduce bias introduced by the acquisition geometry, such as radar shadows, foreshortening and layer [1]. Furthermore, the combination results in a larger coverage and more images. Thus this might improve the temporal

resolution. Additionally it could be that slope and aspect changes cause change in imagery of ascending direction but not in those of descending direction, or vice versa. Since landslides occurring in aspects which are perpendicular to the SAR sensor look direction may not be detectable from images of that look direction [78].

2.2.2 Polarization

Polarization refers to the travel direction of an electromagnetic wave's tip. This can either be vertical (up and down), horizontal (from left to right) or circular (rotating in a constant plane, either left or right). This direction is defined by the orientation of the electric filed, which is perpendicular to the magnetic field [66]. This phenomena is visualized in figure 2.3

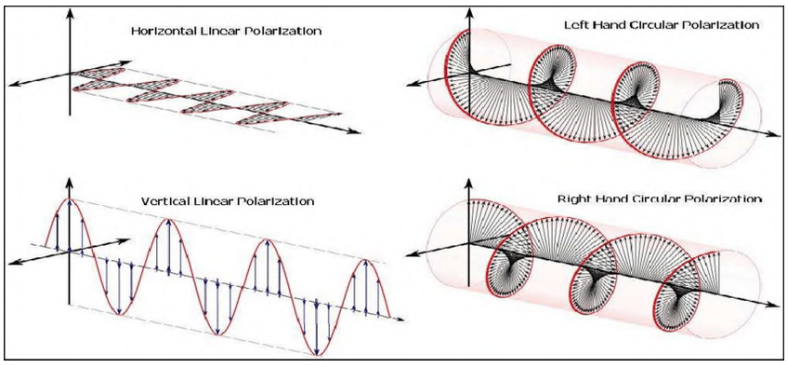

Figure 2.3: Schematic diagram of linear horizontal, linear vertical, right circular and left circularly polarized signals. The y-axis (vertical axis) is the electric field, the magnetic field is perpendicular to it on the x-axis (horizontal axis). Source: [79].

Radar antennas are designed to send and receive electromagnetic waves with a certain polarization. Several types of polarization configurations for radar antennas exist. The most simple is the single-polarization system, which transmits and receives a single polarization. Most often the transmitted and received microwave have the same polarization type, which results in either vertical transmit/ vertical receive vertical transmit / vertical receiver (VV) or horizontal transmit / horizontal receive horizontal transmit / horizontal receiver (HH). Another option is a dual-polarization system, which transmits one polarization but receives two. This then results in either vertical transmit / vertical and horizontal receive vertical transmit/horizontal receiver (VV + VH) or horizontal transmit / horizontal and vertical receive horizontal transmit/vertical receiver (HH + HV). The advantage of dual over single polarization is that dual gets additional detail about the surface through the different and complementary received signals of the varying polarized electromagnetic waves. Finally there is a quad-polarization system which alternates between transmitting horizontal and vertical waves and receives both, which results in VV,vertical transmit / horizontal receiver (VH), HH, horizontal transmit / vertical receiver (HV). In this system the pulse rate of the radar must be twice as high as with a single- or dual-polarization system, since the transmitter has to alternate between H and V for every pulse. This can cause interference between the received signals, which results in false/bad information. The solution for this phenomena is known as quasi-quad-polarization, which operates two dual-polarization (the VV + VH and HH + HV) and modes simultaneously. The two dual-polarization are operated each on a different frequency, such that there is no interference between them [20].

As mentioned before, the back-scattered signal depends on polarization. The different kinds of polarization each give different kind of information about the

subsurface. Cross-polarizations VH and HV are sensitive to forest biomass structure, which makes these useful to identify landslides in vegetated areas [28].

2.2.3 Speckle

Speckle reduced the quality of a SAR images and makes image interpretation difficult. SAR is a coherent imaging system, meaning that scatters are added coherently from a corresponding resolution cell. These scatters may interfere with each other, either constructively or destructively, depending on their phase [87; 23]. This interference causes bright and dark pixels, which is referred to as speckle. Speckle is generally referred to as (speckle) noise, but it is not. The measurement of the signal does not cause these effects, which would be noise, but the effect of the measurement technique causes the speckle. An example of speckle can be seen in figure 2.4. The 'clean' version of the images and their speckled SAR counterpart are shown, which shows the difficulty to interpret the speckled SAR images and how speckle decreases their quality.

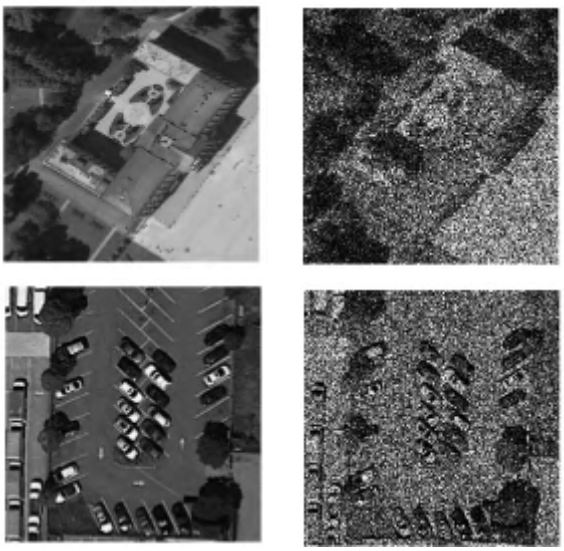

Figure 2.4: Images of a parking lot and a palace (left) and the speckled SAR versions of those (right). Images obtained from [27].

2.2.4 Amplitude

The change in land cover from dense forest to landslide scars and deposits causes changes in surface reflectance and hill slope geometry that is detectable by measuring the change in SAR amplitude. The most commonly used amplitude-based change detection method [36] is

$$A_{ratio} = log_{10}(\frac{A_{pre}}{A_{post}}) \tag{2.1}$$

Where A_{pre} is the amplitude pre-event and A_{post} is the amplitude post-event. When these amplitudes are taken in decibels, the equation can be rewritten to

$$A_{ratio} = A_{pre} - A_{post} \tag{2.2}$$

Positive A_{ratio} generally corresponds to a decrease in SAR amplitude after a landslide event. There can be some negative amplitude changes, aka a negative A_{ratio},

within landslide scars [28; 36; 1].

SAR amplitude change can also result in false positives, since they are not always related to landslides [28]. For example, amplitude change can happen due to change in water content of a farmland, as a result of flooding. Such false positives can be removed by applying a DEM mask to remove areas where landslides are unlikely to occur [28]. [28] demonstrated that rainfall-triggered landslides in the Hiroshima Prefecture cause an overall decrease in SAR amplitude. This decrease in SAR amplitude occurs because the landslide scar and damage acts to decrease back-scattering reflectance to the satellite relative to a pre-failure ground surface [1].

2.2.5 Interferometric Synthetic Aperture Radar

Interferometric Synthetic Aperture Radar (InSAR) relies on the the phase difference between two or more SAR images. In this case the SAR images need to include not only the amplitude, but also the phase. The SAR images may be acquired at different times and/or different positions. The phase difference gives, after some processing, the surface displacement. This phase difference must be done over coherent (stable) pixels over time. Therefore it is difficult to find coherent pixels over vegetated areas, as opposed to mountainous or urbanized areas [38]. This happens because vegetation changes a lot with respect to mountains or e.g. buildings, making the pixels incoherent over time. Besides this difficulty, the processing of InSAR is extensive and time-consuming. In this thesis this will not be explained or explored further, although for example more information can be found in [83; 40; 66].

2.2.6 Amplitude vs. coherence based

Coherent based methods use interferometric phase information, in contrast to amplitude-based methods which do not [36]. InSAR is a type of coherence based methods, since it uses phase information. Amplitude-based change detection outperforms coherence-based change detection in densely vegetated mountainous areas, where landslides tend to occur, since coherence is always low there while amplitude may change [36]. Still many, e.g. [12; 36; 13] use InSAR to detect landslides. [36] compared several detection methods of SAR and found that coherence based works best is most situations. Still [13] showed that these coherence based methods were not able to map individual landslides. In short, experts are not unified on which method is best. This also depends on the type of problem. In this study, the deformation rate of the landslide is not of interest so InSAR is not required. Furthermore as previously discussed in section 1.3.2 the InSAR product is not available in GEE.

2.3 MACHINE LEARNING METHODS

Conventional Machine Learning (CML) methods are used for classification and/or segmentation of objects. In landslide detection these methods can be divided into object-based or pixel-based methods. Values from certain features are sampled into pixels or object and used as input feature vectors for training CML methods. In Deep Learning (DL) methods the derived feature maps are directly used as an image for training [68]. Therefore DL methods require less human supervision than CML methods [24].

In pixel-based methods, the features are sampled over a grid set of points. The features are treated as a raster and co-registered and re-sampled to a common distribution. Then it is computationally efficient to do a per-pixel analysis. Geometric

and contextual information in the image is then ignored [80; 29], thereby the detection of the extent of a landslide is difficult [68]. Change detection can be done with pixel-based methods, but they require a time-series or multi-temporal images [68]. In object-based methods the area of interest is segmented into so-called objects: a group of meaningful homogeneous non-overlapping regions. The approach assumes that a pixel is likely to belong to the same class as its neighboring pixels [63].

Unlike pixel-based and object-based methods, a Convolutional Neural Network CNN can learn directly from images, which removes the need for sampling information in the form of feature vectors (either pixels or objects) [68]. With CML methods efficient features must be produced and selected beforehand, such as spectral derivatives from certain spectral bands, spatial features or geophysical layers. It is tricky to derive these features and also to test which ones are efficient. In contrast, DL methods automatically extract the most efficient features by exploiting the convolutional layers [69]. In that sense, DL can do feature learning in a unsupervised or semi-supervised manner.

The performance of CML versus DL depends per study, since there are many different CML and DL methods and many different scenarios. [24] found that CNN does not outperform CML methods like Artificial Neural Network (ANN), Support Vector Machine (SVM) and Random Forest (RF), only in ideal cases. However, generally DL methods are found to be better working than CML methods. [68] found that the U-Net (a type of CNN) structure outperformed both the pixel- and object-based CML methods. [44] found that the CNNs based on U-Net method were more effective for landslide detection suing SAR data than a pixel-based CML method.

2.3.1 Semantic segmentation

Semantic segmentation is a pixel-based class of classification methods, where each pixel is labeled with the corresponding class. This is often confused with object detection and instance segmentation [6; 54; 61]. Object detection is where each object in an image is detected and typically the result is given as a bounding box around the object. However, the approximated extent of the object is not given, as in the case with semantic segmentation. Instance segmentation gives the extent of the object like semantic segmentation, it does so for each individual object separately thereby detecting each object separately. The latter is not the case for semantic segmentation, where objects of the same class are classified to the same single class. This is best explained by a visualization, as done in figure 2.5.

Figure 2.5: left to right; object detection, semantic segmentation, instance segmentation. Source: [Gkioxari et al.].

2.4 CONVOLUTIONAL NEURAL NETWORK

A Convolutional Neural Network (CNN) tries to replicate how a human perceives information from an image by learning from a large collection of labeled examples [68]. A CNN is efficient for recognition of complex image patterns and semantic clas-

sifications. The CNN transforms pixels values of images into feature values using multiple processing layers. These layers consist of neurons, which refers to the 'neural' part in CNN. Neurons are named after biological neurons, on which the idea of neurons from a CNN are based. Basically the neuron gets an input from other neurons, processes this and gives an output. All neurons in one layer have the same bias. Through training they will receive (different) weights, which will change the output. Neurons only have weights and bias and therefore are linear. To introduce non-linearity, which is necessary since many applications of CNNs are non-linear, activation functions are used. The activation layer goes over the entire layer, processes all (weights of) neurons and gives an output. Max pooling, batch normalization and dropout are used to keep the speed, structure and performance of the neural network optimal. Finally, loss functions are used to define what weights should be given to which neuron, based on the quality of the output. These terms will be explained further along in this section.

A neural network can generally have three types of layers: input layers that take raw input from the domain, hidden layers that take input from another layer and pass output to another layer, and output layers that make a prediction. Generally the deeper the network, the more the network will learn and the better it becomes [53]. "Deeper" in this context means with more convolutional layers. However, there is a certain tipping point, which differs per application, where more layers make the result less accurate/worse. This is a type of over-fitting, where a model is too closely fitted on a limited amount of data. A CNN is subject to many hyper-parameters, which need to be tuned while training to obtain an optimal result. These terms will also be explained furhter along in this section.

2.4.1 Convolutional layers and max pooling

A convolution layer executes the filtering process on the input image. Convolution preserves the spatial relationship between pixels, while learning image features. A pooling layer is placed almost immediately after the convolution layer. The pooling layer reduces the dimensions of each feature map, but retains the most important information [61]. This can be done by taking an average, sum or maximum function over a spatial neighborhood of pixels. The choice of function is not arbitrary, still the most used one is the maximum function. An example of max pooling with a 2*2 window is shown in figure 2.6, which means the 4*4 pixels are down-sampled to a 1*1 pixel matrix.

The pooling layer has several advantages. It makes the input representation (feature dimensions) smaller and therefore more manageable. Additionally it reduces the number of parameters and computations in the network, thereby controlling over-fitting. The CNN becomes invariant to small transformations, such as translations or distortions in the input image. This is since the maximum/average value is taken in a local neighborhood, thereby reducing the influence of said distortions. Lastly, This pooling layer decreases the position sensitivity in the features, i.e. objects can be detected in an image no matter where they are located. The final extracted features are integrated and recognized by the fully connected layer(s) [45].

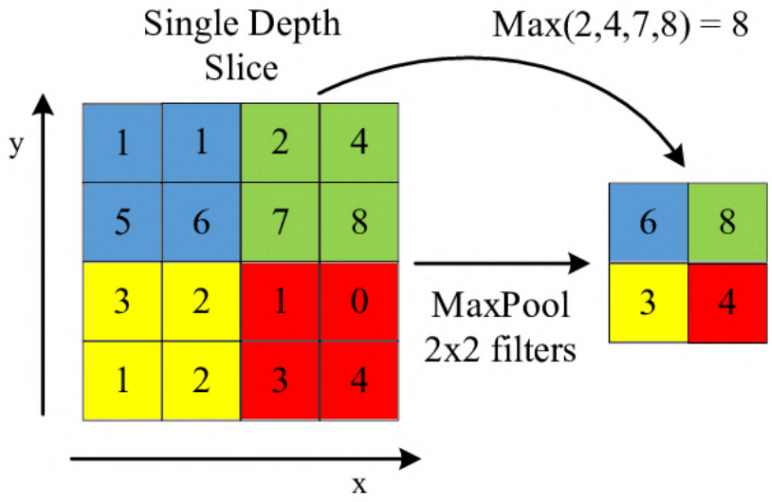

Figure 2.6: Max pooling with a 2*2 window. Source: [39].

2.4.2 Dropout

The term "dropout" refers to dropping out units (both hidden and visible) in a neural network. Dropout refers to ignoring a randomly selected set of neurons during each training stage. This is done to prevent over-fitting, since neurons are codependent amongst each other. Due to this co-dependency, the individual information of each neuron is decreased. By ignoring a different randomly selected set of neurons during each training step, this codependency is decreased and thereby over-fitting as well [77; 22]. The number of iterations of the CNN increases with dropout, but the training time per epoch decreases.

2.4.3 Batch normalization

Batch normalization is a technique which normalizes the activation of each layer in the network. Normalization is subtracting the mean value of a batch from each layer and dividing by the batch standard deviation [77]. Standardizing the activation of the prior layer means that assumptions the subsequent layer makes about the spread and distribution of inputs during the weight update will not change, at least not dramatically. This makes the training more stable and helps coordinating the updates of the model, as such the training time is reduced [22].

2.4.4 Activation layers

An activation layer introduces non-linearity. Convolution is a linear operation, since it uses element-wise matrix multiplication and addition. Non-linearity is therefore not included and is therefore added by an activation layer. Therefore an activation function must be applied at least once in the CNN [61]. The type(s) and amount of activation functions has a large impact and differs per CNN.

As mentioned before, CNN contain three type of layers: input, hidden and output layers. An activation function is used in each layer, including the hidden layers. After or within the processing of each node in the network an activation function is used. Networks often contain the same activation function for all nodes in a layer. For the hidden layers a differentiable non-linear activation function is used, such that the model can learn complex, (non-)linear functions. The activation function of the input and output layer may differ. The activation function of the output layer

depends on the type of prediction required by the model. For example, a regression problem requires at least a linear activation function at the output layer.

There are many possible activation function, in the following list (some of) the most popular ones are listed. The first three are most often used for hidden layers, since they are differentiable and non-linear.

- *The Rectified Linear Unit Rectified Linear Unit (ReLU)*. It is a non-linear operation, which outputs zero if the input is negative. Positive input is left as is. [75].
- *The Leaky ReLu*. This is an 'optimized' ReLU function, where the output is not zero when the input is negative. Instead, the slope is changed to y = 0.01x when x is negative (where y represents the output and x the input value). This solves the 'dying ReLU problem', which will not be further explained in this study. [75].
- *The sigmoid activation function*, in mathematical terms:

$$sigmoid(x) = \frac{1}{(1 + e^{(-x)})} \tag{2.3}$$

  Small values, which depends on your data, are returned as a value close to zero, while large values are returned as a value close to 1. [75].
- *The hyperbolic tangent activation function*, also referred to as tanh, is in mathematical terms

$$\frac{e^{x} \check{} e^{-x}}{e^{x} + e^{-x}} \tag{2.4}$$

  The function takes any real value as input and outputs values in the range -1 to 1. Large values (more positive) will be returned with values close to 1.0, whereas smaller values (more negative) will be returned as values close to -1.0. [75].
- *The softmax activation function* converts a vector of values to a probability distribution. In mathematical terms it is described as

$$\frac{e^{x}}{\sum e^{x}} \tag{2.5}$$

  , where x is the vector of values .The elements of the output vector are in range (0, 1) and sum to 1. Softmax is often used as the activation for the last layer of a classification network because the result could be interpreted as a probability distribution. [75].

2.4.5 Loss functions

Loss functions evaluate how well the DL model models the data. If predictions deviate a lot from the true value, the loss function will be a large number and vice versa. In combination with the optimization functions, the loss function learns to reduce the error in the prediction. There are many possible loss functions and it depends on several factors, including on the type of model which is most suitable [85]. In this section some of the most common loss functions for classification/segmentation problems will be discussed.

- *Binary Cross-Entropy Loss*. Binary cross-entropy is the default loss function to use for binary classification problems. The mathematical equation is:

$$BCE(p, y) = \begin{cases} -log(p) & \text{if } y = 1 \\ -log(1 - p) & \text{otherwise} \end{cases} \tag{2.6}$$

where p is the predicted probability and y is the label with value 1 or 0. In this equation the p value is for label 1. If p is 0.8, then the predicted label has 0.8 probability of begin 1 and 0.2 probability of being 0. For label 0 this would be vice-versa. This can be rewritten to a simpler equation which covers both cases, if

$$p_t = \begin{cases} p & \text{if } y = 1 \\ 1 - p & \text{otherwise} \end{cases} \tag{2.7}$$

, then,

$$BCE(p, y) = BCE(p_t) = -log(p_t) \tag{2.8}$$

here p and 1-p are replaced by p_t. If label y = 1 then p_t is p, a larger p_t corresponds to larger p as well, so nearer to label 1. If label y = 0 then p_t is 1-p, such that a bigger p_t corresponds to a smaller p, so nearer to label 0. Thus the larger p_t, the closer the prediction is to the true value/label, so the more accurate the prediction is.

The binary cross entropy loss is relatively simple, which makes it smooth and fast to train. It considers all pixels equally when calculating the loss, therefore it is not well suited for class imbalanced data sets. This can be potentially solved by weighted cross entropy, where the minority class' pixels receive more weight than those of the majority class. The optimal value of this weight is difficult to find [35].

- *Dice (coefficient) loss*. The dice coefficient is a measure of overlap of the predicted mask and the ground truth, as will be explained in detail later in this chapter (2.9.2). The dice coefficient is a metric which is used specifically for segmentation problems. Since it does not account for the background class, it cannot dominate over the smaller segmentation class. The dice coefficient outputs a score in the range [0,1] where 1 is a perfect overlap. It works well with class imbalanced data, but it has a potential for gradient explosion and is slow in training [35].

- *Focal loss* Focal loss was introduced by [51], which sought after to improve binary cross entropy with regards to class imbalanced data. This is done by trying to decrease the total loss according to the feedback of each pixel. Then the loss of well trained pixels has large decrease , while losses of poorly trained pixels have little decrease. In this way class imbalance is taken into account, and the model focuses on training pixels which have not been well trained yet. The mathematical equation for focal loss is :

$$F(p_t) = -\alpha_t (1 - p_t)^{\gamma} log(p_t) \tag{2.9}$$

$(1 - p_t)$ is used to decrease the cross entropy loss, $log(p_t)$. If pt gets bigger and is close to 1, 1-pt gets smaller and is close to 0, thus the original cross entropy loss is largely decreased. If pt gets smaller and is close to 0, 1-pt gets larger and is close to 1, thus the original cross entropy loss is trivially decreased. [51] put α to 1 and tested different γ values. A higher γ results in a lower training loss, and vice versa.

- *Focal dice loss* Focal dice loss is the combination of focal loss and dice loss, and is therefore well used on segmentation problems which have class imbalanced data. This loss was created and shared via GitHub by Yakubovskiy [84].

- *Combination of binary cross entropy and dice loss* This combination is better for segmentation problems, while avoiding the potential of gradient explosion by mixing the dice loss with binary cross entropy. This loss was created and shared via GitHub by Yakubovskiy [84].

2.5 TUNING OF A CONVOLUTIONAL NEURAL NETWORK

It is important to tune a CNN model to the data to get the most optimal result. The model should fit the data well, and to that extent certain choices must be made and tested. This next section will explain the theory behind this tuning and how tuning is done.

2.5.1 Hyper-parameters

Hyper-parameters in a CNN are the variables which determine the network structure and how it is trained. Hyper-parameters are set before training, but can be tuned/optimized while training. A CNN has multiple hyper-parameters such as number of training epochs, learning rate, weight decay, dropout ratios [50]. Also the network architecture itself introduces hyper-parameters (depth, number of units, size of convolutional kernels, etc.).

Ideally each hyper-parameter would have a separate validation set, with which it can be tuned/the most suitable value can be found. Once all hyper-parameters are fixed in this way, the various validation sets can be merged into the final training set. By using the same validation set to tune several hyper-parameters, the estimate (of the hyper-parameters) will become biased. However, by using separate validation sets for a number of hyper-parameters, the remaining amount of training data will be diminished. Since a CNN needs many training data, there is a chance that the CNN will then become under-fitted. Besides the reduction of training data, the tuning of hyper-parameters may also take a long time and have high computational cost. Therefore this is resolved often by copying popular neural networks and then using an educated guess.

A training sample is a single row of data, in this case one source and target image pair. Batch size refers to the amount of samples to work through before the internal parameters are updated. The batch size can be equal to the total amount of training samples, or smaller than that. When the batch size is smaller than the amount of training samples, then it is called a mini-batch. One epoch is when each sample in the total training data-set has been worked through, i.e. each sample has contributed to the update of the internal parameters. When the batch size is equal to the entire training data-set, it is therefore by definition also equal to one epoch. With use of mini-batches, one epoch contains multiple mini-batches. The amount of epochs is thus the amount of times the CNN will work through the entire training data-set. Too little epochs may lead to under-fitting, while too many leads to over-fitting. This depends per problem and CNN, the amount of epochs in literature varies from 100 to 10000 (and more). Therefore it is common to create a plot with on the x-axis the amount of epochs, and on the y-axis the error of the model. In this way the most suitable amount of epochs can be chosen.

2.5.2 Well generalized, over- and under-fitting

Generalization in machine learning refers to a models ability to adapt to new, unseen data. A model is well generalized when it can make accurate predictions on new data. Under-fitting in machine learning refers to the situation when a model cannot capture the underlying trend of the data. Under-fitting can be avoided (amongst others) by using more data or by using a more complicated/better fitting model. Over-fitting happens when 'too much' data is used for training. The model then starts learning from the noise and inaccurate data of the training data. Consequently the model does not categorize the data correctly, because of too many details and noise. Another way to describe over-fitting is when the model performs well on training data, but poorly on test data [77; 37]. The causes of over-fitting are the non-parametric and non-linear methods because these types of machine learning algorithms have more freedom in building the model based on the data-set and therefore they can really build unrealistic models. A solution to avoid over-fitting is using a more simple model which fits the data better, or regularization (see Section 2.4.3,Section 2.4.2). For example, a linear algorithm should be used for linear data.

Bias and variance

In machine learning the terms bias and variance are often used, and they differ from their respective definitions in statistics. Bias means here the assumptions made by a model to make a function easier to learn. Variance refers to the process where a model is trained on training data and a very low error is obtained, but upon changing the data and then training the same previous model a high error is obtained. An under-fitted model has high bias an low variance, whereas an over-fitted model has low bias and high variance [77].

2.5.3 Learning curve

A learning curve is a line plot, with the learning metric on the y-axis and the duration on the x-axis. A learning curve can be used to see if and when the model has an under, over or good fit [8]. This 'when' is also important, since it gives an indication of when to stop training the model/ at which epoch the training of the model should be stopped. The metric which is used varies per model, mostly accuracy or loss are used. For loss, the loss error should minimize over the training epochs, where a value of 0.0 means the model has learned perfectly. The learning curve can be done over the training data-set, which indicates how well the models is learning. When the learning curve is taken over the validation data-set, it indicates how well the models is generalizing. Generalizing meaning in this context, how well the model can predict data on which it is not trained/which is unseen.

For typical CNNs, including U-Net, a good fit can be derived from a learning curve displaying the loss of the training and validation set. Moreover the most suitable epoch, e.g. where the model has a good fit but before it turns into an over-fit, can be derived from this curve. This good fit in this learning curve is defined as when the loss of the training and validation set both decrease to a point of stability, with a small gap between these two loss values. This gap will almost never be zero, since the model will almost always have a lower loss on the training data-set than on the validation data-set. This gap is called the 'generalization' gap.

The learning curve can also show a convergence fail, when the CNN fails to converge. This happens with under-fitting, when no converge is found. An example of an under fit, a good fit and an over-fit model in a learning curve are visualized in figures 2.7,2.8, 2.9 respectively.

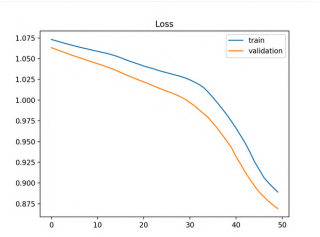

Figure 2.7: Learning curve showing under fit. Source: [8].

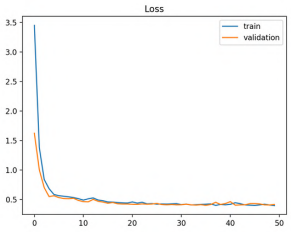

Figure 2.8: Learning curve showing good fit. Source: [8].

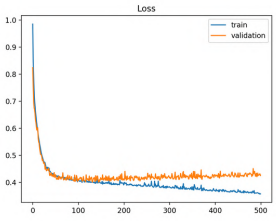

Figure 2.9: Learning curve showing over fit. Source: [8].

2.6 U-NET

The U-Net model, shown in figure 2.10 was proposed by [71]. The name originates from the shape of the network, which is symmetrically shaped as a U. It was originally applied to medical imagery, but currently applied to many other semantic segmentation problems. The left half contains a compression channel, whereas the right half contains an expansion channel. The compression channel is a typical CNN, with convolutional and max pooling layers. This left side extracts the 'important' features and reduces the size of the image. "Important" means the information which best describes the image in this context. The expansion channel performs a de-convolution operation to halve the dimension of the feature map, and then stitches the feature maps corresponding to the compression channel together. In this way it reconstruct a feature map with double the size [53]. The U-Net model combines this encoding-decoding structure, i.e. the compression on the left and the extension on the right, with a jumping network. This jumping network consists of skip connections, which connects each corresponding level of encoding and decoding layers. Currently skip connections are standard in many CNNs. A skip connection skips some layer in the network and feeds the output of one layer as the input to the next layer(s), instead of only the next one. This is beneficial for the model convergence [71].

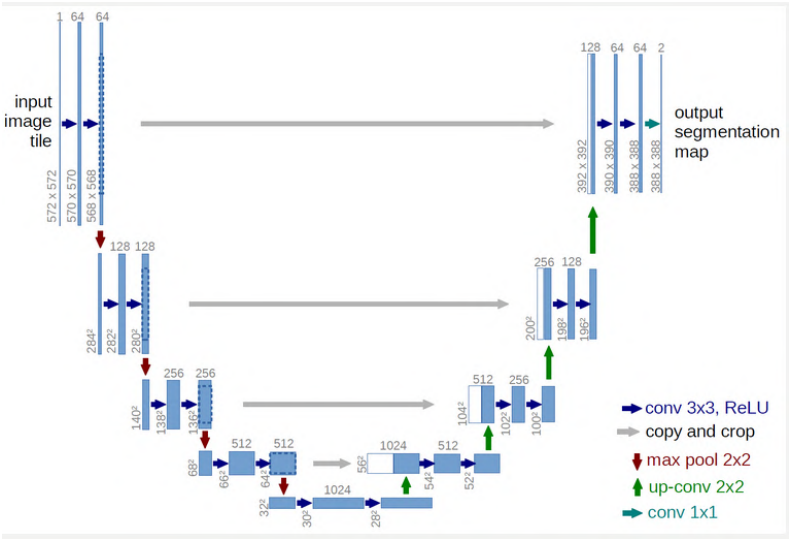

Figure 2.10: U-Net architecture as described by [71].

The U-Net is a segmentation model, which segments one or more classes into objects or polygons. This is visualized in a scheme in figure 2.11.

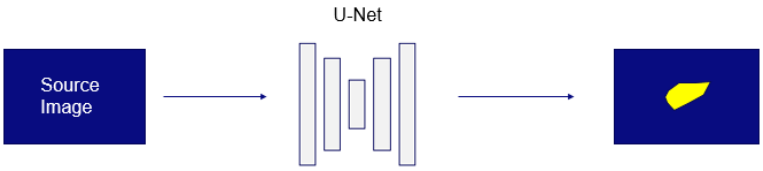

Figure 2.11: Scheme of the U-Net [71]. The source image is used as input, the output is a masked image. The mask gives the segmented landslide.

2.6.1 U-Net for landslide segmentation

Semantic segmentation based deep-learning architecture (like U-Net) is expected to outperform a sliding-window CNN for detection of landslides [68]. The U-Net model has a simple and effective structure to extract target features by using a small number of samples [53]. Many papers have used U-Net on landslide detection, e.g. [69; 53; 68; 44]. U-Net has been widely used for landslide segmentation using optical satellite imagery [69; 53]. U-Net has also been used on SAR satellite imagery by [44], where it has a dice coefficient of 0.8267 and 0.6378 for the train and test data set. The resulting segmented landslides by the U-Net model of [53] are shown in figure 2.12

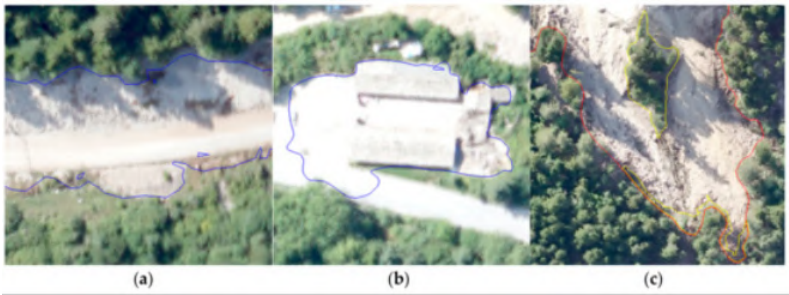

Figure 2.12: Results from landslide segmentation using a U-Net by [53]. a) and b) show incorrect extractions, c) is an correct extraction. The input images where optical images with red,green,blue and digital surface model, slope and aspect band. The overall accuracy of the method was 0.93.

2.7 GENERATIVE ADVERSARIAL NETWORKS

The original generative adversarial network GAN framework was proposed by [74]. The generative part refers to the generator G, which aims to capture the data distribution. The adversarial part refers to the adversarial process in which the generator G is trained simultaneously with another model. This model is the discriminator D, which aims to estimate the probability whether a sample came from the training data (true) or from a generated (fake) sample from the generator [74; 59]. In other words, the goal of the generator is to fool the discriminator, while the goal of the discriminator is to detect fake generated data.

Conditional Generative Adversarial Network

GANs can be extended to a conditional model, namely conditional generative adversarial network (cGAN), as first done in [60]. The conditioning factor can be imple-

mented via either G, D or on some extra information y. This data y could be class labels for example [59]. This cGAN is visualized in figure 2.13.

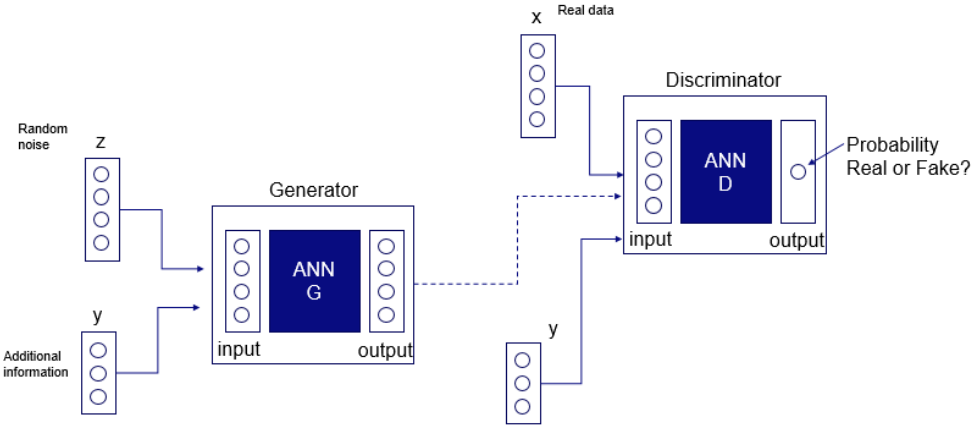

Figure 2.13: Scheme of a conditional Generative Adversarial Network. The generator receives additional information y and/or random noise z. This than generates real and fake images and gives these to the discriminator. The discriminator also receives information x, which is a label referring to whether an image is real or fake. The output of the discriminator is whether the received images are real or fake.

2.7.1 Pix2Pix conditional Generative Adversarial Network

Pix2Pix cGAN [34] is an image to image translation method. Meaning that the input image is translated to an output image. The model described by [34] was further explained in [9] and the latter is used in this study, although some alternations were made.

Discriminator

The discriminator is a CNN which performs conditional image classification. It takes a source and target image (pair) as input. Some of these pairs are real, while others are generated by the generator (see Section 2.7.1). The discriminator predicts the likelihood of the target image to be a real or false image translation of the source image. This is done via a so called patchGAN model, which maps each output prediction of the model to a 70 by 70 square patch of the input image. This size of the patch is recommended by the authors of the Pix2Pix model after extensive research. This patchGAN gives the advantage that any size of input image can be applied to the discriminator. Hence the output of the discriminator is one value or a map of values (depending on the size of the input image) where each value is a probability of likelihood that a patch in the image is real or fake. The discriminator is optimized during training using binary cross entropy. In order to slow down the changes of the discriminator relative to those of the generator during training, the optimized weights have half the effect of the previous optimized weights.

Generator

The generator has a U-Net structure, as previously described in Section 2.4. The input is a source image, while the output is a target image. In this way both source and target images are generated, which are fed to the discriminator. The generator is trained via the discriminator, since this gives back whether the generated images are good or not. The generator wants to be able to fool the discriminator, by generating images which are false but which the discriminator believes to be real. In this

way the generated images from the generator become better. The generator is also updated by the loss/error between the generated image and the target image.

Composite model

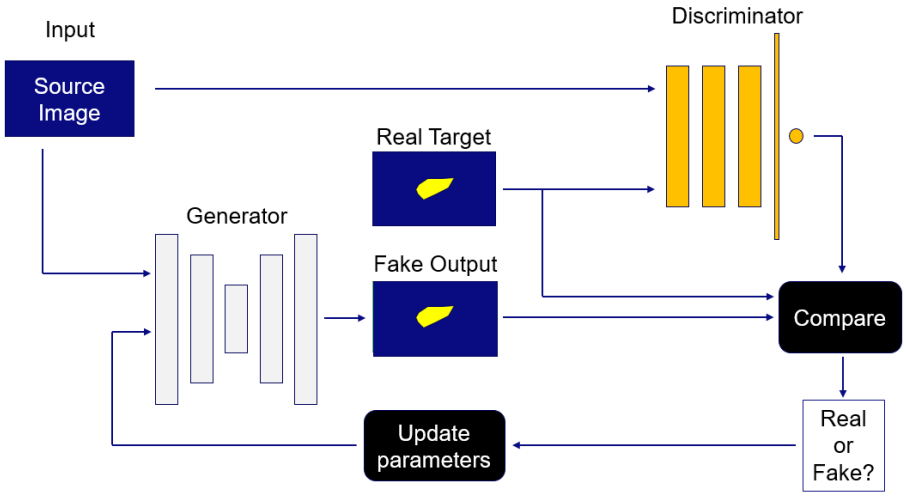

Figure 2.14: Scheme of the composite model, with a source image as input which is translated to an output image. The output image is generated by the generator, both a fake and a real output image is made. Then the discriminator has to say which is true and which is fake. Through this process both generator and discriminator learn. The goal is to have a generator which generates the accurate output corresponding to the input image.

The composite model combines the generator and discriminator in one model. This composite models is visualized schematically in figure 3.22.The discriminator is updated in a standalone manner, so the weights are reused in this composite model but are marked as not trainable. The composite model updates by 2 loss functions. One indicates whether the generated images were real, which motivates the generator to produce more real images. The other is the real translation of the source image, e.g. the target image, which is compared against the output of the generator model (the generated target image). The composite model with the input of the RGB image is visualized in figure 3.22.

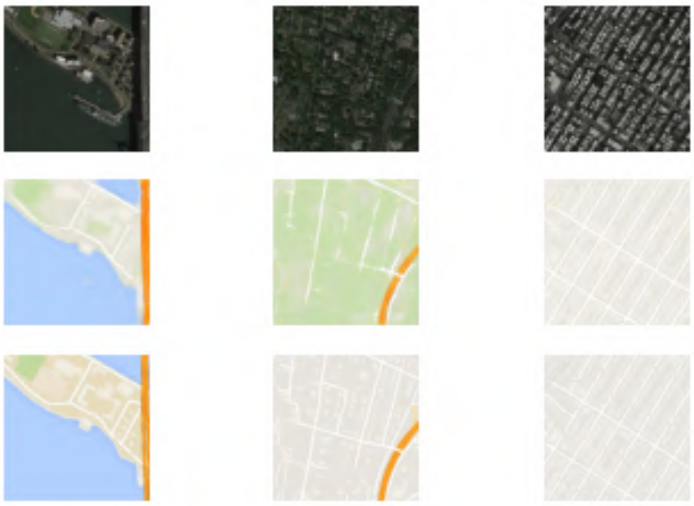

Figure 2.15: Plot of three source images, generated images and true target images on the first, second and last row respectively.

Figure 2.15 shows the results of the pix2pix cGAN image translation model on its original data. The data set consists of source images depicting areal photos and target images depicting the corresponding 'maps' of these photos. As can be seen, then model is quite well in translating the source image to a target image which is very similar to the true target image. This is the aim of the model, and it is successful in that.

2.7.2 Generative Adversarial Networks versus Convolutional Neural Networks

Most semantic segmentation of satellite imagery, such as landslide mapping, is done in a fully supervised CNN. This type of network is mostly applied to a certain small region, since the generalization of these CNNs on such problems is poor. This is due to the diversity of landscape types and seasons of landslide prone regions. The satellite data is limited and image resolution might not be optimal. Finally, labeled data-sets are scarce among different geographies and seasons [46]. Imbalanced samples are common, since generally more normal samples than anomalous samples are available. This limits CNNs in their ability to do anomaly detection, even with the help of data augmentation. GANs can produce images similar to the training data, thereby overcoming this imbalance [81; 73].

GAN is unsupervised, contrary to most CNN methods, and does not need labeled data. cGAN are superior to CNN in image-to-image translation tasks such as image coloring, dense labeling and noise reduction [46]. [46] shows that their cGAN outperforms a U-Net on semantic segmentation of land cover classes on 5 out of 6 classes of the test set. There the performance benefits of cGAN versus CNN on unseen data are demonstrated.

The main drawback of GAN is the difficulty to train it, as the balance between the discriminator and generator is crucial to be maintained [55]. Additional to this, the architecture and hyper-parameters of the model are crucial. GAN is implicitly models the data distribution, hence for anomaly detection it needs a costly optimization

procedure. This is impractical for large data-sets or real-time applications [88].

2.7.3 Learning curve of Generative Adversarial Networks

The learning curves were explained in 2.5.3. For a U-Net model the learning curve will look like the typical curve as stated in 2.7,2.8, 2.9 depending on the fit of the model to the data. This makes it relatively easy for a Unet, as opposed to a GAN, to see how good the fit of the model is and when to stop training. For GANs this is more complicated, since the loss functions do not converge as in the CNN. This is due to the nature of the GAN. Just as the discriminator and generator are competing with each other when training the model, their loss functions do the same. This means that instead of converging, the loss functions are in a equilibrium. It is still possible to get a learning curve of these loss functions, which gives an indication if the training of the model is going well. Also, considering the training of GANs is unstable, GAN failure can be deduced from this learning curve. It is not possible, or difficult, to define the epoch at which to stop training by the learning curve, as is possible with a typical CNN.

Figure 2.16 shows an example of normal/expected loss of a cGAN during training. It is typical for the generator and discriminator loss to start off erratic, while later they 'converge' to a stable equilibrium. The discriminator has two loss functions, namely one for the true examples and one for fake examples. Ideally, discriminator loss for real and fake samples is about the same at or around 0.5, and loss for the generator is slightly higher between 0.5 and 2.0. Figure 2.16 shows that the losses start of erratic, but stabilize around epoch 100. So from there onward, there is a possibly good fit. The best epoch should then be found with other metrics.

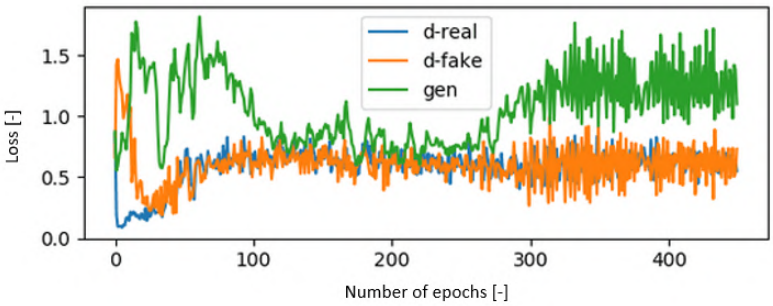

Figure 2.16: Learning curve of a GAN model showing a normal/expected loss. The "d-real" and "d-fake" plotted lines describe the discriminator losses with regards to real and fake targets respectively. The "gen" plotted line describes the generator loss. Source : [10].

A failure for a GAN to converge refers to not finding an equilibrium between the discriminator and the generator. This failure typically happens with the discriminator loss going (close) to zero. The cause of this failure is often the generator generating bad images which the discriminator can easily identify. Therefore the discriminator is not learning and its loss is (close to) zero. It is possible that this happens in the beginning of training, and that the model will recover after some batches or epochs.

Another failure type is mode failure, where the generator can only generate one or a relatively little amount of outcomes, e.g. modes. The generator model generates identical output images for different source image, so the output is always the same image or nearly the same image. This can failure can be reorganized from

the learning curve by the generator loss, which will oscillate, but not decrease over time. The discriminator often also shows small oscillations.

2.7.4 Generative Adversarial Networks for landslide segmentation

GANs have not been used, to current knowledge, on landslides yet. However, [57] used a cGAN on a segmentation problem, where several landcover classes such as urbanized and farmland were segmented based on high resolution SAR imagery. The overall dice score was 84.07, which was higher than other CNN and U-Net tested in the study [57].

2.8 CLASS IMBALANCE

In a segmentation problem there are multiple classes to be identified. Class imbalance is defined as the situation where the classes are not represented equally in the training images. This can lead to biased predictions.

There are several approaches to deal with class imbalance. There are several which tend to transform an imbalanced data-set to a balanced data-set, through data augmentation and/or resampling. One type of re-sampling is under-sampling where data from the majority class is removed such that the amount of data from the minority and majority class is the same. The big disadvantage of this approach is that the amount of training data is diminished. Another re-sampling method is over-sampling where data of the minority class is duplicated and added to the training data, such to increase the amount of data in the minority class to the amount of the majority class [77].

Another more complicated approach is to alter the DL method itself. As a result the imbalanced data-set itself does not have to be changed, but the model is accommodated to it. One way to alter the model is to use a specific loss function. The loss function is a mathematical way of defining how far off a predicted value is from the real value of a data point. Focal loss is one of the possible, common loss function which can solve the class imbalance problem. Instead of trying to reduce outliers or predictions where the model's prediction is far off from the truth, Focal loss reduces the weight (or impact) the values it predicted correctly carry.

2.8.1 Data augmentation

Data enhancement is done to increase the variety of the training images. This then makes the model stronger and enables its generalization, since it is trained on more diverse images [77]. For example, it can recognize both images in their original shape and when they are transposed, rotated, or flipped. Generally a large number of training samples is needed for a DL structure. For deep-learning data-sets, data enhancement, such as rotating and flipping, is usually applied on the original images. This must not be done excessively, since it can cause over-fitting of the model. [53] found that the most suitable data enhancement for their model was only 90 ° rotating, 270 ° rotating and horizontal flipping. Most common data augmentation are; rotation, flipping, zooming, cutting. It depends on the aim of the model and the type of data if any/which augmentation(s) is/are suitable.

Data augmentation can be used to concur the problem of class imbalance. When oversampling is combined with data augmentation, the over-sampled samples are

not merely duplicates of each other anymore but differently augmented samples. The disadvantage of this is that predicted values might also be data augmented, for example a blurred output. Carefulness is needed when working with data augmentation, since it can make the model worse when the 'wrong' augmentation techniques are applied. For example, a model which aims to classify digits from 0 to 9 would not benefit from a upwards/downwards flip, since then the 6 and 9 would be too similar.

2.9 VALIDATION

2.9.1 Training, testing and validation data sets

There exists much confusion in machine learning about the training, testing and validation of a method. An important bottom line is that as soon as you use a portion of the data to choose which model performs better, you are already biasing your model towards that data.

Training data should be data of which the expected outcome/result is known. In this case it would be that images labeled as landslides are truly landslides and vice versa. Then part of the data is used to further tune the model, to asses whether the model is well generalized, or over- or under-fitting. This data is often called validation data or test set in literature. Then finally there is the test set. The test set is for the evaluation of the final model. Hence this data can not be used until the final model is produced. The validation data can only be used to test the model during training, so it can not be used as training data or for evaluation of the final model. Many studies omit this test set and/or confuse the validation and test set. Often the test set is omitted since there is not enough data to spare for the test set [7].

The ratio of training, (validation) and test data is an important decision and it is not trivial. For the model to work well, there must be enough training data. Still, for the evaluation of the model, enough testing data is needed and for the testing of the model enough validation data. If the ratio is not chosen well, the model might become biased to one of the sets. Sometimes there is not enough data to ensure this, so it is chosen to neglect the validation data and just use a training and a testing set. Then usually the ratio of train and test data is 70,30 or 80,20 respectively. The most popular ratio for training, validation and testing is 50,25,25 percent of the total data respectively. What is 'enough' depends (among others) on the type of method and the data. For CNNs this is typically at least 10,000 images for training data.

2.9.2 Validation metrics

There exist many validation metrics, which are the most suitable depends (among other) per method. For pixel-based methods, such as classification, the following metrics are of importance. Object based methods, such as segmentation, are based on the idea of pixel-based methods. Furthermore, these metrics for pixel-based or classification methods can be interesting for object based or segmentation methods as well. The following definitions are pixel-based metrics and they define whether a pixel is classified correctly or not:

- True Positive (TP) is a positive sample which is classified as positive.
- True Negative (TN) is a negative sample which is classified as negative.
- False Positive (FP) is a positive example which is classified as negative.

- False Negative (FN) is a negative example which is classified as positive.

Pixel-based methods

Precision and recall are both pixel-based methods. Precision refers to the proportion of positive examples in the extracted results, while the recall refers to the proportion of the positive examples that are extracted among all the positive examples [53]. Their formulas are:

$$Precision = \frac{TP}{FP + TP} \tag{2.10}$$

$$Recall = \frac{TP}{FN + TP} \tag{2.11}$$

True Positive Rate (TPR) and False Positive Rate (FPR) are both pixel-based methods. FPR is the probability that a positive result will be given when the true value is negative. TPR is the same as recall, the probability that an actual positive will test positive. Their formulas are:

$$True\ Positive\ Rate = \frac{TP}{TP + FN} \tag{2.12}$$

$$False\ Positive\ Rate = \frac{FP}{FP + TN} \tag{2.13}$$

The TPR and FPR are often visualized in the same graph, which is called the ROC curve. This ROC curves gives the trade-off between the TPR and FPR while using different probability thresholds. Precision and recall are used in precision-recall curves gives this same trade-off, but are more appropriate for class imbalanced datasets, while ROC curves are more appropriate for balanced datasets [8].

Intersection over Union

Intersection over Union (IoU) is a metric which is often used for semantic segmentation, and it is an object-based method. It gives the ratio of intersection and union of two sets. These two sets are the true value and the predicted value. Since the true value is unknown, this will be the number of interpreted landslides. The larger the ratio, i.e. the IoU, the higher the accuracy rate [53]. The equation is as follows when stated in TP, FP, TP and FN;

$$IoU = \frac{TP}{FP + TP + FN} \tag{2.14}$$

For semantic segmentation, the formula is easier to interpret from the visualization in figure 2.17.

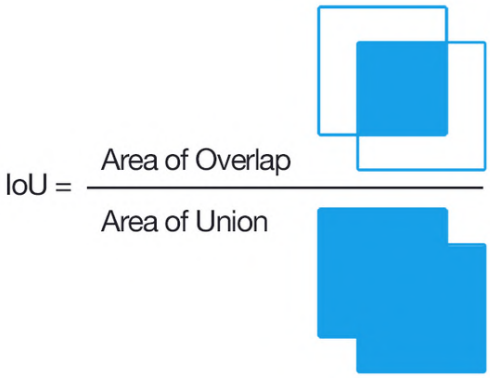

Figure 2.17: Intersection over Union. Source: [72].

F1-Score (Dice coefficient)

The F1 score, or Dice coefficient, is the harmonic average of Precision and Recall. The higher the F1 sore, the better the performance of the model. The F1 score's equation [53] :

$$F1 = (1 + \beta^2)\frac{Precision * Recall}{Precision + Recall} \qquad \beta = 1 \tag{2.15}$$

For semantic segmentation, the formula is easier to interpret from the visualization in figure 2.18.

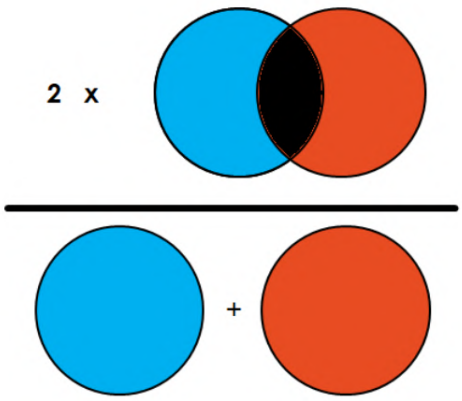

Figure 2.18: F1 score/dice coefficient is 2 * the area of overlap divided by the total number of pixels in both images [82].

The Dice coefficient is very similar to the IoU. They are positively correlated, meaning that one says model A is better than model B at segmenting an image, then the other will say the same. Like the IoU, they both range from 0 to 1, with 1 signifying the greatest similarity between predicted and truth.

3 METHODOLOGY

This chapter describes the methodology used in this research.

Two Deep Learning (DL) models are used, the U-Net and the conditional Generative Adversarial Network (cGAN), on three different image representations. By means of hyperparameter tuning and by variation of these three representations, multiple tests are done.

The first section 3.1 **Ground truth data** describes the data used in the study. The ground truth data is based on databases which contain landslide images. So landslides and these landslides databases are discussed. Furthermore the Copernicus Sentinel-1 (S-1) Synthetic Aperture Radar (SAR) satellite data and its properties are discussed. The second section 3.2 **Representations** describes how the three different representations of the SAR imagery are made. This is done with regards to temporal averaging and masking of the input image. Three methods are described which result in three different representations of the images used in the models. A workflow of each method is given in figures 3.1 and 3.2.

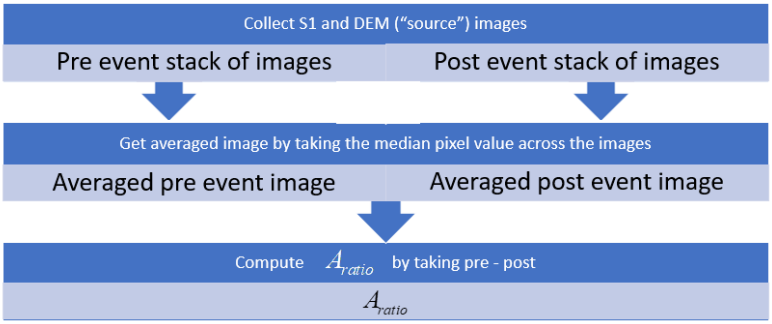

Figure 3.1: Workflow of the making of two representations, with regards to temporal averaging and masking. The first two representations are similar, so these are visualized in the same figure. Their difference lies in that one is masked based upon a slope image, while the other is not masked but has this slope image as an additional band.

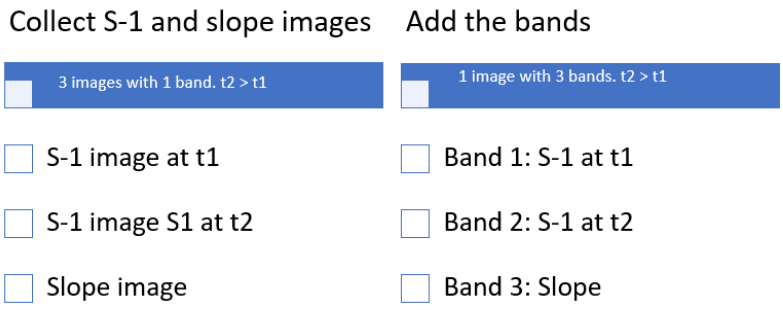

Figure 3.2: Workflow of the third method to produce the input images. S-1 refers to the Copernicus Sentinel-1 satellite. t2 and t1 refer to the time of the image, where time 2 is later than time 1.

The third section 3.3 **Production of input images** is about the production of the source and target images of the training,testing and validation data set. It will detail how the representations from section 3.2 are processed into the input images which are ready to be used in the U-Net and the cGAN models. A workflow diagram of this production is given in figure 3.3.

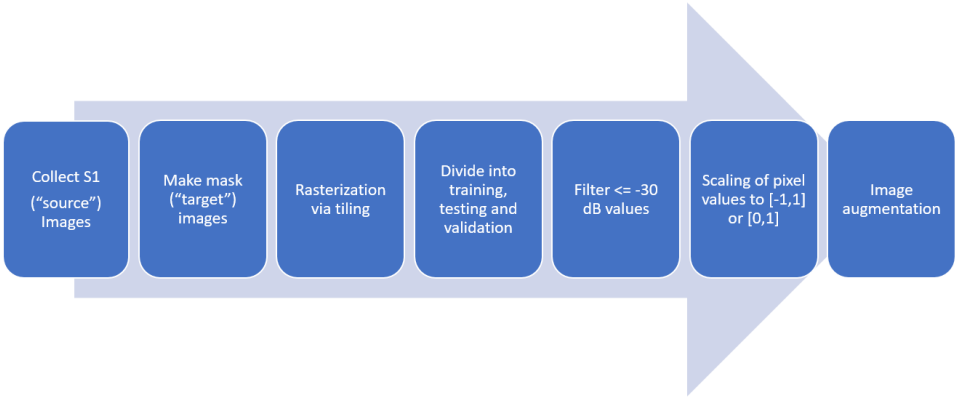

Figure 3.3: Workflow of the production of the training, testing and validation data set containing source and target images which will be used in the DL models.

The fifth section 3.4 **Model setup** describes the two models used in this study: a U-Net and Pix2Pix cGAN. The sixth section 3.5 **Model Evaluation** describes how to models will be evaluated and describes the choice of ratio of the training, testing and validation set. Finally the last section 3.6 **Software** describes the software used, the processing unit and how this is optimally used.

3.1 GROUND TRUTH DATA

3.1.1 Landslides

Landslides are gravity-driven movement of a mass of rock, debris or earth down a slope [16]. Landslides can be initiated by various causes, but landslides triggered by earthquakes or heavy rainfall are generally the most destructive and largest in size [47]. Landslides are variable in size, soil type and cause. They can be caused by heavy rainfall, due to which the soil becomes less stable and in combination with elevation/slope the land may slide. They can also be triggered by earthquakes, after a heavy earthquake many landslides can occur.

Landslides typically leave scars on the Earth's surface, which makes them recognizable in satellite imagery. When landslides occur in an area full of vegetation, it will leave a piece of ground bare. This piece of ground is called a landslide scar. In a snow filled area, the landslide leaves a barren footprint in the snowy landscape. These landslide scars can be confused with seasonal change of crops or deforestation. Especially rice fields, with their change in crops and are land and their location on slopes are notoriously mistaken for landslides. They can also be confused with roads, construction or flooding. Additionally, a landslide may change the topography of the affected area [68], the degree of slope and aspect of the topography of that area may change [78].

3.1.2 Landslide databases

In this study several databases are used which contain verified landslide polygons related to a large landslide event. These landslides have happened due to catastrophic earthquakes or extreme rainfall. After these events the landslides were mapped by experts, therefore these four databases are considered as ground truth in this study. This data is both used as validation, test and training data (see Section 2.9). The time range around the event of the landslides in the various regions is given in table 3.1.

	Hokkaido, Japan	Hiroshima, Japan	Lombok, Indonesia	Tsangpo gorge, China
Date t1 (dd/mm/yyyy)	06/10/2018	28/06/2018	28/07/2018	16/11/2017
Date t2 (dd/mm/yyyy)	07/10/2018	09/07/2018	20/08/2018	19/11/2017

Table 3.1: The time range of the landslide events per region considered in this study.

The first database is provided by the Geospatial Information Authority of Japan (GSI). This database covers the area of Hiroshima, Japan during the July 2018 heavy rain which caused many landslides in the area [of Japanese Geographers (AJG)]. Another database in Japan is provided by [89]. This covers the area in Hokkaido, Japan during the September 2018 6.7 Mj earthquake. This earthquake caused many landslides in the area. A database in Lombok, Indonesia is provided by [19]. This database contains 4823 landslides which where triggered by multiple sequential 6.9 Mw earthquakes in August 2018. Finally a database in the Tsangpo gorge, China is provided by [33]. These landslides where triggered by an earthquake in 6.9 Mw in November 2017.

The goal of this study is among others that the method can be generalized to worldwide use. Therefore the aim is to train on images from different areas in the world. Unfortunately, the databases used in this study are focused only in Asia. On the internet more landslide events and areas can be found. The disadvantage of these other landslide databases is that the featured landslides are not verified polygons, so these cannot be considered as ground truth. Moreover, the event date of the landslide might not be accurate, compared with the previously mentioned 4 databases. Finally, there is need for many landslide images as input for the DL model, too many to manually acquire a landslide database for the scope of this study. Therefore the main data focus will be on the four databases as mentioned in table 3.1. In the coming paragraphs further details will be given about each data set.

Hiroshima

The Hiroshima landslides were caused by heavy rainfall, which lasted a total of 72 hours. The began June 28^{th} and ended July 9^{th} 2018, so these dates define the pre- and post-event date respectively. The data set contains mapped verified polygons, but not a exact event date for each landslide/polygon. Since the rain lasted several days, the landslides occurred on different days most likely as well. The training data must contain truthful images, otherwise the model will be trained on false samples.The landslides are mainly debris flows. The data set contains approximately 3800 landslides, which are visualized in figure 3.4. There is no information on (visible) landslides prior to the rainfall.

Figure 3.4: The landslide database in Hiroshima, Japan. The red polygons (in the zoomed in figure these appear red, in the larger figure these are more black) are landslides.

Hokkaido

A M6.7 earthquake occurred on September 6th 2018 in Hokkaido region, which occurred one day after heavy rain from Typhoon Jebi. These events intensified each other and caused landslides, since the soil was soaked after the heavy rain and then shaking occurred [of Japanese Geographers (AJG)]. The aftershocks of the initial earthquake occurred until October 31st of 2018 as reported by during that time. The largest aftershock finally came on the 21st of February 2019. This aftershock was the strongest after the main one, with a magnitude of 5.7 MW aftershock struck 10 kilometer north of the epicenter of the main earthquake. This has been the strongest recorded aftershock of the earthquake. No major damages were reported, nor landslides.

The landslides are mainly translational landslides of small to medium scale with long run distances, with an average size of 8238 squared meters. There is one deep-seated landslide with an area of approximately 0.57 squared kilometers. The area contains also older landslides, some of which where reactivated by the September 6th earthquake. Luckily these reactivated and older landslides are also mapped, but the exact dates/causes of the latter are not known. In the area around there have also been landslides in the past. The (reactivated) landslides triggered by the earthquake and old landslides are visualized in figure 3.5.

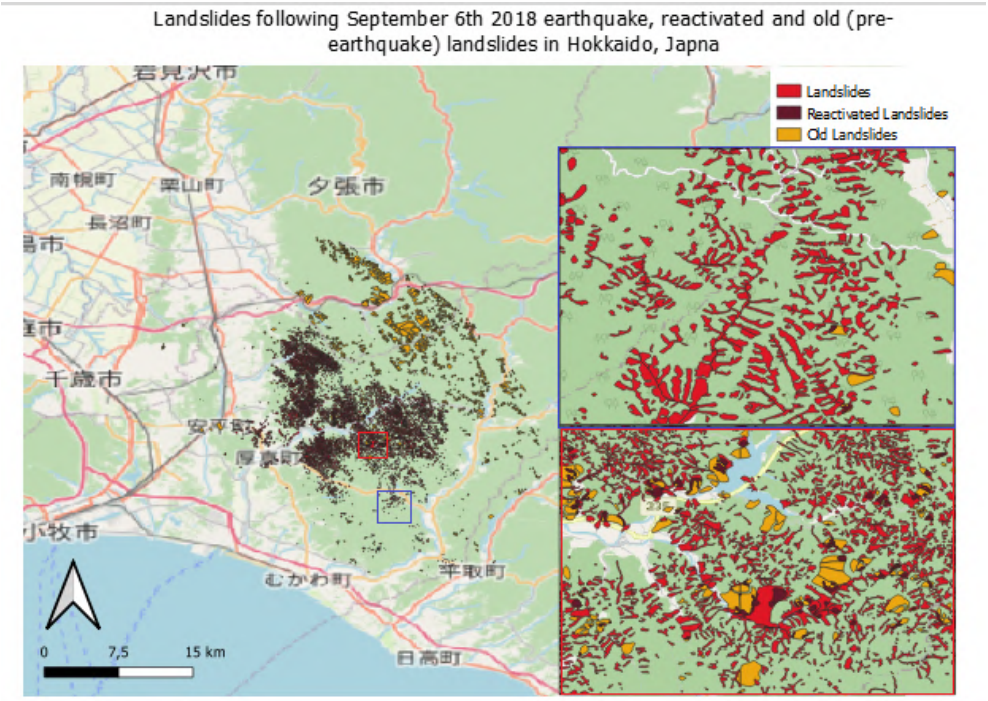

Figure 3.5: The landslide database in Hokkaido, Japan. The smaller figures on the right are zoomed in figures from the larger figure. The red, dark red and orange polygons (in the zoomed in figure these appear red, in the larger figure these are more black) are landslides. The bright red polygons are landslides caused by the event described in this section, the dark red landslides are reactivated by this same event and the orange landslides are old landslides which are still visible after the event.

Lombok

A sequence of four earthquakes caused many landslides in Lombok, Indonesia. The data set contains two sets of mapped landslide polygons, triggered by the August 5^{th} and August 19^{th} 2018 earthquake events respectively.

The first inventory, of the 05/08 event, comprises 4823 landslides. The second inventory, of the 19/08 event, comprises 9319 landslides. These are visualized together in figure 3.6 with different colors. The average area of individual landslides is approximately 1100 square meters.

The mapping of possible landslides triggered by the 28/07 event was not done due to persistent cloud cover. No significant landslides were recognized in the cloud-free portions of the first available image following the 28/07 event on 01/08. There were preexisting landslides before the first event in July 2018, these are not included in the mapped landslides polygons unless they were reactivated during the earthquake sequence. The 05/08 event caused a small tsunami with waves up to 50 centimeters. This could have caused floods, which may have led to false positives (be interpreted as landslides). This means there is some uncertainty regarding this data set. First of all, preexisting landslides are not mapped nor is their date and cause known. Secondly, the landslides of the first and second event are in the same area. Thus the definition of pre landslide images in this area must be done with caution, since undocumented landslides of the first event are present.

Lombok is subject to a wed and dry season, which runs from November to March and April to October respectively. Therefore the landslides were initiated in the dry season, so it is assumed no rain-induced landslides occurred then. Since the mapping was done with images of the dry season only, no rain-induced landslides are

assumed to be in either of the data sets.

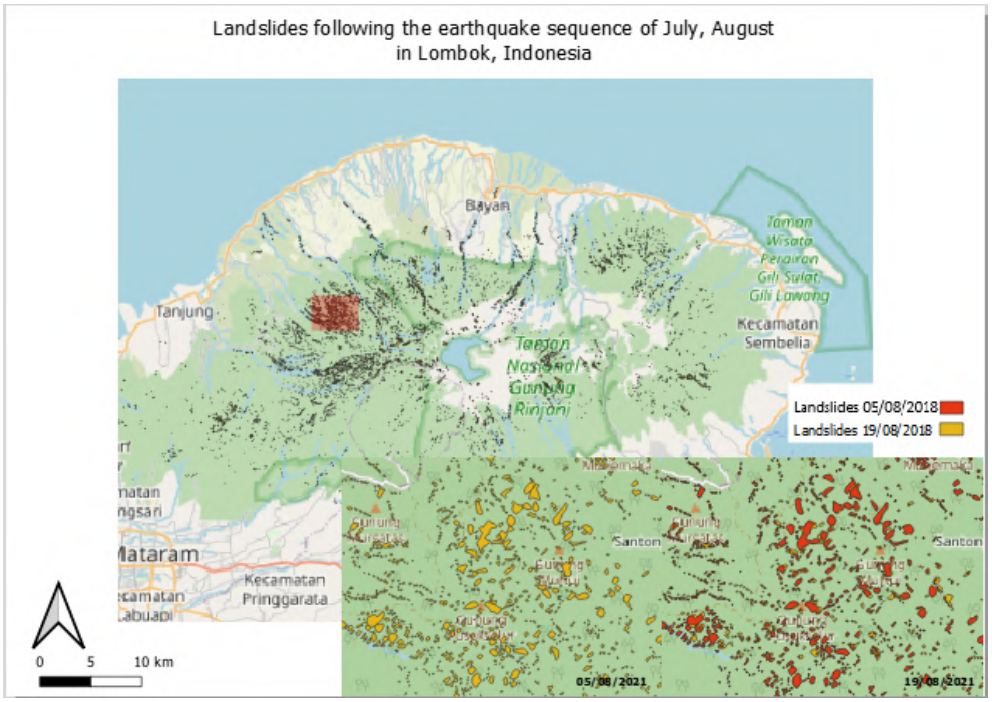

Figure 3.6: The landslide database in Lombok, Indonesia. The figure shows landslide polygons, and is zoomed in in two smaller figures on the bottom. The orange landslides are triggered by the event of 5 August 2018. The red polygons are landslides triggered by the subsequent event of 19 August 2018.

Tsangpo gorge

Tsangpo Gorge is a canyon along the Yarlung Tsangpo River in the Tibet Autonomous Region of China. Landslides were triggered by a Ms6.9 earthquake on November 18^{th} 2017. The earthquake induced a series of secondary hazards, aside from landslides also collapses and dammed lakes. The landslides triggered by the earthquake were a total of 766 with an area of 33.61 squared kilometer. Most landslides were shallow and small, some of these are not identified due to their small size. The largest cluster of landslides covers an area of about 4.9 square kilometers, the minimum is 62.79 square meters, and the mean area is about 0.05 kilometer.

Many debris flows happened in the area of the data set prior the to the earthquake. These debris flows can block rivers and form dammed lakes. The earthquake also caused several of these dammed lakes to overflow and thereby cause floods. Many landslides are debris like/ have loads of loose material. Due to the climate in the gorge, with ice and snow melting water in spring and heavy rain in summer, these loose materials are then turned into large debris flows. These kind of events fall outside of the period studied in this study. Figure 3.7 visualizes the approximately 700 landslides in the Tsangpo gorge area.

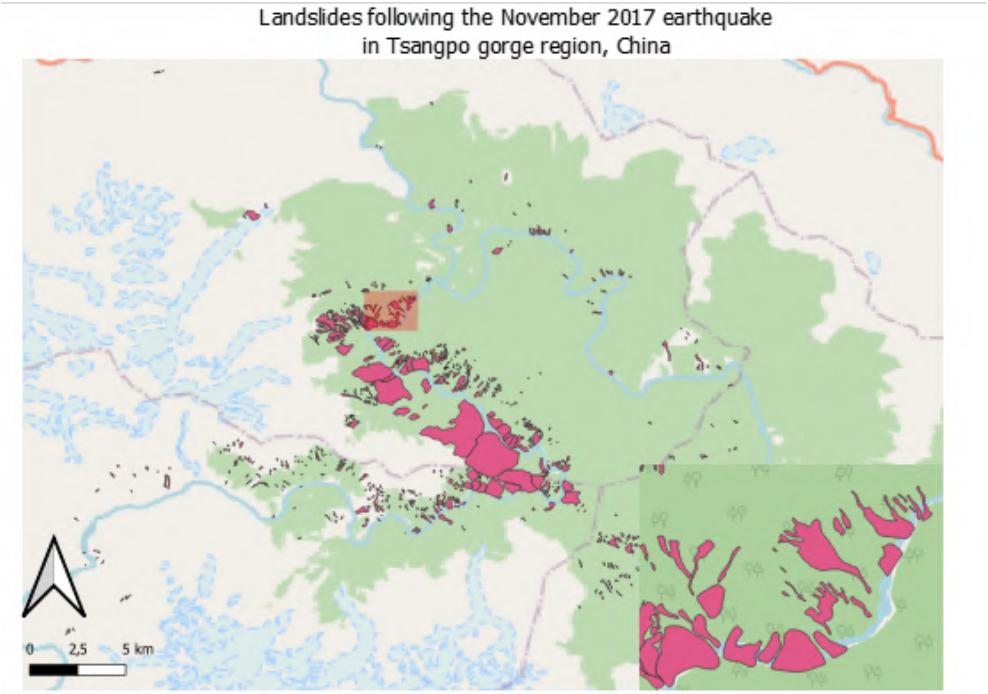

Figure 3.7: The landslide database in Tsangpo gorge, China. The pink polygons are landslides, the figure on the right is a zoomed in version.

3.1.3 Digital Elevation Model satellite imagery

In this method a Digital Elevation Model (DEM) is used to support the SAR satellite imagery. Currently, globally accessible Shuttle Radar Topography Mission (SRTM) data had a ground sampling distance of 1 arcsecond (approximately 30 m), and the relative vertical accuracy is approximately 6 m [1]. The slope product of the DEM is used in this study, since the slope has influence on the probability of a landslide event. In the rest of this study this slope product will be referred to as DEM.

3.1.4 Copernicus Sentinel-1 satellite imagery

The SAR data used in this paper is from the Copernicus Sentinel-1 (S-1) satellite constellation. Google Earth Engine (GEE) can access S-1 Ground Range Detected (GRD) products that are processed to remove thermal noise and are radiometric and terrain calibrated using SRTM DEM. Speckle is reduced by multi-look processing. The spatial resolution of the Interferometric Wide Swath (IW) GRD products is 20 x 22 m. The images have 10,25 or 40 m pixel spacing. Since minimum pixels size is 10 * 10 meters, landslides smaller than that will most probably not be detected, since the method is pixel-based. The polarization modes of VH are used in this study, since these are sensitive to forest biomass structure and therefore useful to identify landslides in vegetated areas [28]. The number of satellite images used per region can be found in table 3.2. A more detailed version of this table can be found in the Appendix, where the full names of the satellite images date used for each region are specified in the tables A.1,A.2,A.3 and A.4.

	descending, pre-event	descending, post-event	ascending, pre-event	ascending, post-event
Hiroshima	5	9	4	4
Hokkaido	4	5	4	5
Lombok	5	5	8	6
Tsangpo Gorge	2	4	3	3

Table 3.2: Amount of satellite images used for each region.

Float vs. Decibel product values

The S-1 SAR imagery data comes in two products, one where the amplitude is given in decibel (dB) and one in the original dimension in float. When translating to dB, some information is lost. Therefore manipulations of the data, such as applying smoothing filters or averaging over space and/or time, should be done on the float value.

Both dB and float products were tried in this study. The pixel values need to be scaled in order to be used in the model (more on this in Section 3.3.3). The scaling of the float values is difficult, since the range of the float is wide. This means that the maximum and minimum values, on which the scaling to an appropriate range for training of the neural network is done, have a large deviation from the mean value. This deviation is so large that almost all values were scaled to a margin of the range. In this sense the information of the scaled floats was lost. The dB product was made since the float product' range is so wide. The dB product has a relatively smaller range.

GEE recommends to filter SAR amplitude values smaller than or equal to -30 dB. This is done by replacing these values with -29 dB. The amount of pixels smaller than or equal to -30 dB on a random sub-sampled set of images is shown in figure 3.8. From there it can be concluded that even though most images contain pixels with these values, the relative amount is little. Therefore it is assumed this will not cause a significant change in the image and that these are considered outliers which may be removed.

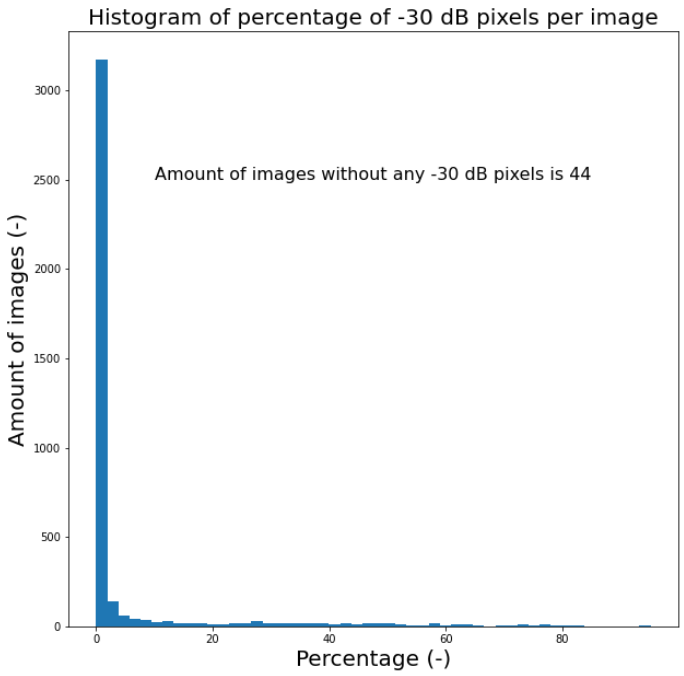

Figure 3.8: Percentage of amount of pixels with value smaller than or equal to -30 dB per SAR amplitude image. This is calculated over a random subset of approximately 3000 images from the training set. These images are smaller images made from the larger satellite images, which is detailed further in section 3.3. It can be concluded that almost all images include a percentage of smaller than -30 dB pixels, but that most of these percentages are smaller than 10 percent.

3.2 REPRESENTATIONS

In the introduction and previous chapter 2 the choice for SAR amplitude satellite imagery over other SAR satellite imagery products and optical satellite imagery has been discussed. In this section the possible representation of these SAR amplitude imagery will be explored, by discussing in what way temporal averaging and masks will be used. The various representations will be visually compared on a specific area in the region of the Hiroshima Prefacture, Japan (see figure 3.9).

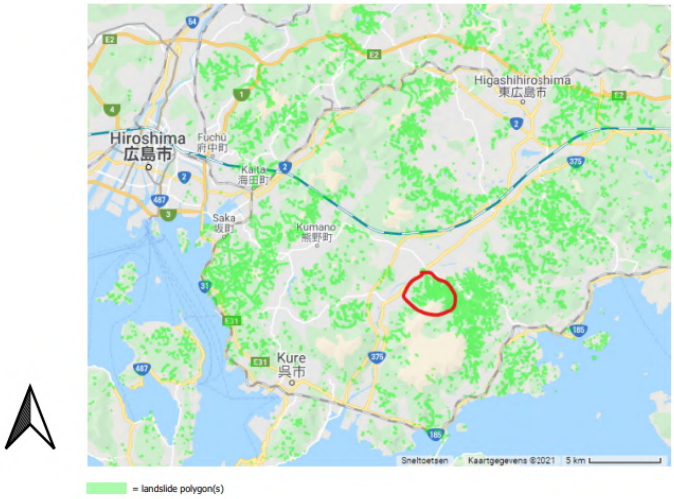

Figure 3.9: An area in the region of the Hiroshima Prefacture, Japan is selected, visible in the figure by the red circle. This area inside the circle will be used as example area to compare the different representations.

Figure 3.10 shows an optical image before and after the event (visualized are the red, green and blue band), from which the landslide scars can be recognized. Figures 3.10 and 3.11 both show an image before and after a landslide event. The images were selected as the closest image before and after the landslide event. For the optical images, only images which have 10 percent or less cloud cover were selected. On these images a cloud mask was applied. Figure 3.12 shows the landslides from the landslides database in a mask image. When compared to figure 3.12, Figure 3.10 shows that the landslide scars can be recognized with a human eye, while this is not the case for figure 3.11. If you look closely at figure 3.11 and compare it to figure 3.12, one might recognize some of the landslide patterns. It is more difficult then in figure 3.11.

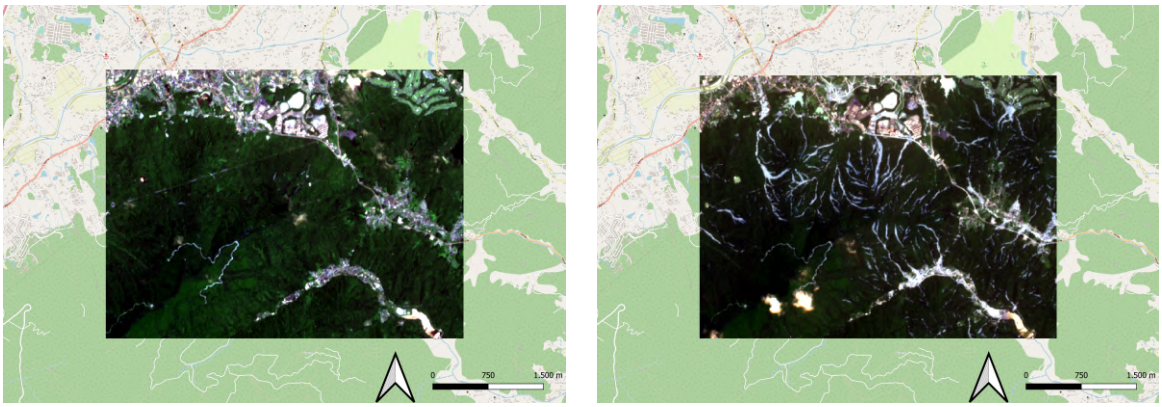

Figure 3.10: Images of the S-2 (optical) satellite, visualized in the red, green and blue band. The left image is the last available S-2 image before the landslide event, the right the first available S-2 image after the event.

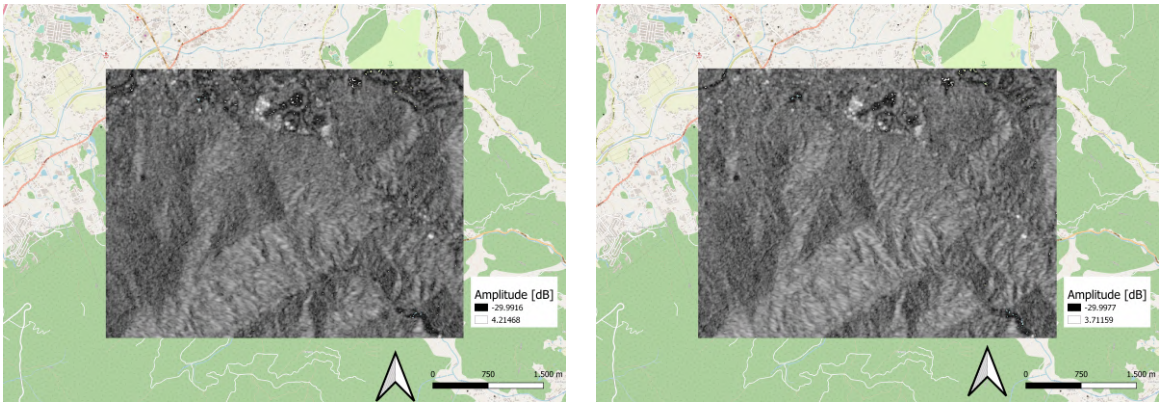

Figure 3.11: Images of the S-1 SAR satellite, with amplitude dB values. The images are of descending direction. The left image is the last available S-1 image before the landslide event, the right the first available S-1 image after the event.

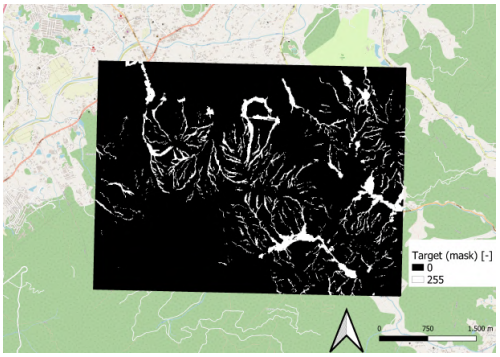

Figure 3.12: Mask image of landslide scars according to the (ground truth) landslide database. The white pixels represent landslide, the black pixels represent the background.

3.2.1 Temporal averaging

Averaging images in either the time and/or space can increase the quality of the image. By image averaging noise is suppressed. Also it gives the opportunity to include certain image manipulations such as a speckle- and a slope mask. These masks do not work well over single images. By aggregation the resolution over which dimension is aggregated, such as time or space, is reduced. Also it might not be possible to do aggregation depending on the aim of your study.

Numerous papers use aggregation of images before a landslide event and images after a landslide event [28; 44]. Aggregation here means the following : first the images pre- and post-event are stacked into a pile of images. Then the average of these images are taken. This is done by taking an averaging metric, e.g. the median, of each pixel across the images. In that way an averaged image pre- and post-event is produced. By averaging like this, noise and speckle is suppressed. [28] found that the more images are in the stack to be averaged over, the less noise and speckle will remain. This stacking of images also gives the opportunity to include certain image manipulations such as a speckle- and a slope mask. These masks do not work well over single images. The difference of these images can be taken, such that A_{ratio} is obtained. This A_{ratio} can give change detection, such as landslides. Since the pre- and post-image over which the difference is taken are reduced in noise and speckle, landslides can be recognized better.

By the stacking of this pre- and post-images the resolution over which dimension is aggregated, such as time or space, is reduced. Also the date of the landslide must be known for this approach. Furthermore if the post-image is averaged over a large time window, then that time has to be passed after the event. This means that an event could only be detected after a certain time has passed. If this would be e.g. a year this might be very unpractical regarding that the detection would be needed on short term with regards to rescue efforts and risk mitigation.

A moving average could be a solution, since it gives the opportunity to do averaging of the pre- and post-images while still keeping the exact event date unknown. A moving average can be done in many ways and would take a lot of testing. It could either be done in such a way that the pre- and post-images are averaged over the same amount of time. Or that the pre images are averaged over more time than the post-image. Also it could be that for some landslides more time averaging is needed then for other. For example, it could be that a particular landslide can be found when averaging over 3 months, while another can only be found within a month or less.

If the post-image are averaged over less time compared to the pre images, it might be possible to detect the landslide without letting a large time window pass after the event. It would be the average of 1 to n images and then the average of 1 to n+1 images. The difference between the two could give an anomaly, which would then suggest the anomaly is in the $(n+1)^{th}$ image. This would suppress noise, but at the same time not make use of the predefined information of the landslide event. The pre event averaged image then has very little speckle, but the post-image a lot. Then the difference of those, the A_{ratio}, might not be able to detect the landslide since the noise is too present from the post-event image.

These various options will be explored in this study, in the next subsections the various methods will be detailed.

3.2.2 Use of masks

Masks can be put over images, to filter out certain pixels based on a condition. For landslide detection slope and curvature masks are used often [28], which are based on the DEM image. Such as slope image for the example area as described by figure 3.9 is visualized in 3.13. In that way pixels which are unlikely to be a landslide, since they lie on flat areas, are filtered out. As such false positives are reduced. Masks are also used to concur noise and speckle, the speckle effect is explained in section 2.2.3. In this study only the slope and curvature masks will be applied. This is partly due to a time constrain, since these masks can be difficulty and need testing. Another is that the manual preprocessing is aimed to be as low as possible, since it is assumed that by using deep learning these sorts of manual processing will be less or even not necessary.

The advantage of not using masks during preprocessing saved time and possible biased decisions. Such masks are namely based on assumptions the user makes. For example, landslides generally occur on a slope within the range of 12 to 35 degrees [31]. Hence a slope mask will mask all areas which have a slope below 12 degrees. Then there might be landslides which lie on a smaller slope, even 5 degrees is possible, which are then masked. When the slope mask is put upon 5 degrees, some relatively flat services such as rice fields will not be masked which might cause false positives. When the DEM slope band is given to the model as is, the model will define the correlation between slope and landslides. The models definition might be better than the user's assumption and it saves time. Figure 3.13 gives an example of a slope image and of a mask such a slope image can produce.

In this case the mask is based on the condition that the slope must be greater than 0.5 degrees.

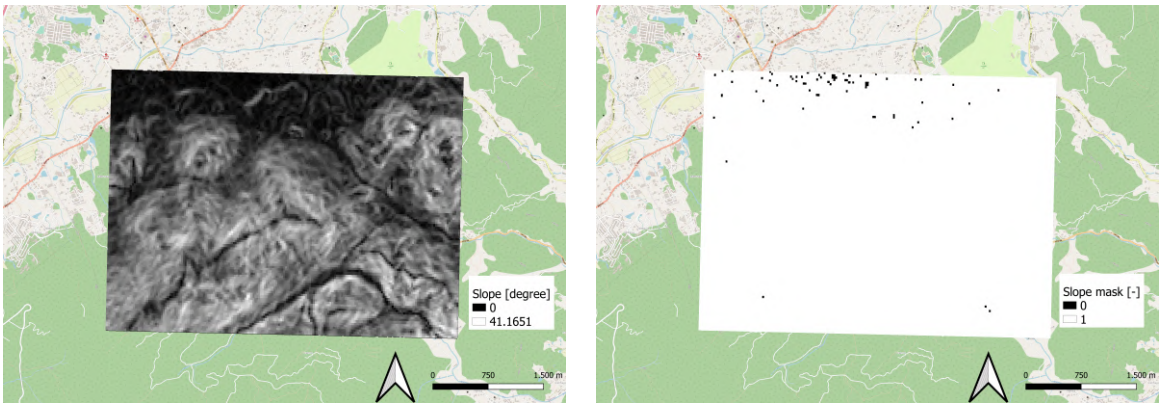

Figure 3.13: A slope image of the DEM (left) and the corresponding mask (right), based on a degree larger than 0.5 degrees.

3.2.3 A_{ratio} representation

This type of representation of the source image is inspired by [28]. The pre- and post-images are stacked and averaged over time. In this study this time is a maximum of a month, so the images averaged over still correspond to the images in table 3.2. The averaging is done by taking the median pixel value across the images. Then the difference of this averaged pre- and post-image, aka the A_{ratio}, is taken. Just as in [28] the A_{ratio} of the descending and the ascending direction are summed, and then divided by two.

This image is then masked by a slope and curvature mask, both slope and curvature are based on the DEM image. The slope mask is defined as greater than or equal to 0.5 degrees, e.g. all pixels which have a slope smaller than 0.5 degrees are filtered out. The curvature is made by taking gradient of the smoothed elevation. Then the x and y gradient are selected and added to each other, which produces the curvature. The curvature mask is produced by taking a curvature threshold of -0.005 $\frac{m}{m^2}$ and filtering the pixels on a curvature bigger or equal to this threshold. The A_rastio with these masks for the example area as described by figure 3.9 is visualized in 3.14. These images are styled to a color scale to enhance the landslide detection.

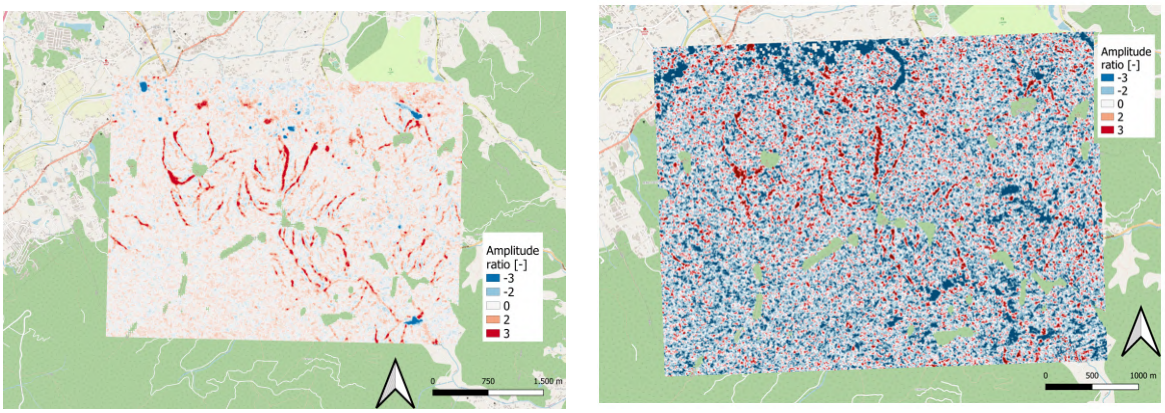

Figure 3.14: The A_{ratio} representations, with on the left temporal averaging done over multiple years and on the right after one month. This image is made in QGIS, these images are styled to a color scale to enhance the landslide detection.

3.2.4 Ratioslope representation

This image is similar to the A_{ratio} representation, only the slope mask is not included. So it is just an A_{ratio} image. In order to include the DEM slope information, this DEM slope image is added as a band to the A_{ratio} image. Therefore the resulting image has two bands, one which is the A_{ratio} and one which is the DEM slope band. This representation is called 'ratioslope' in this study. The A_{ratio} image for the example area as described by figure 3.9 is visualized in 3.14. The additional slope band is not added in this image, that can be seen in figure 3.13.

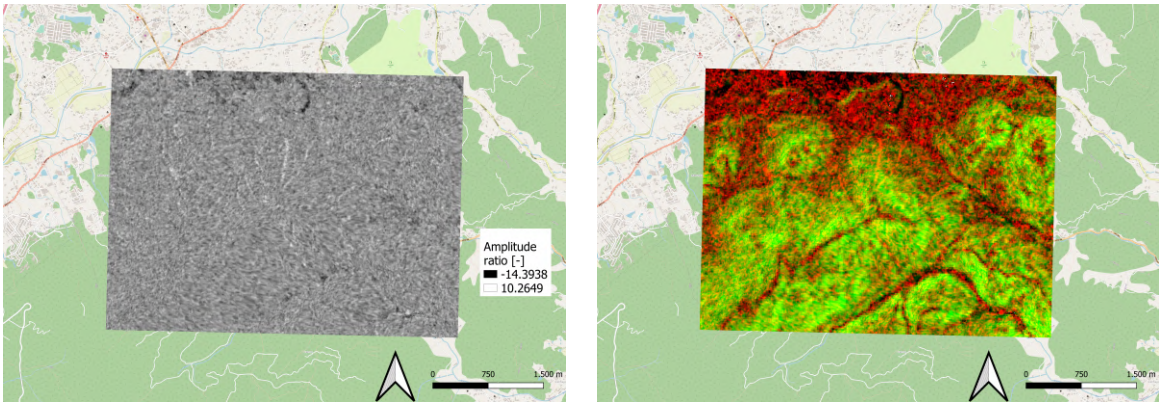

Figure 3.15: On the left the A_{ratio} image is visualized, the same as in figure 3.14, but without the mask and in the original color. On the right the slope band is added, which results in the two-banded ratioslope image.

3.2.5 Ampslope representation

The 'ampslope' representation of the source image is novel and first used, to our current knowledge, in this study. With this type of image, the images will be aggregated over time nor space, as to keep the temporal and spatial resolution. To that extent the images will be used in a composite image of three bands, which will be referred to from now on as 'ampslope' image. The advantage of this ampslope image is that temporal averaging, but also masks based on the slope derivative of a DEM will not be used. The advantage of not using temporal averaging has already been explained previously.

The first band of the ampslope image is the (SAR) image at time 1, the second band is the time at time 2 and the third band can be the slope. The slope is derived from a DEM which varies spatially but not temporally. These ampslope images are made in a consecutive manner, so each time step is represented in a ampslope image image. The importance of a timestep in the image, is since the absolute amplitude is not able to detect a landslide. However, the relative amplitude between time can [28]. Some studies take the difference image between two times, before the start of training by a DL model. This requires extra feature engineering which might not be necessary, as [5] showed that a CNN model could understand such a bi-temporal image.

One timestep is defined in this study as time step of one available image in a specific region to the next available image in that region. In the case of SAR images, the extra criteria of the same respective direction (in this case descending) is added. These are then called consecutive images. As described in Section 3.1.2, the landslide events and triggered landslides do not all happen in one timestep.

There are three kinds of ampslope images in this study; with pre-pre landslide, pre-post landslide and post-post landslide. The ampslope images can contain two images in the first two bands which are both before the event, or both after the event. These are so-called pre-pre and post-post images respectively. Post-pre landslide images are not present, the assumption is made is that the landslide scar will not be overgrown with vegetation within a month's time. For the pre-post landslide images, the pre image will be the last available image before the landslide event (the range as stated in table 3.1) while the post-image will be the first available image after the landslide event. Hence these pre-post landslides are not all, in contrast to the pre-pre and post-post, composites with consecutive images. The time steps per region for the pre-post composites are stated in table 3.1. Two pre-post ampslope images for the example area as described by figure 3.9 is visualized in 3.16.

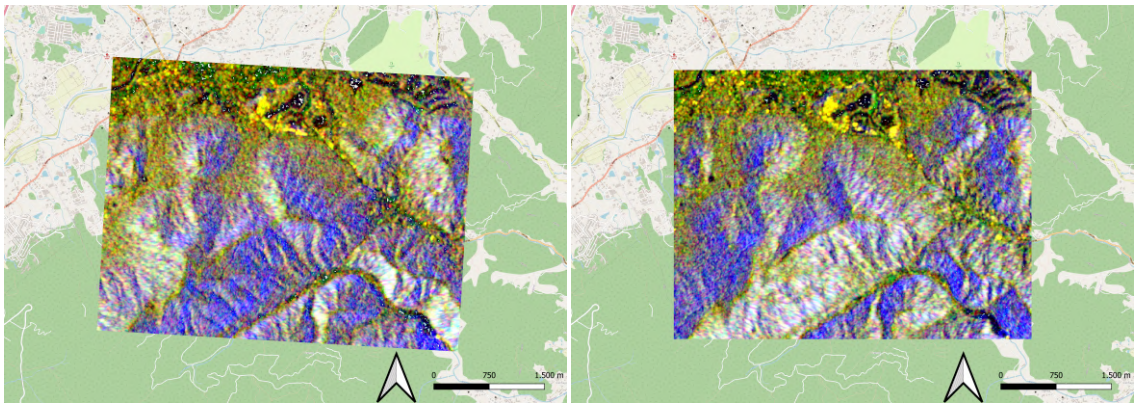

Figure 3.16: The ampslope images, where the left is of ascending order and the right of descending order.

3.3 PRODUCTION OF INPUT IMAGES

3.3.1 Independent (source) and dependent (target) images

The independent variables, also known as 'source' data, are made from the SAR amplitude and DEM imagery. These will be based on the representations as described in section 3.2. The dependent variable, known as 'target' data, needs to be predicted by the model. The landslides extents/shapes are to be predicted by the model, so in

this case the target data are images depicting the landslide shapes. These landslide shapes are given by the data sets described in section 3.1.2. The target images are masked images, which are produced by a threshold method. A threshold method is done by replacing pixels below a certain threshold with a black pixel, and pixels above this same threshold with a white pixel. In this way a masked image is produced, in this case the landslide polygons are replaced by white pixels, while the background is replaced by black pixels. The dependent variable corresponds to an independent variable. So in this case, a source image from SAR amplitude and DEM imagery describing a landslide is accompanied by a target image from a black and white mask describing the corresponding landslides. Vice versa, a source image without landslides corresponds to a target image which is completely black. In this way, there is no need for exact labeling per pixel as is required by various CML methods.

The landslide polygons described in Section 3.1.2 and the corresponding binary image as described above are visualized in figure 3.17 for the region of Tsangpo gorge, China [33].

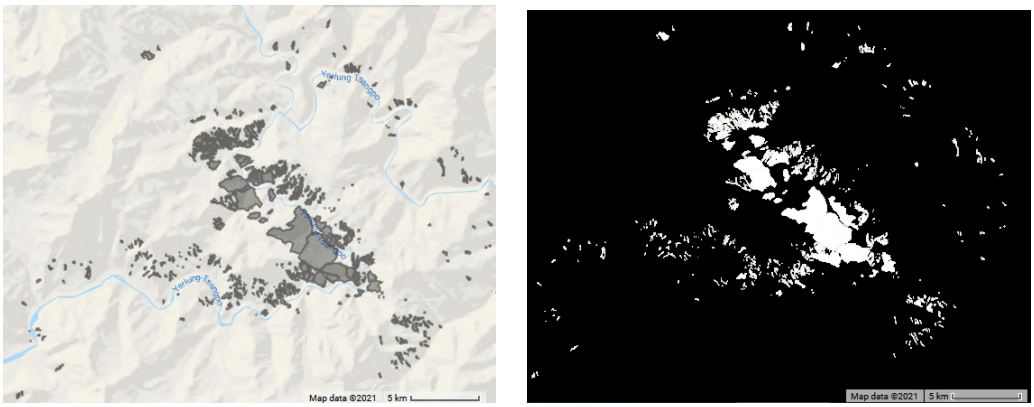

Figure 3.17: Landslide polygons (left) and binary image of polygons (right) of the area of Tsangpo Gorge, China.

3.3.2 Rasterizing images into tile-sized images

The areas of the landslide databases are rasterized in tiles, such that multiple smaller images are created. This is necessary since the DL models cannot train on very large images. Also the larger the images shape/size, the slower the DL model becomes. In order to have sufficiently small image sizes, rasterization of the large area is done. The source and target images are rasterized to the same tiles, such that they are of the same size. The advantage of this method is that the tiles do not overlap and are of the same size. The disadvantage is that some tiles have a higher density of landslides than others. A visualization of the tile method for the region of Hokkaido, Japan is given in figure 3.18.

The images are centred around the vertices of the landslide polygon, with some area around it. The scale of the images can vary, as long as the shape is the same. A constant shape is a requirement for input images in the same CNN. The input data must be of a square shape, since the model is build for that kind of shape. The model can be changed to fit a rectangular, non-square shape, but this would be an extra effort. The shape (size) of the images, e.g. 256*256, is a hyperparameter of a CNN and can be tuned while training. Due to time constrain, the images shape will be 256*256 pixels for all tests. The scale of the tiles within one region is all the same, between different regions the scale varies a bit but is approximately the same 2 * 2 square kilometers. The images of 256 * 256 pixels is thus approximately 2 by 2 kilometers, and has an approximate area of square kilometers. This means that each

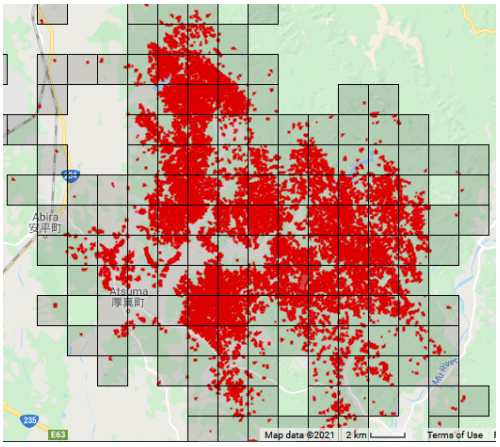

Figure 3.18: Tiling of geometry containing landslide polygons (in red) in the area of Hokkaido, Japan.

pixel is approximately 7.8 by 7.8 meters and has an area of 61 square meters. The amount of images, rasterized in pixels, with and without landslides are described in table 3.3.

	Landslide image	Non Landslide images	Total amount of images
Hiroshima	638	450	1088
Hokkaido	30	306	336
Lombok	232	99	331
Tsangpo Gorge	105	187	292
Total	1005	1042	2047

Table 3.3: Amount of images with and without landslides pixels for each region and in total.

3.3.3 Image scaling

When training the cGAN the pixel values of the images must be within the range of [-1,1], since this uses a hyperbolic tangent activation function. When training the U-Net the pixels values must be within the range of [0,1], since this uses a sigmoid activation function. To change the range of the images, the original range e.g. minimum and maximum pixel value must be known. For the target images this is straightforward, since these are black and white color images the range is [0,255]. For the source images, consisting of bands with SAR amplitude and DEM pixel values, this is not as straightforward. Therefore the minimum and maximum values of the SAR amplitude pixel values and the DEM pixel values are calculated for only the train data set. The test and validation set are not used for this, since that would make them unreliable. Consequently the source images of both the train, validation and test data set are scaled on the calculated minimum and maximum values of the train data set.

Since the pixels which are equal or smaller than -30 dB are filtered out, the minimum value of the SAR amplitude pixels will never be smaller than -29 dB. The minimum value of the DEM pixels will also never be smaller than 0, since only images on land are considered. The target images are only comprised of black and white pixels, such that the pixels have either the value 0 or 255. Hence these will also not be different for unseen or testing data. Therefore all these values can be mapped to either -1 for minima and 1 for maxima. The maxima of the SAR amplitude and DEM pixel values can be larger than the ones found in the training data.

Therefore those are not scaled to 1, as to ensure that unseen (test) data with possibly larger maxima values can still be applied to the model. Therefore these maxima are scaled to 0.8 instead of 1. The choice for 0.8 is an empirical choice, another value such as 0.5 or 0.7 might work better. For the A_{ratio} images, the maxima and minima are different. These are scaled to 0.8 and -0.8 respectively, since for this image in contrast to the SAR bands of the ampslope image, the minimum value might be different in the testing and/or validation data set.There is a possibility that this scaling to 0.8 instead of 1 will confuse the model, and that it might be better to then also scale the maxima of the mask image to 0.8. Due to time constraints this is not tested in this study.

	minimum	maximum	scaled minimum	scaled maximum
SAR image (source) (descending)	-29	26	-1	0.8
DEM image (source) (descending)	0	85	-1	0.8
SAR image (source) (ascending)	-29	24	-1	0.8
A_{ratio} image (source)	- 16*	12*	-0.8	0.8
DEM image (source) (ascending)	0	85	-1	0.8
mask image (target)	0	255	-1	1

Table 3.4: Minimum and maximum pixel values of the ampslope source (with bands SAR and DEM), A_{ratio} source and target images. The values to which these minima and maxima are scaled are specified in the last two columns. The minima and maxima are calculated from the training set only. *The A_{ratio} with mask and without have roughly the same minima and maxima values, the values displayed here have been rounded to two digits.

The source and target images after scaling for the three different representations as described in section 3.2 are visualized in figure 3.19. The masked values of A_{ratio}, which correspond mostly to the pixel values which before taking the ratio had a pixel value of -30 dB or smaller, are scaled to -1. This is because these filtered values are NaN, which give problems in the model. This is done for the A_{ratio} image and the A_{ratio} band in the ratioslope image.

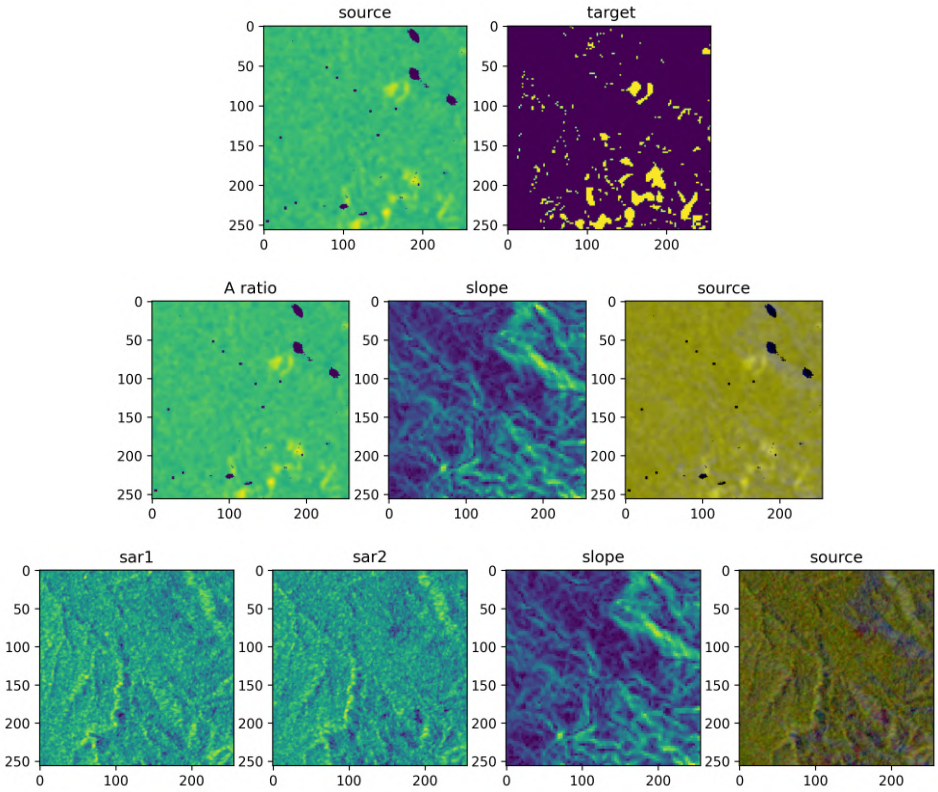

Figure 3.19: Source and target images of the three representations. The top row displays the A_{ratio} source image and the target image. The second row displays the ratioslope representation. The first two images show the two bands, the A_{ratio} and the slope band respectively. The third image shows the (ratioslope) source image, which consists of these two bands. The third row displays the ampslope representation. The first three images show the three bands, the SAR amplitude band pre- and post-image and the slope image respectively. The fourth image shows the (ampslope) source image, which consists of these three bands.

3.3.4 Data augmentation

In this study augmentation of the images of the training set is done during training of the model. This is done in both the cGAN and the U-Net model. The augmentation methods used are flipping of the image, both top to bottom and left to right.

3.4 MODEL SETUP

3.4.1 Hyperparameter optimization

The original loss function for the U-Net was the sparse categorical cross entropy, for the cGAN the binary cross entropy. Categorical cross entropy was used for the U-Net since the original data set had more than two classes. For two classes, the equivalent of cross entropy is the binary cross entropy. Since the problem at hand is a segmentation problem, dice loss or a combination of binary cross entropy and dice loss can be better than binary cross entropy. For a class imbalanced data set, binary focal dice loss works better than dice loss or cross entropy. Therefore the loss functions will be varying between binary cross entropy, a combination of dice loss and binary cross entropy and binary focal dice loss. For further explanation of loss function please refer to section 2.4.5.

The other model parameter which will be varied is the batch size and the number of epochs (for explanation see section 2.5.1). The most suitable epoch depends per training session and will be deduced from the learning curve. The batch size will be varied over the tests. Another important factor which is generally changed a lot is the images size/shape. It is chosen to leave image size the same along training. Firstly this is because there are already many variable parameters which result in many possible different tests, which costs time. Another is because the used cGAN was written on a image shape of 256 by 256 pixels, which is recommended by [34], so therefore this shape was kept. For the U-Net it could be changed, due to time constraints the 256 by 256 pixels shape was kept here as well.

Furthermore, augmentations of the images will be applied. Generally this make a model more powerful since it can handle more various data. However, too much augmentation can make the training data to varying resulting in overfitting [53]. To stay on the safe side only minimal augmentation is done, as mentioned in section 3.3.4. Also due to time constraint multiple kind of augmentations are not tested. The augmentation used is a horizontal and a vertical flip.

Besides training on the described data set(s) from this study, the models used will be run on their original data set as well. This is done to ensure that if the model cannot find a relation between the SAR amplitude images and the segmentation of landslides, that this may be due to the data rather than due to the model.

3.4.2 U-Net

The U-Net [71] is an image segmentation model. The U-Net used in this study was originally written for a data set with images of pets, where the pet was segmented from the background [15]. The source images were pets with a random background, the target images was a mask image of this pet with a background. So the model will give the target image as output to a source image, which shows the segmentation of the desired object. An example of the result of this U-Net on its original data set is given in figure 3.20.

Figure 3.20: The input image and corresponding segmentation mask of the Oxford Pets dataset, on which the U-Net is trained originally. Source: [15].

Now the same principal/ model will be used on the landslide data set where the landslide should be segmented from the background. The loss function is a mixture of binary cross entropy and dice loss, on a data set of just images with landslides. Figure 3.21 gives a schematic overview of the U-Net as used in this study, with the source and target images.

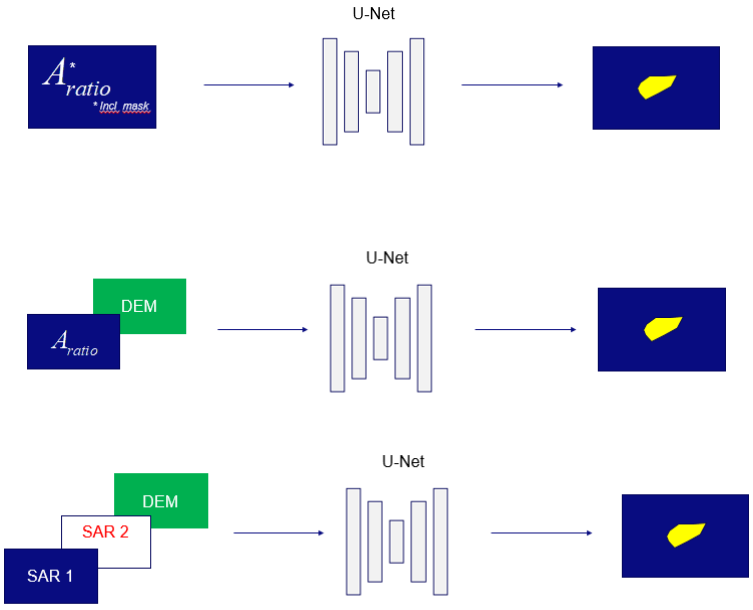

Figure 3.21: Scheme of the U-Net,with a source image as input and target images as output. The input image varies among the various tests, the first figure shows the A_{ratio}, the second the ratioslope and the third the ampslope image. The target image is a binary mask showing whether a pixel is a (part of) a landslide or not. The output image is generated by the U-Net, the goal is to produce the accurate output corresponding to the input image.

3.4.3 Pix2Pix conditional Generative Adversarial Network

The Pix2Pix conditional GAN [34] is explained in 2.7. The model described by [34] was further explained in [9] and the latter is used in this study, although some alternations were made. the details surrounding the activation function, loss function and weight distribution between generator and discriminator will be listed below. Also the alterations will be mentioned.

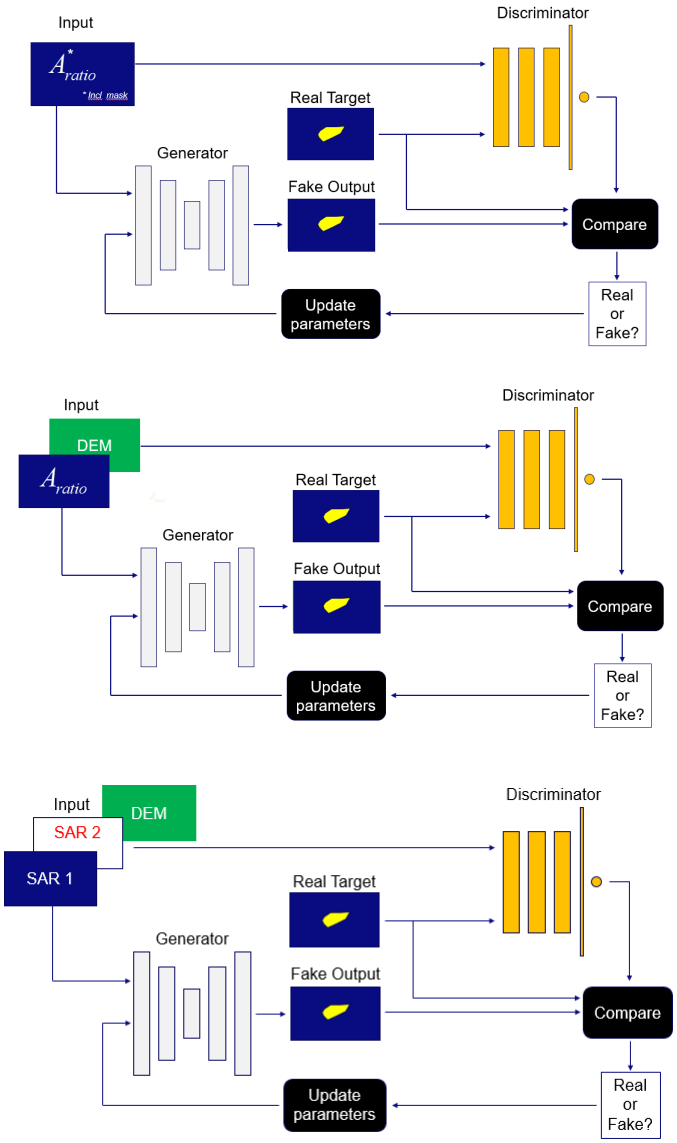

Figure 3.22: Scheme of the composite model,with a source image as input which is translated to an output image. The input image varies among the various tests, the target image is a binary mask showing whether a pixel is a (part of) a landslide or not. The output image is generated by the generator, both a fake and a real output image is made. Then the discriminator has to say which is true and which is fake. Through this process both generator and discriminator learn. The goal is to have a generator which generates the accurate output corresponding to the input image.

Figure 3.22 shows the cGAN as used in this study, In this case the source image is translated to a target image, where the target image is a binary mask which depicts a landslide or not. As previously mentioned in section 3.2, the source images in this cGAN model are varied.

3.5 MODEL EVALUATION

For both the U-Net and the cGAN model two learning curves (see section 2.5.3) are made. One learning curves presents the loss(es) (see section 2.4.5), while the other presents the Intersection over Union (IoU) (see section 2.9.2). For the U-Net only the best model will be saved, which is defined by the loss function. Since for the

cGAN this is less straightforward, every 10 epochs the model will be saved. Then after training, by evaluating the learning curve, the best model can be chosen and possibly trained further.

The IoU metric is calculated only for images containing landslide polygons, for both the training and the validation data set. The validation data set therefore consists only of images containing landslide images, both smaller and larger. This is not altered for the runs with images containing only large landslides. This is since it was found that images containing no landslides, so masks which are all black, are easy for the cGAN to generate. These give back a IoU of a perfect 1. The images with landslide polygons give back a lower IoU. Since this is what we are most interested in, and since it can be assumed that the model is good at creating true negatives, only the IoU of the images containing landslides was saved during training .

3.5.1 Training, testing and validation data set ratio

The division of training, testing and validation is chosen to be 70, 20, 10 respectively. There has been no experimenting with other ratios. The ratio may differ a bit per region due to class imbalance, as explained in section 2.8. The scaling of pixel values is based on the minima and maxima of the train data set. Therefore the images which are part of the training data set, and in turn also those of the test and validation data set, are restricted by this calculation of the minima and maxima. This calculation takes some time, so this is only done once. Therefore there can be no experimenting with K-fold cross validation, since in such a scheme the train data set and validation data set would change between training tests. Therefore varying multiple sets with varying training/validation/testing ratio takes a lot of time. Also there are many others parameters to experiment with in this study. Therefore it is chosen to not experiment with this division ratio.

The amount of images in the training, testing and validation set for each input image method is specified in tables 3.5 and 3.6.

	ascending	descending	combined
training set 'large' landslides	201	201	402
training set all landslides	528	528	1056
validation set all landslides	155	154	309
testing set all landslides	319	318	637

Table 3.5: Amount of images for the training, testing and validation sets of the ampslope images. The training set is both made for 'large' landslides and all landslides, but the testing and validation set are with all landslides only. Ascending and descending refer to the orbit direction, combined is the combination of both the ascending and descending sets. Both ascending, descending and combined sets are used.

	A_{ratio} masked	ratioslope representations
training set	1316	1287
validation set	368	371
testing set	148	182

Table 3.6: Amount of images for the training, testing and validation sets of the A_{ratio} and ratioslope representations.

3.5.2 Landslide images and non-landslide images

The models can be trained only on landslide images or also on images containing no landslides. The advantage of training on only landslide images is it is easier for the model to successfully detect landslides. A disadvantage of training on only landslide images, could be that the model becomes biased towards landslides. Then the model might not understand an image which does not contain a landslide. Or worse, that the model would always give back some landslide polygons, even when it is tested on an image which does not contain any landslides. It can also be argued that from the images containing these landslides, a large part of the image still contains no landslide. E.g., the ratio of landslides over non landslide pixels is quite low, since many images contain a small landslide and for the rest area which does not cover any landslides. For this reason, pixel-wise the data set with images which all contain landslides is not biased on landslide pixels, since it might contain as many or even more non landslide pixels. The ratio of landslide versus non landslide images according to table 3.3 is approximately 1, so it does not cause class imbalance (see section 2.8). However, if this is viewed at pixel level, then there might be a class imbalance.

For the A_{ratio} and ratioslope representations, the amount of non-landslide images and landslide images can be found in table 3.3. This is since the amplitude log-based ratio is always taken over a landslide event. Therefore only tiles which do not contain a landslide give images without landslides. If the ampslope representations are only taken with the SAR image at band 1 and SAR image at band 2 before and after the landslide event, then the amount of non-landslide images and landslide images also correspond to table 3.3.

The ampslope representations can contain more non-landslide images than the A_{ratio} and ratioslope representations. This is since for the ampslope, the SAR image at band 1 and SAR image at band 2 can also both be before the event or both after the event, instead of before and after the event respectively. This could also be possible for the other representations, but was not tried out.

Beside tiles which do not contain landslides, the images containing landslides only have 1 (pre-post) image, while the images containing no landslides (pre-pre,post-post) have multiple. Since the images are taken over a range of a month before and after the landslide event, there could be multiple pre-pre and post-post images per tile, while there could only be 1 (or sometimes none) for that same tile. This results in class imbalance. Then the data is cut op into train, test and validation data set. The test and validation data set cannot undergo any augmentation or resampling, unlike the training set. Therefore it is important that they contain some landslide images. Since for the training data set, these could be increased by resampling and/or data augmentation, while for the test and validation data sets this is not the case. For some regions the very common 70/30/20 is used, while for others there is a little deviation from this to ensure the test and validation data set have enough images which contain landslides.

3.6 SOFTWARE

Part of the method is developed in the Google Earth Engine GEE browser-based graphical user interface using the JavaScript API, while part is done in Python where GEE can also be used as a package. In the JavaScript API of GEE the initial script was written with functions to filter and accumulate the images. This API is used initially, since functions and resulting images can be easily tested and visu-

alized in this format. Later this script was translated to Python in Google Colab, which is connected to the GEE server and functions trough the Earth Engine (ee) module package. Google Colab hosts a Python Notebook, in which modules such as Tensorflow and Keras can be used. These modules are well known for their deep learning applications, a DL can be build with help of these models.

3.6.1 Processing Unit

In google Colab multiple central processing units Central Processing Unit (CPU) are possible. CPU is the electric circuitry that executes the algorithm from a computer program. The kind of CPU thereby has a influence on the speed of the computation. Google Colab offers a free graphics processing unit Graphics Processing Unit (GPU), which is the Tesla K80 GPU, which is designed for video tasks and then takes over (part of) the tasks of the CPU. GPU works on convolutions specifically, and makes the computation time of any CNN a lot faster than with CPU.

Google Colab also offers a tensor processing unit Tensor Processing Unit (TPU), which is a feature which can only be used when using the Tensorflow package. Data must be transformed into tfrecords (Tensorflow records) and uploaded to a cloud platform such as Google Cloud Services (GCS) to store the data and run the model. While CPU and GPU are freely available, TPU is not in the sense that data storage in the GCS is not free. Instead of GCS, Kaggle [**?**] can be used to upload data to cloud storage for free. In that way it is possible to run TPU on Google Colab for free. In this study TPU is used instead of GPU, since with using GPU the RAM memory of Colab runs out. With use of TPU, the tasks are paralleled into eight TPUs which makes the training 8 to 10 times faster than with GPU.

3.6.2 Optimal usage

The computation time is faster with a higher batch size. The code runs on eight TPUs in parallel. When the batch size is higher, the advantage of these parallel TPUs increases, such that the total runtime is lower. The rule of thumb is to use batches of 128 elements per core (example: batch size of 128*8=1024 for a TPU with 8 cores). At this size, the 128x128 hardware matrix multipliers of the TPU (see hardware section below) are most likely to be kept busy. Interesting speedups start to happen from a batch size of 8 per core.

For optimum memory usage, use the largest batch size that will fit in memory. Each TPU core uses a 128 x 128 memory cell matrix for processing. In general, the batch size should be evenly divisible by 128 to most effectively use the TPU memory.

4 RESULTS

Three types of representations of the input images were introduced in section 3.2. The A_{ratio}, the ratioslope and the ampslope representation. Two models are trained, a U-Net and a Pix2Pix conditional Generative Adversarial Network (cGAN). Various tests have been done with these three representations. Multiple hyperparameters of the models have been changed over the coarse of various tests, mainly the the batch size, the loss metrics and the amount of epochs.

The first section 4.1 **Human interpretation** visualizes the aforementioned three representations. These three representations are compared, as well as some variations of these representations. The quality of the representations is discussed in terms of human interpretation, whether landslides can be recognized by a person. Furthermore the importance of look direction of the Copernicus Sentinel 1 (S-1) satellite on the quality of the images and forthcoming possible landslide detection is discussed. Also the amount of temporal averaging on said quality and possible detection is discussed.

The next three sections are dedicated to the training of the three respective representations on the two models. For each representation various results are summarized in their respective section. Firstly the learning curves are given, one featuring the loss metric(s) and one the mean Intersection over Union (mIoU). This is done for both the U-Net and the cGAN. Furthermore for the cGAN model, a random subset of the source, target and generated target images during training are visualized. Also the predicted (target) images of the testing set of the cGAN are visualized. The results from these three sections, sections 4.2,4.3 and 4.4, are discussed in section 4.5 **Discussion**.

4.1 HUMAN INTERPRETATION

In this section an example area in Hiroshima, Japan will be used to compare the various representations. Figure 4.1 shows the target image, which visualizes the landslide scars as stated by the ground truth landslide database. This database is described in section 3.1.2.

Figure 4.2 shows the optical representation by Copernicus Sentinel-2 (S-2) satellite. The optical images are single images before and after the landslide, the least cloudy images within a month before and after the event are selected respectively. The optical image after the landslide event shows the landslide scars clearly, when compared to the target image in figure 4.1.

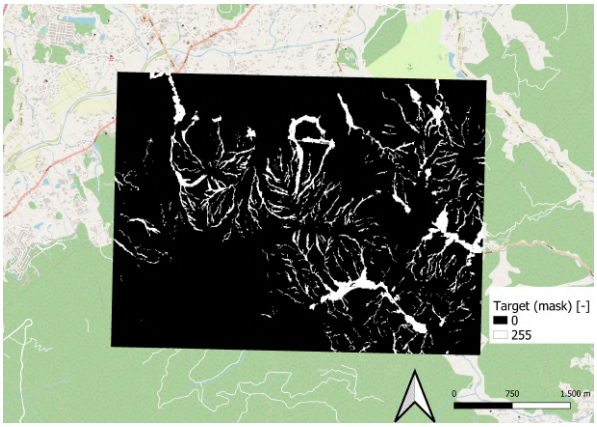

Figure 4.1: The target image shows the landslide scars. This image is a mask, which represents the landslide scars as white pixels and the background in black pixels.

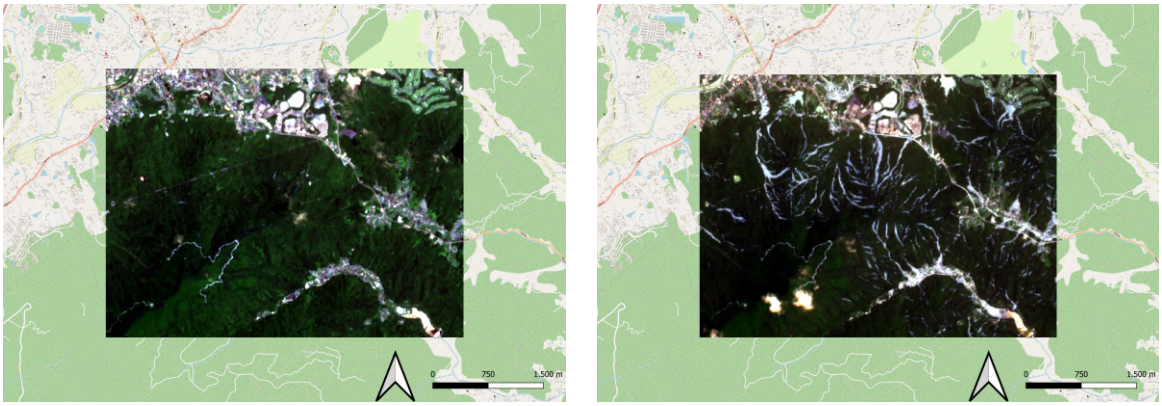

Figure 4.2: This figure shows Copernicus Sentinel-2 (S-2) optical satellite images before (left) and after (right) a landslide event. The optical images are single images before and after the landslide, the least cloudy image within a month's range from the event is selected.

Figure 4.3 shows pre- and post landslide event SAR S-1 amplitude images. When compared to figure 4.1, the landslide scars are somewhat visible in the post-event images. Still, they are not as clear as the corresponding optical images from figure 4.2. This is partly due to the spectral resolution of the optical images as opposed to that of the SAR images. The visualization of the green, red and blue spectral bands are easy to interpret as humans. For the other part it is due to the speckle and geometric effects of the SAR amplitude imagery, which decrease the quality of the image and make it harder to interpret. Temporal averaging of SAR amplitude imagery decreases speckle and geometric effects, thereby increasing quality and the ease of interpretation of the image. This can also be seen when comparing the single post-image (top right image) and the averaged post-image (bottom right image) of figure 4.2.

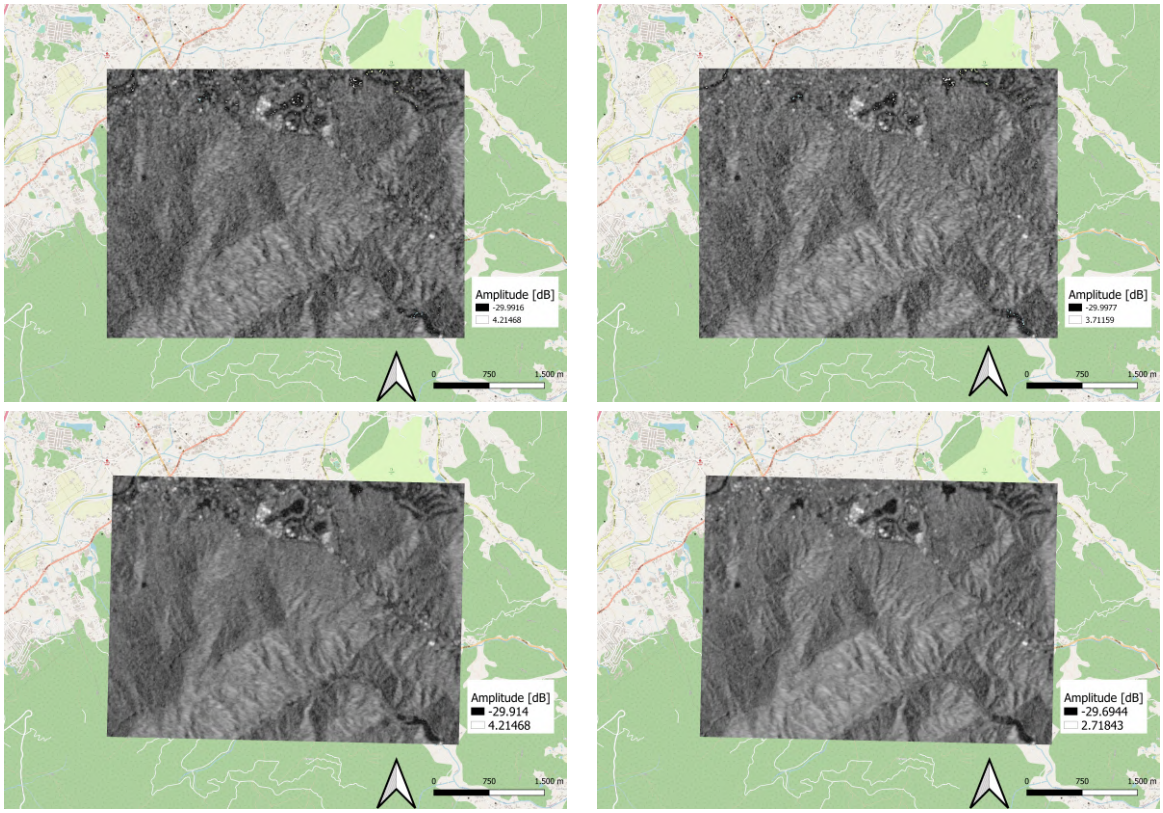

Figure 4.3: This figure shows images before (left) and after (right) a landslide event. The top row shows images which have not been temporally averaged, the pre image is the last available image before the event and the post one is the last available image after the event. The bottom row shows images which have been temporally averaged over the course of one month.

In section 2.2.1 the importance of viewing geometry in SAR amplitude imagery was discussed. Sivasankar et al. [78] argued that look angle and look direction must be consistent in imagery in order to compare them, so only images of same direction can be compared to each other. Sivasankar et al. [78] also found in their study that descending images could detect landslides, while ascending could not. From this it can be concluded that in some cases landslide detection can be done with images of descending direction, but not with those of ascending direction (and probably vice versa). Handwerger et al. [28] averaged images from both directions, which decreased geometric effects and thereby increased the quality and ability to detect landslides. Therefore Handwerger et al. [28] showed, in contrast to Sivasankar et al. [78], that images of the two directions can be combined since that improves possible change detection. This is visualized in figure 4.4. The image of descending direction shows the landslide scars more clearly than that of ascending, which corresponds to Sivasankar et al. [78] findings. When these two images are averaged into one, as Handwerger et al. [28] found, than the geometric effects are indeed visibly reduced. This makes it easier to recognize the landslide scars.

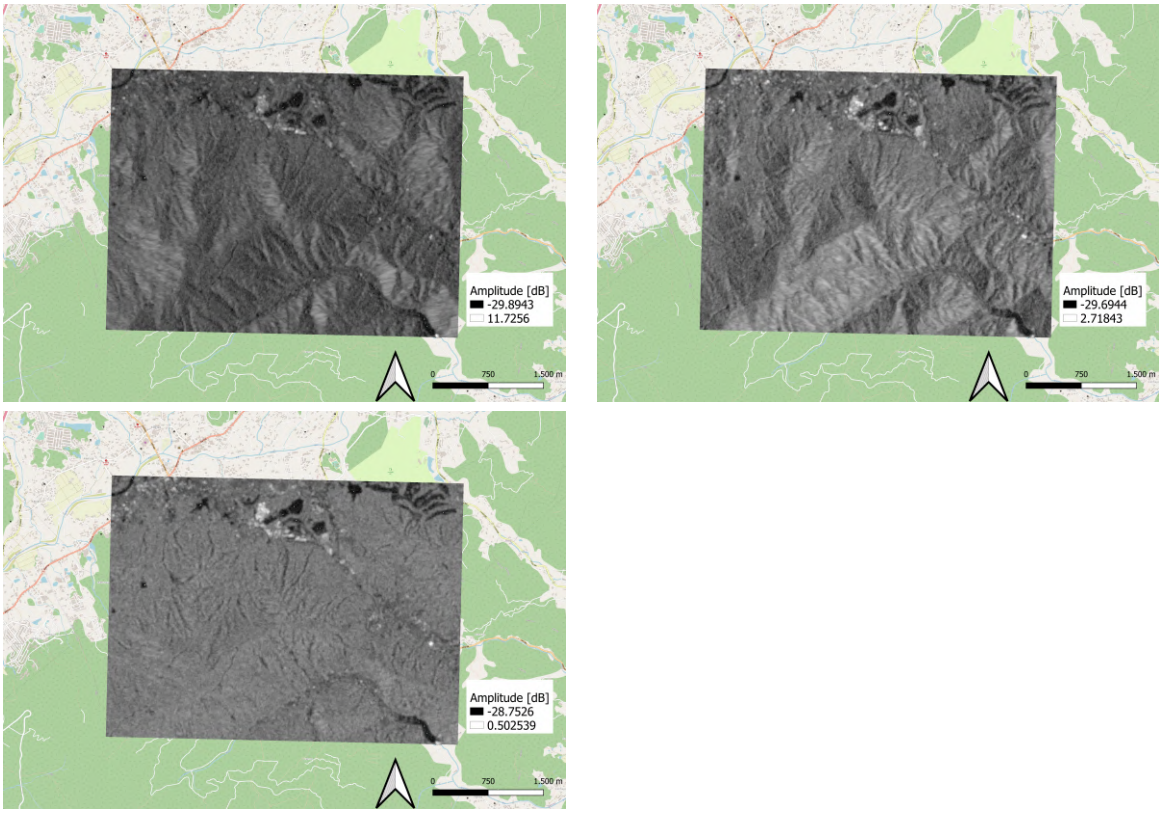

Figure 4.4: This figure shows the post-event images, which have been temporally averaged over a month. The top image is averaged over images of ascending, the second of descending. The images of ascending and descending during a month post-event have been averaged together, resulting in the third image.

Figure 4.5 shows the log-based amplitude ratio, the A_{ratio}, which has undergone various forms of temporal averaging. In this way, the influence on temporal averaging on the quality and ability of landslide detection of the A_{ratio} can be compared. The A_{ratio} which is averaged over multiple years shows the landslide scars the most clearly. Also the A_{ratio} image which is averaged over one month shows the landslide scars more clearly than the one which has undergone no temporal averaging.

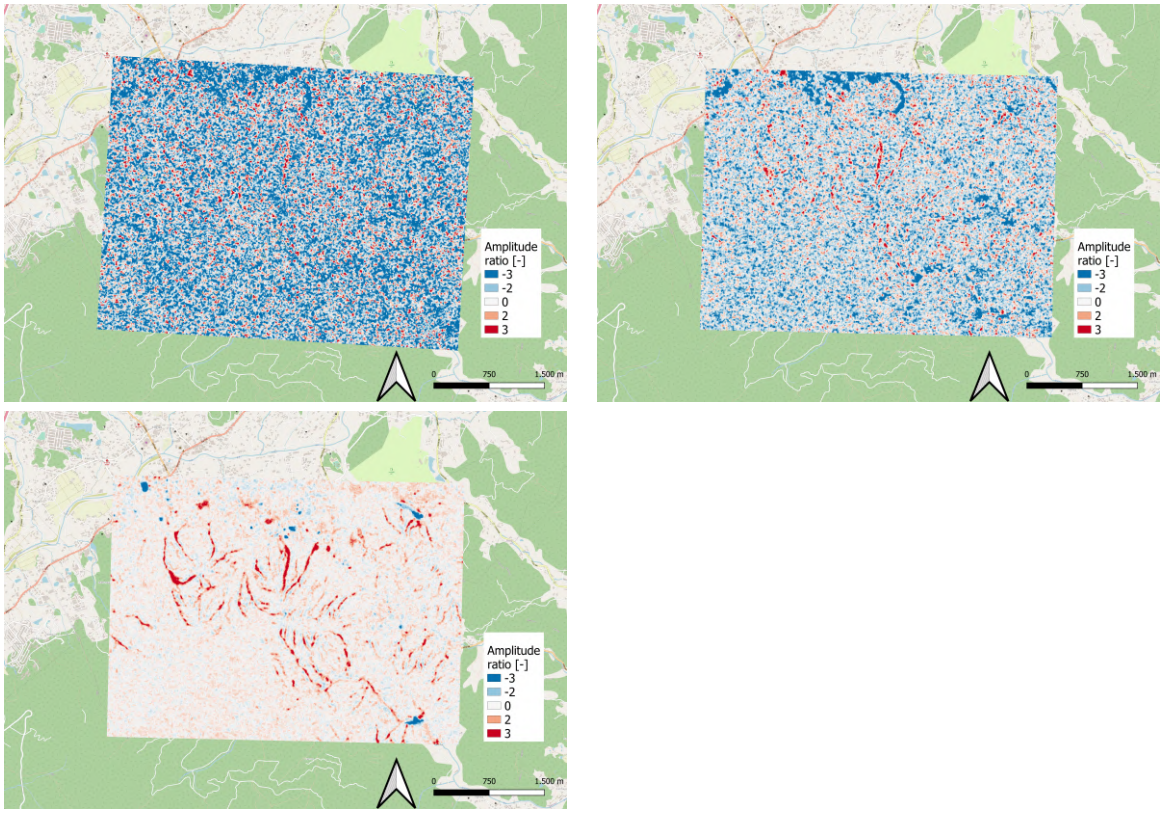

Figure 4.5: This figure shows the log-based amplitude ratio, the A_{ratio}, for three images. The top image has not undergone any temporal averaging, the A_{ratio} is taken from a single pre- and post-event image. The second image shows an A_{ratio} where the pre- and post-event images are temporally averaged over the course of one month. The third images shows an A_{ratio} where the pre- and post-images are temporally averaged over the course of two years. The images are visualized with a color range.

Figure 4.6 shows the same A_{ratio} images as in figure 4.5, but then in masked form. In this way they can be compared to the target mask from figure 4.1, such that the ability to detect landslides can be visualized. The mask is made by filtering the pixels on a threshold of 2.5, where the landslide pixels are assumed to have an A_{ratio} above 2.5 and the nonlandslide pixels underneath. From the mask of the A_{ratio} averaged over years some landslide scars can be read, more than from the other two. From the mask of the A_{ratio} averaged over one month a few landslide scars can be recognized when compared to the actual target (figure 4.1). Still many scars are not visible, and some pixels are wrongly classified which looks like noise. From the mask of the non-averaged A_{ratio}, landslide scars cannot be recognized. The mask looks like a noisy image.

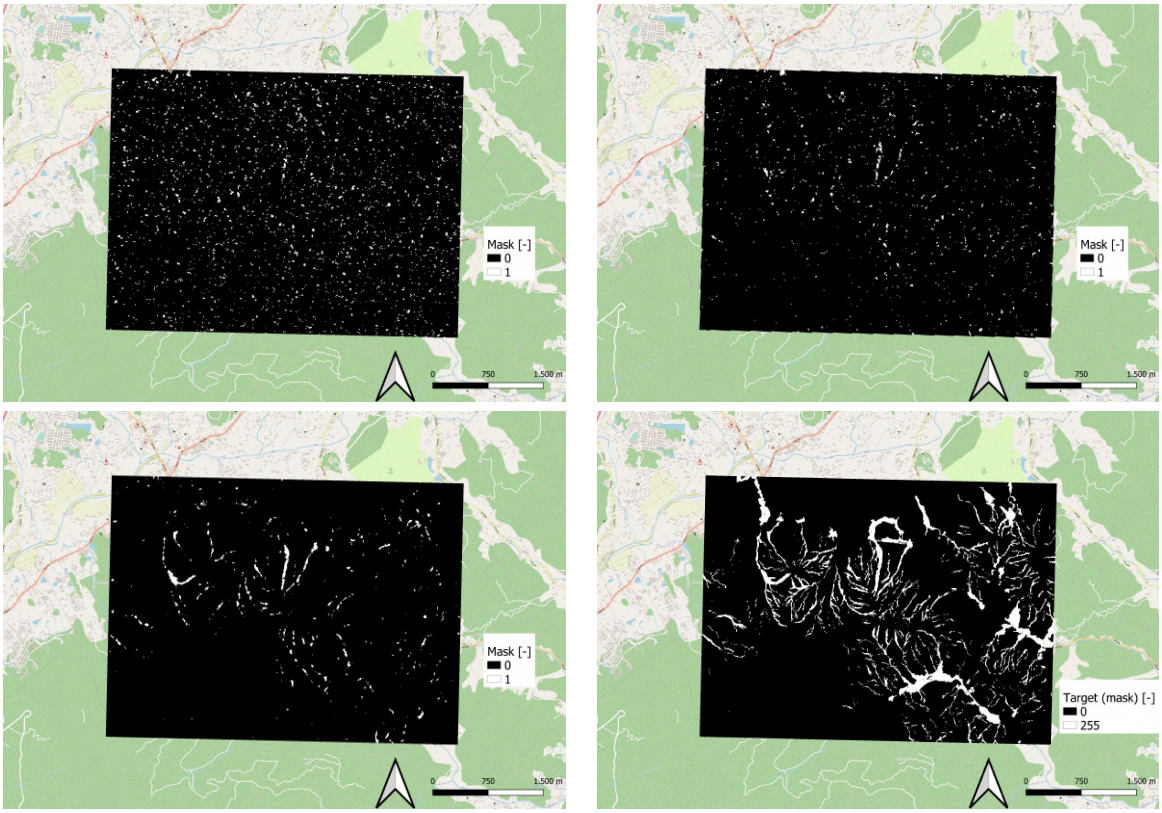

Figure 4.6: Masked form of A_{ratio} images, where a threshold of A_{ratio} bigger than 2.5 is taken. TLR: masked A_{ratio} without temporal averaging, with temporal averaging of one month, with temporal averaging of two years, ground truth target image.

Figure 4.7 shows the ampslope and ratioslope representations. The landslides cannot be clearly detected from this. This is partly because the slope image band is very dominant in the images compared to the SAR amplitude band(s). By scaling of the pixel values of the bands, as described in section 3.3.3, this dominance is lessened. The other part is since the other bands, the pre- and post SAR amplitude band as seen in figure 4.3 of the ampslope and the A_{ratio} band as seen in figure 4.5 of the ratioslope representation, are both limited in their ability to detect landslides.

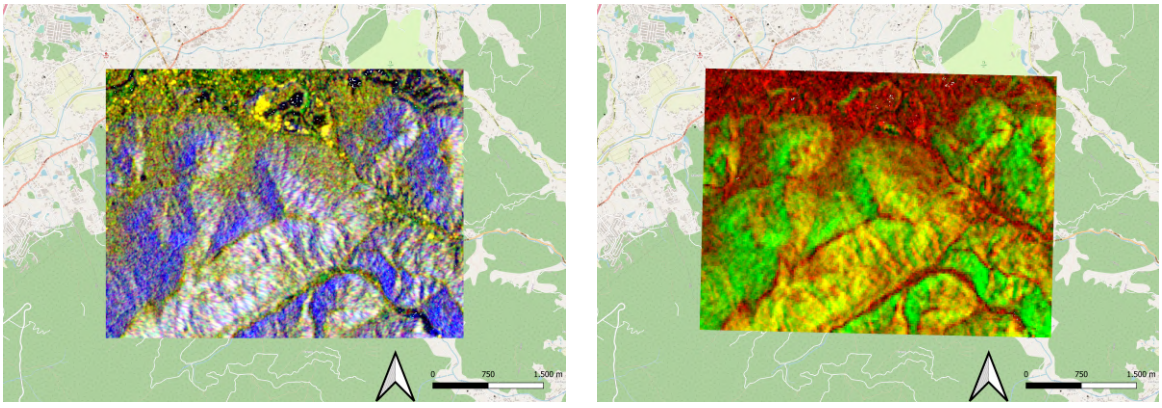

Figure 4.7: The ampslope representation (left) and ratioslope representation (right).

Section 3.3.2 describes the method of rasterizing the larger S-1 satellite images into smaller tiles. This is necessary since the DL models cannot train on very large images. This does change the scale of the landslides in the image and the amount of landslides in one image. It could be possible that landslides in an area are more

difficult to recognize when said area is cut into tiles. This can be seen in figure 4.8, where part of the area with landslides of the previous figures in this section is captured into one smaller tile. For this tile the landslides seem to still be represented quite well. However, for other tiles it might be that a landslide is cut into half by rasterization, which makes it more difficult to interpret and detect. This is the type of image and size, on which the U-Net and Pix2Pix cGAN are trained on.

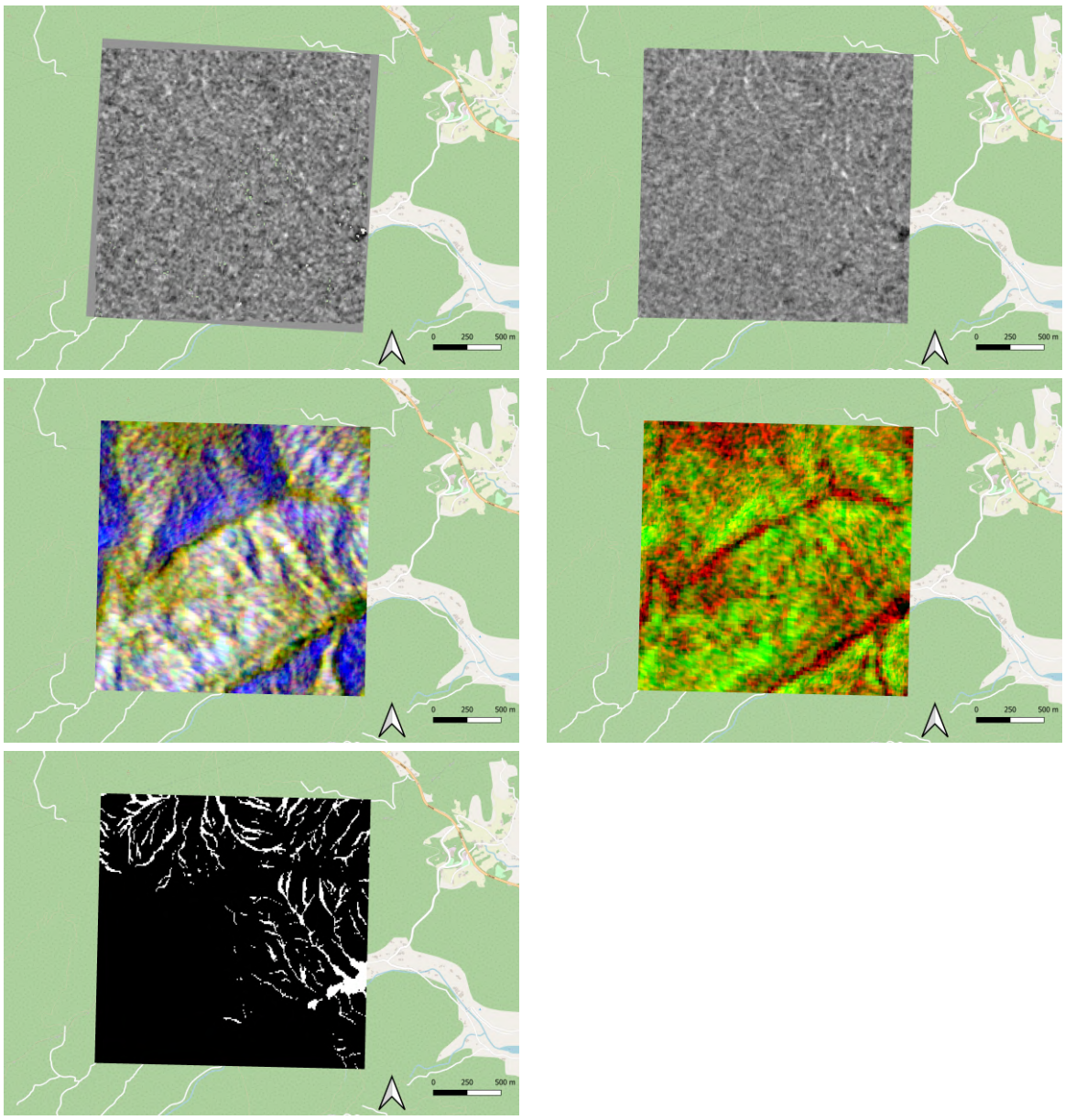

Figure 4.8: Tile-sized images, with the shape and size of which the images are trained in the U-Net and cGAN. RTL: The A_{ratio} without temporal averaging, A_{ratio} with temporal averaging of one month, the ampslope, the ratioslope and the target image.

4.2 TESTS WITH A_{ratio} REPRESENTATION

The A_{ratio} representation is explained in section 3.2.3 and some visualizations of it are also visible in section 4.1. The images of ascending and descending direction are averaged, which reduces geometric effects [28]. This is discussed in both sec-

tions 4.1, 3.2.3 and 2.2.1. Training for this representation has only been done on one training data set, which contains both landslide and nonlandslide images. The amount of images can be found in table 3.3. The ratio of landslide/nonlandslide images is more or less 1, so there is no class imbalance. Therefore the binary cross entropy loss mixed with dice loss, as explained in section 2.4.5, is used in the tests for this representation. Various batch sizes have been tested: 8, 16,32 and 64.

Figure 4.9 shows the learning curve based on the loss functions of the cGAN. The learning curve showing the loss shows three lines. The 'gen' line is the loss of the generator, the 'd-real' is the loss of the discriminator with regards to the real targets and the 'd-fake' is the loss of the discriminator with regards to the generated fake targets. The learning curves of the loss function are not similar to the ideal example of a cGAN as seen in figure 2.16. The discriminator and generator loss are expected to start off erratic and converge to a stable equilibrium [10]. However, figure 4.9 shows that both discriminator loss curves remain close to zero, which means the discriminator is not learning (a lot). This is most likely since the generator is not able to generate good images, in this case target images which are similar to the true target images. Hence the discriminator can recognize the fake/generated images too easily and therefore also the true images. This means the discriminator is not learning, and therefore the generator in turn is also not learning from the discriminator. The learning curves containing these losses show a convergence failure, since the generator and discriminator losses do not converge to a stable equilibrium. This makes it difficult to find the best fit among the epochs. Most likely there is no good fit among the epochs.

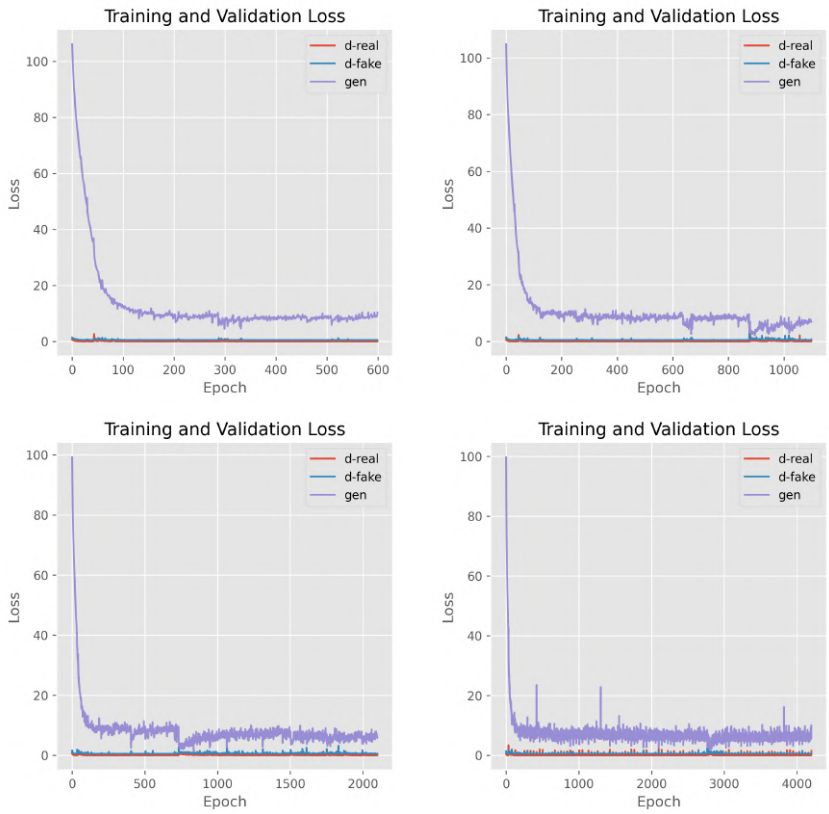

Figure 4.9: Learning curves showing the loss for the A_{ratio} representation from the cGAN. The 'gen' line is the loss of the generator, the 'd-real' is the loss of the discriminator with regards to the real targets and the 'd-fake' is the loss of the discriminator with regards to the generated fake targets. TLR: The graphs are made with different batch sizes: 64,32,16 and 8 respectively. The x-axis shows the amount of tests, e.g. the amount of epochs * the amount of batches per epoch. The amount of tests is therefore higher with a smaller batch size. The y-axis shows the loss value.

Figure 4.10 shows the learning curves based on the mIoU by the cGAN. The mIoU is the average of the IoU, which is calculated by equation 2.9.2, over all the images in one epoch in this case. A threshold was taken into account in this case, since the outputted values were in the range between -1 and 1. So a threshold of 0 was taken, to create a binary pixel classification. Pixels above the threshold were classified as landslide pixels, and below the threshold as non landslide pixels. Then based on these binary classified pixels the IoU and subsequently the mIoU were calculated.

Some large decreases in mIoU of both validation and training can be seen in the various plots. It is unclear what causes these decreases. In order to understand the output of the model better, the generated target images are also plotted. Ideally the epoch which gives the best fit of the model is chosen based on the learning curves. Since that is not clear from these learning curves, the last epoch is chosen. Hence the source, target and generated target images are visualized for the trained model from epoch 100.

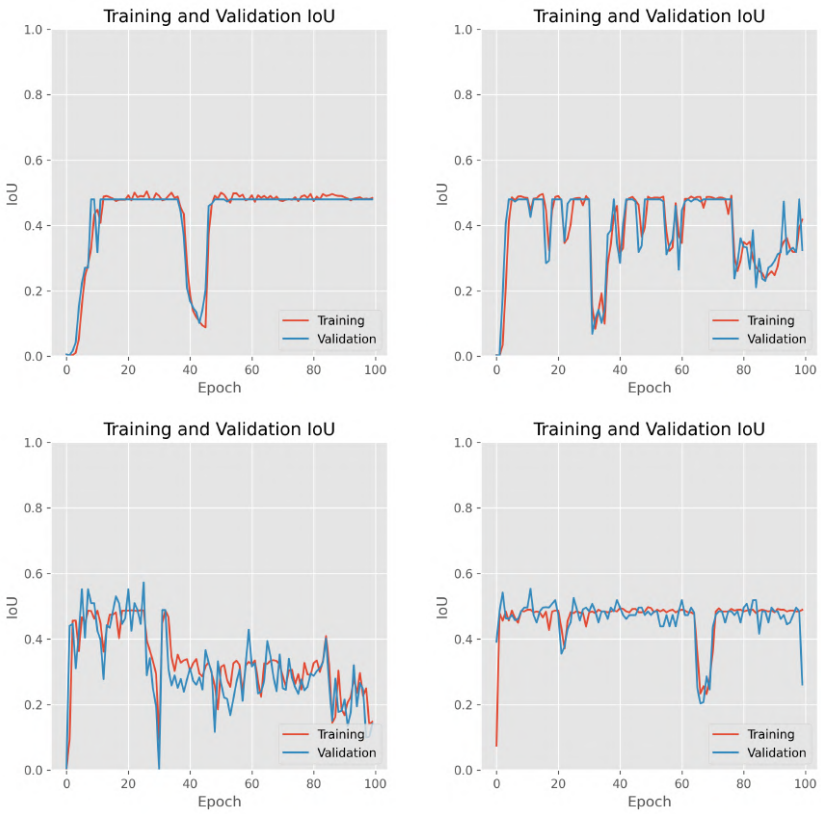

Figure 4.10: Learning curves showing the mean Intersection over Union (mIoU) for the A_{ratio} representation from the cGAN. TLR: batch sizes of 64,32,16 and 8 respectively.

In general the mIoU is not larger than 0.5 for any of the epochs. An mIoU of 0.5 means that half of the pixels are accurately predicted. Considering that roughly half of the images are nonlandslide images, this does not necessarily indicate that many landslide pixels are correctly classified. Since when a target image no landslide pixels, e.g. only landslide pixels, it is easy to get a perfect IoU score of 1. Then by getting the mIoU, the lower IoU values of the landslide images are averaged to a mIoU of 1. This is confirmed by figure 4.11, which shows the mIoU for only the landslide images. This mIoU is below 0.01.

Figure 4.11: Learning curve showing the mean Intersection over Union (mIoU) for the A_{ratio} representation from the cGAN. The IoU values which contribute to the mIoU are only calculated on landslide images. The batch size for this test was 64.

Figure4.12 shows the loss and IoU curve of the U-Net. Similar to that of the cGAN the loss and mIoU is low.

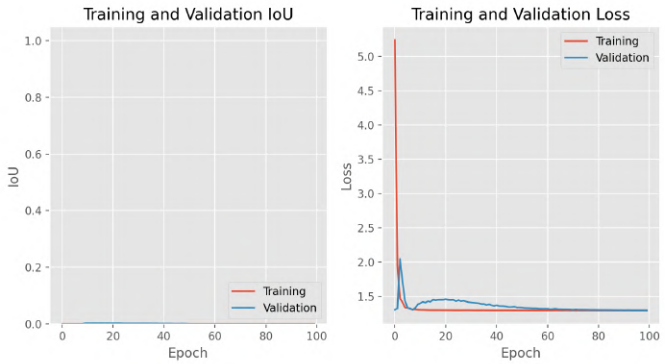

Figure 4.12: Learning curve showing the loss and the Mean Intersection over Union (mIoU) for the A_{ratio} representation from the U-Net. The batch size for this test was 64.

Figure 4.13 shows a random subset of the source, generated target and target images during training,with as input the A_{ratio} source images and the corresponding target images. The mask is based on the slope of the Digital Elevation Model (DEM) satellite image. Pixels which correspond to a slope smaller than or equal to 0.5 degrees are filtered. It can be seen that the generated target is not an accurate representation of the true target, so the generator does not succeed in generating accurate target images.

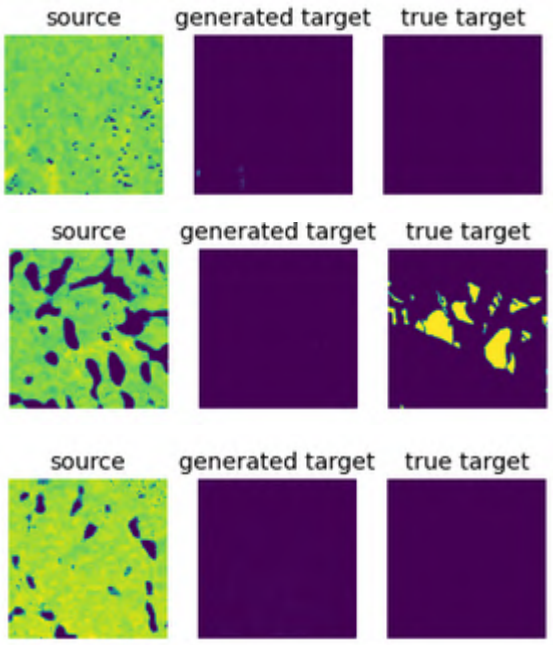

Figure 4.13: Source, target and generated target images for the A_{ratio} representation produced by the cGAN. The batch size of this test was 8.

Figure 4.14 shows a subset of true target and predicted target images, both from the testing data set. The testing data set is unseen data which is not used as a training set by the model before. The predicted targets were not accurate. Some targets which do not contain any landslides were accurately predicted, while other

predicted targets contain landslide pixels which were not there. The landslides were neither correctly mapped nor predicted.

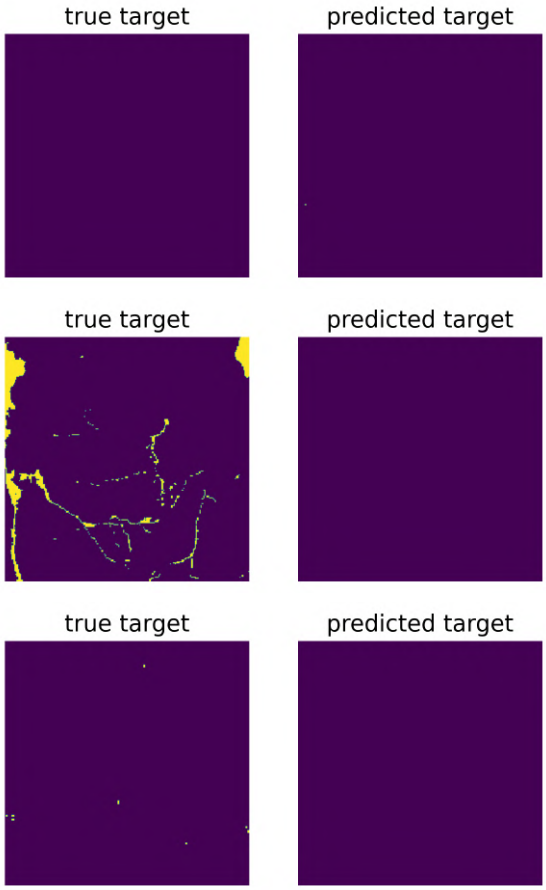

Figure 4.14: True target and predicted target from testing data set of the A_{ratio} representations, with batch size 64.

It could be that the A_{ratio} band still contains too much speckle and geometric effects. Averaging over a month for the pre- event stack of images and post-event stack of images might be too little to take away a sufficient amount of speckle. Sufficient in this context meaning the ability of this model to understand the A_{ratio} images and detect landslides from them. One could argue that the inability to detect landslides lies with the model and not with the data. Figure 3.14 shows that taking a month to average the pre- and post-event stack of images over, before taking the difference resulting in the A_{ratio}, does not result in landslides which can be recognized 'easily'. Only when averaging over more time, e.g. two years, landslides can be recognized. And even then there is still noise and not everywhere the positive A_{ratio} corresponds to a landslide.

4.3 TESTS WITH RATIOSLOPE REPRESENTATION

The ratioslope representation is explained in section 3.2.4 and some visualizations of it are also visible in section 4.1. For the first band of the ratioslope representation, the A_{ratio}, the images of ascending and descending direction are averaged, which reduces geometric effects [28]. Training for this representation has only been done on one training data set, which contains both landslide and nonlandslide images. The amount of images can be found in table 3.3. The ratio of landslide/nonlandslide images is more or less 1, so there is no class imbalance. Therefore the binary

cross entropy loss mixed with dice loss, as explained in section 2.4.5, is used in the tests for this representation.

Figure 4.15 shows a random subset of the source, generated target and target images during training, with as input the ratioslope source images and the corresponding target images. These images were generated during training of the cGAN. The generated target only has 'not a number ' (NaN) values, hence there are no generated images. The generator was not able to generate images based on the training set, so the source images could not be translate into target images. Therefore the cGAN model in combination with this input data (see 3.2.4) is neither able to detect nor segment landslides. Further results such as the learning curves and predicted images on the testing sets were not made, since the generator is unable to generate target images based on the training set. Also there was no training done on the U-Net, since the cGAN did not work and the U-Net has little chance of giving a btter result. This is since the U-Net also did not give a better result than the cGAN with the A_{ratio} representation.

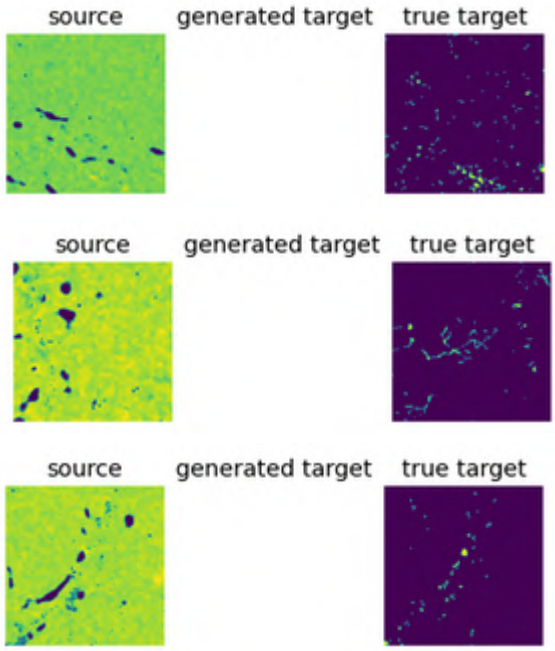

Figure 4.15: Source, target and generated target images for the ratioslope representations. The generated target images only contained 'not a number' (NaN) values, therefore these are not visualized. The batch size of this test was 64.

The reason for the inability of the generator to generate a target image might be that the algorithm does not understand the relationship between the A_{ratio} band and the slope band. On top of that, as mentioned in the for the A_{ratio} images in section 4.2, it could also be that the A_{ratio} band contains too much speckle still. Averaging over a month for the pre event stack of images and post-event stack of images might be too little to take away a sufficient amount of speckle. Sufficient in this context meaning the ability of this model to understand the A_{ratio} images and detect landslides from them. Furthermore, the algorithm might not work well with a 2-banded image. Traditionally either single band image or three banded images are used as input in algorithms such as the cGAN [34]. It might be that if a third band was added to the source images, such as a band depicting the aspect of the DEM or just a dummy band containing zero values, the generator would be able to generate a target. This target image would likely have a low accuracy as well, just

like the resulting target images from section 4.2.

4.4 TESTS WITH AMPSLOPE REPRESENTATION

The ampslope representation is explained in section 3.2.5 and some visualizations of it are also visible in section 4.1. Multiple variations of this ampslope representation have been used to train the U-Net and cGAN, but only of a few the results are displayed in this section. For the results visualized in this section, the training data set was filtered on relatively 'large' landslides, so only images with relatively 'large' landslides were kept. The minimum size of these 'large' landslides per landslide database are stated in the Appendix. The validation and testing data sets do contain all landslide images though, including smaller landslides. For these two data sets nonlandslide images were filtered. For this training dataset, three variations are used. One contains only images of descending direction, the other only of ascending direction. The third contains both training sets, so descending and ascending mixed together. Since there is no class imbalance in this case, the binary cross entropy loss mixed with dice loss was used, as explained in section 2.4.5.

Other tests with different training data sets and parameters but the same ampslope representation have been run as well. One training data set contains all landslides, also smaller ones. This has been tested with the same parameters as the one with only large landslides, and with the various orbit directions. Additionally there have been tests with nonlandslide images and landslide images together. Various resampling methods have been tried. One used oversampling on the landslide images (the minority class) combined with data augmentation, such that the oversampled images were not duplicates. There was also a test with this same training set but without resampling. Instead the loss function focal dice loss was used, since this counteracts class imbalance (see section 2.4.5). All these tests have similar results as the results which will be shown in this section.

Figure 4.16 shows the learning curve based on the loss functions of the cGAN. Three graphs are plotted, one is trained on a training set with images of ascending direction. The other is trained on images of descending direction. The last is trained on all images, so both images of descending and ascending order mixed together. The learning curve showing the loss shows three lines. The 'gen' line is the loss of the generator, the 'd-real' shows the loss of the discriminator with regards to the real targets and the 'd-fake' shows the loss of the discriminator with regards to the generated fake targets. Both discriminator loss curves remain close to zero, which means the discriminator is not learning properly. This is most likely since the generator is not able to generate good images, in this case target images which are similar to the true target images. Hence the discriminator can recognize the fake/generated images too easily and therefore also the true images. This means the discriminator is not learning, and therefore the generator in turn is also not learning from the discriminator. The learning curves containing these losses show a convergence failure, since the generator and discriminator losses do not converge to a stable equilibrium. This makes it difficult to find the best fit among the epochs. Most likely there is no good fit among the epochs. The loss curves shows similar behaviour as described in both 4.2.

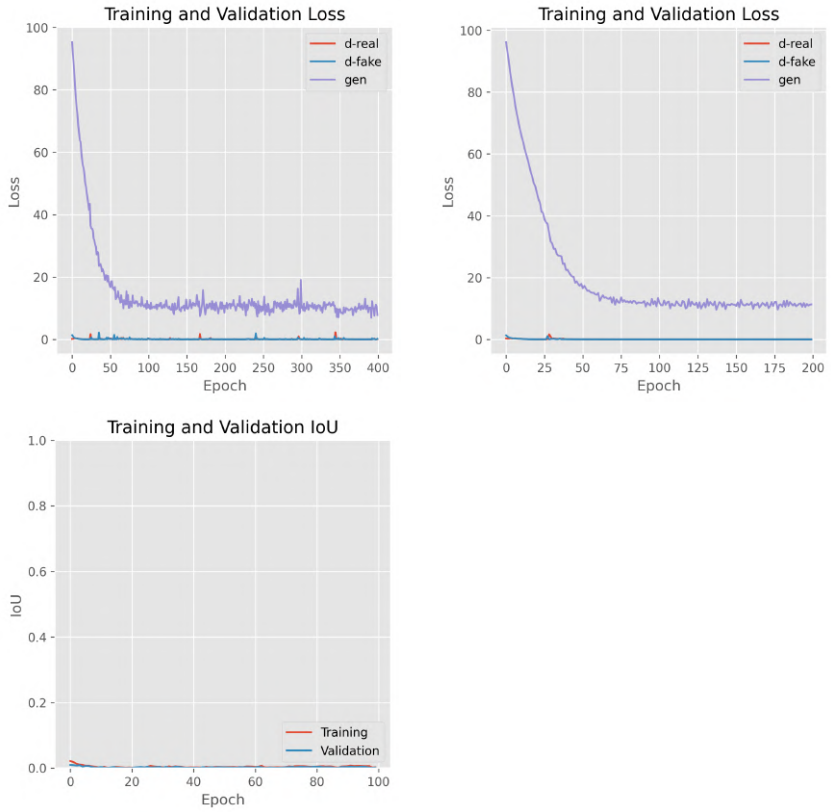

Figure 4.16: Learning curves showing the loss for the ampslope representation featuring large landslides only from the cGAN. The 'gen' line is the loss of the generator, the 'd-real' shows the loss of the discriminator with regards to the real targets and the 'd-fake' shows the loss of the discriminator with regards to the generated fake targets. The top figure shows the learning curve for the data set of ascending direction, the middle for that of descending direction, and the last a combination of the two. The batch size of these models are 64. The x-axis shows the amount of tests, e.g. the amount of epochs * the amount of batches per epoch. The y-axis shows the loss value.

Figure 4.17 shows the learning curves based on the mIoU from the cGAN. Again three graphs are shown, which correspond to the learning curves with the loss metrics. One is trained on a training set with images of ascending direction, the other on images of descending direction and the last is trained on all images, so both images of descending and ascending order mixed together. The mIoU was calculated in the same way as described in 4.2. The mIoU values of these learning curves are all low, below 0.05. This means that very little pixels are classified correctly. Again, from the learning curves it is difficult to find an epoch which shows the best fit. Therefore the source, target and generated target images are visualized for the trained model from epoch 100.

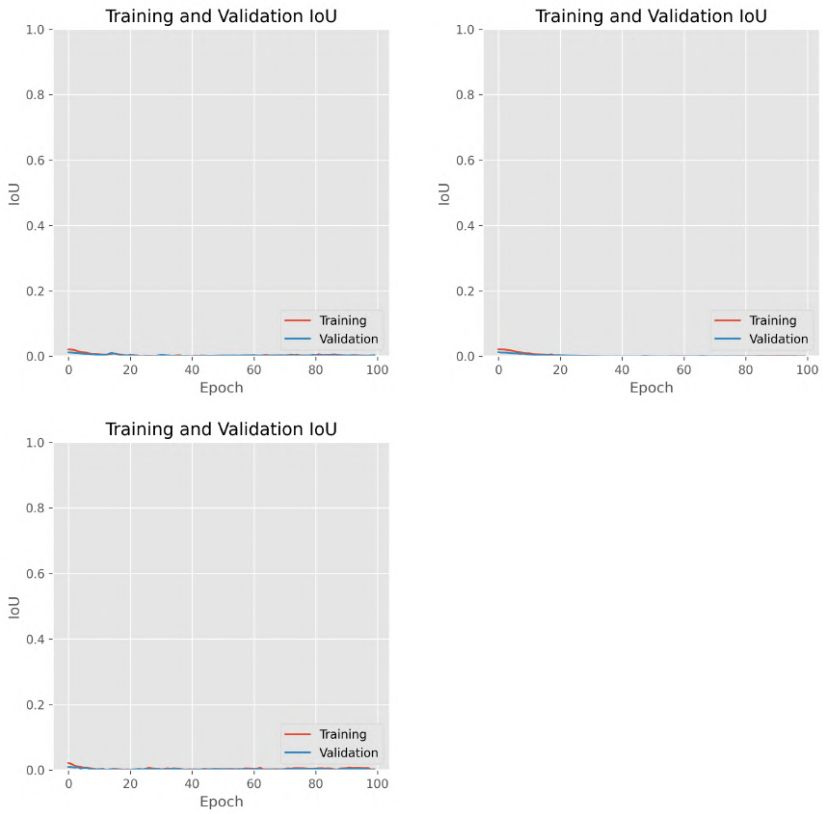

Figure 4.17: Learning curves showing the mean Intersection over Union mIoU for the ampslope representation featuring large landslides only from the cGAN. The top figure shows the learning curve for the data set of ascending direction, the middle for that of descending direction, and the last a combination of the two.

Figure4.18 shows the loss and mIoU learning curves of the U-Net. Again three graphs are shown, which correspond to the learning curves of the cGAN. Similar to that of the cGAN the loss and mIoU values are low.

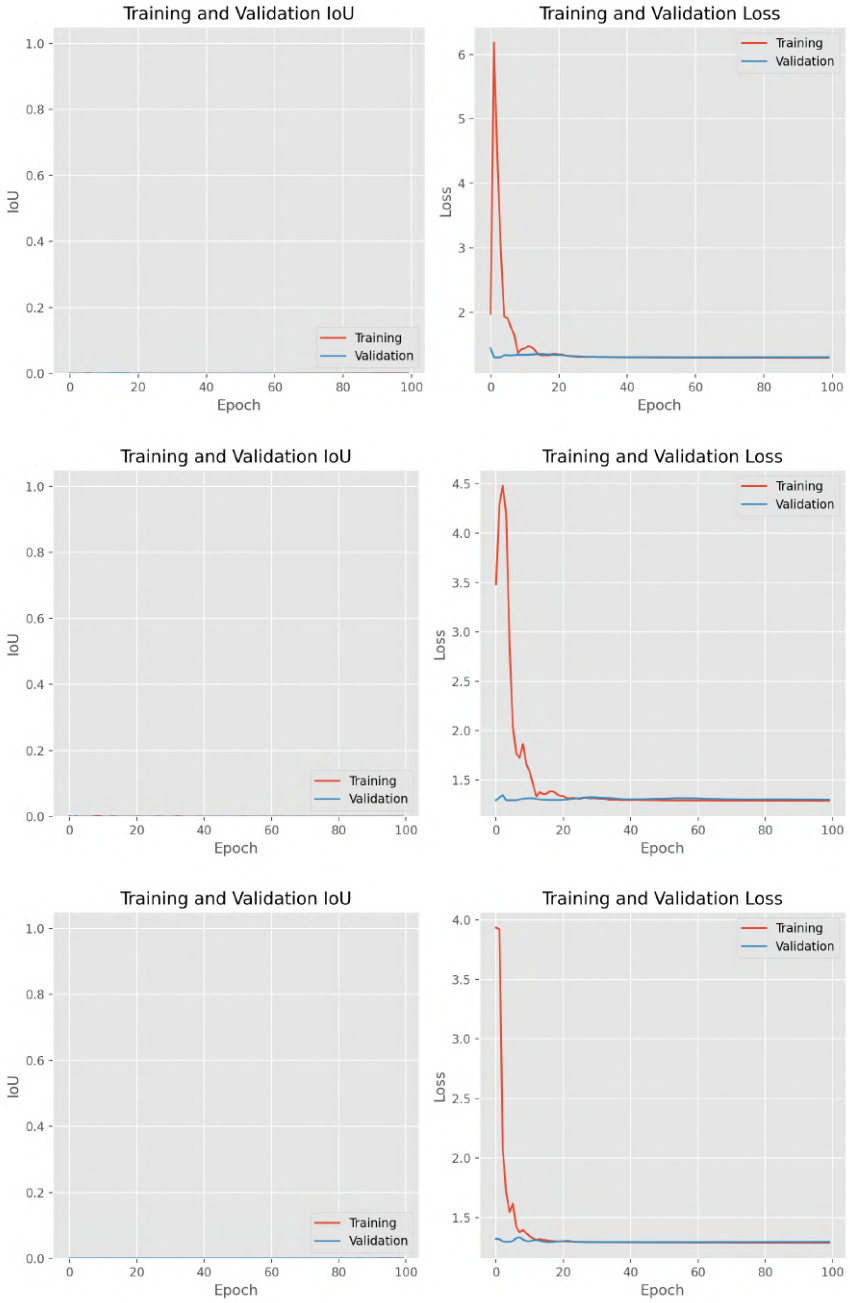

Figure 4.18: Learning curves showing the loss and mean Intersection over Union mIoU for the ampslope representation featuring large landslides only from the U-Net. The top figure shows the learning curve for the data set of ascending direction, the middle for that of descending direction, and the last a combination of the two.

Figure 4.19 shows a random subset of the source, generated target and target images during training, with as input the ampslope source images and the corresponding target images. Shown here are only the images of ascending direction, the rest is shown in the Appendix. The figures show that the generated target is not an accurate representation of the true target, so the generator does not succeed in generating accurate target images.

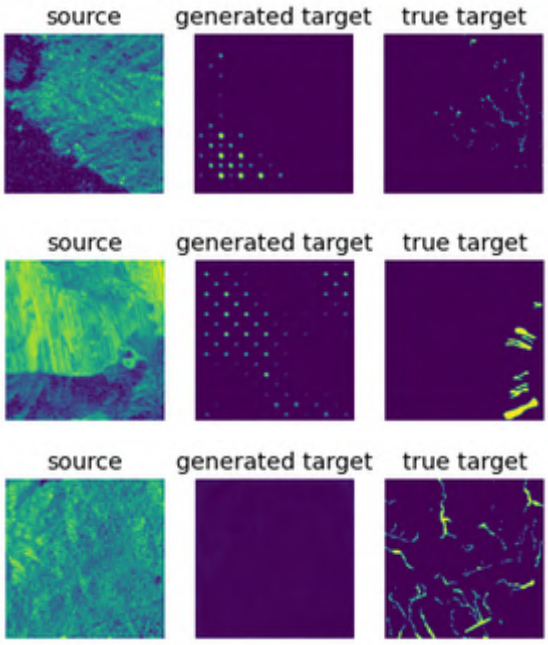

Figure 4.19: Source, target and generated target images for the ampslope representation featuring large landslides only of ascending direction from the cGAN, with batch size 32.

Figure 4.20 shows a subset of true target and predicted target images, both from the testing data set. The testing data set is unseen data which is not used as a training set by the model before. The predicted targets were not accurate. Some targets which do not contain any landslides were accurately predicted. The landslides were neither correctly mapped nor predicted. Some landslides were mapped on other places than where they should be, so these were falsely positive classified pixels.

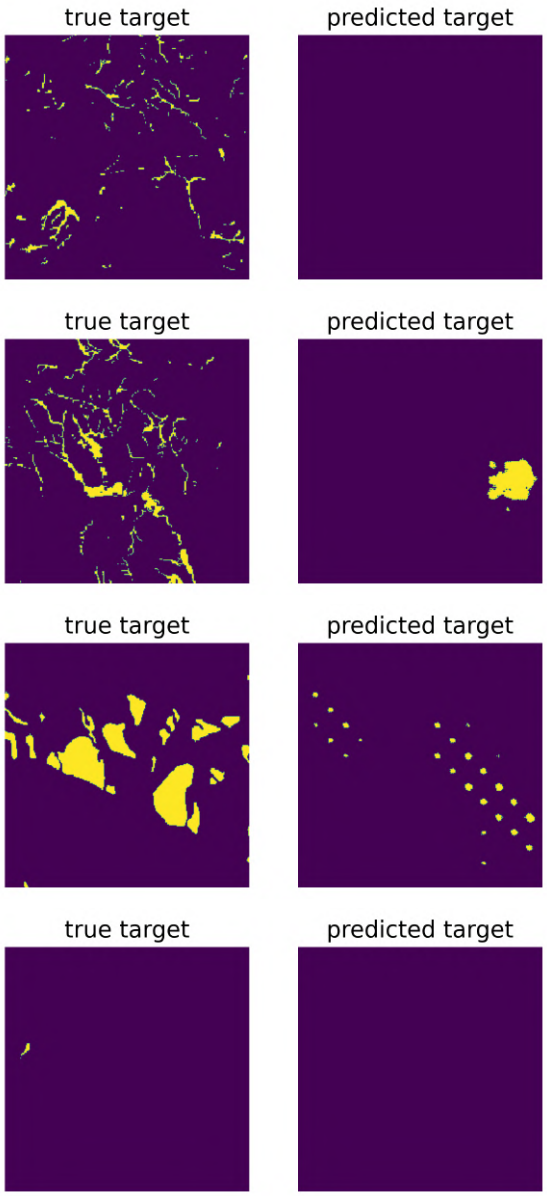

Figure 4.20: True target and predicted target from testing data set of the ampslope representation featuring large landslides only of ascending direction from the cGAN,with batch size 32.

The model is not able to detect and map landslides based on these ampslope source and target images containing only 'large' landslides. This is probably since the model does not understand the ampslope image. The SAR amplitude images were not denoised in any way by masking or temporal averaging. The speckle, which decreases the quality of the SAR images, could be a reason for the failure of the model to detect landslides. Furthermore the A_{ratio} was not taken, but the SAR images pre- and post-event were given as two bands in a three band image. It could be that the model does not understand such an image, e.g. it should be presented as an A_{ratio} image instead.

4.5 DISCUSSION

Both the U-Net and Pix2Pix cGAN did not succeed in detection and segmentation of the landslides. Since the models both run well on other imagery, as described in sections 2.6 and 2.7 respectively, the models seem to be valid. Furthermore with the various data used in this study, many tests have been run on both models. Not all adjustable hyperparameters of these models have been tuned, only the batch size, loss function and amount of epochs. Nevertheless the amount of tests and experiments done in this study are extensive. The A_{ratio} and ampslope representations both did not return any mIoU value above 0.01. Since this returned value is so low, it can be stated that all tests from both models completely fail to detect any landslide. Therefore the tuning of any other parameter or the introduction of another loss function are not expected to improve the results by that large a margin, e.g. from 0.01 to 0.6. The models trained on the ratioslope representation do not succeed in landslide detection and mapping. For these images, the cGAN was not even able to generate a target image based on the training set. Therefore nonlandslide images could also not be recognized from this data. Thus it can be concluded that the representations in combination with the models used in this study are not capable of detecting and mapping landslides.

Hence the cause of the failure is the input data, the SAR satellite imagery, itself. The speckle and geometric effects of the ampslope representations have not been decreased in any way, by neither masks nor temporal averaging. The difference between the pre-event image and post-event image was not taken, as is traditionally done by calculating the A_{ratio}. Instead these two image were used as bands, along with the slope image based on the DEM resulting in a three-band image. The goal was for the model to understand the difference between the first two bands is the correlation for a possible landslide/change detection. By looking at figure 4.3, it is quite understandable that from the difference between the two SAR bands it is difficult to detect a landslide. This is supported by figure 4.6, where from the A_{ratio} of the pre- and post-event images which make up the first two bands of the ampslope image landslides cannot be detected. This is since the pre- and post-images have too much speckle and geometric effects. It is also not sure if the models understand the correlation of the slope band and the landslides. Furthermore it is not sure if the model understands the relation between the pre and post-image, when given as two bands instead of as A_{ratio} image.

There are several options to change the ampslope representation, such that it might be able to detect landslides. Firstly, a larger timestep between the two images making up the first two bands could be done. The difference between the two images might be larger then, such that a change can be detected more easily. However, by increasing the timestep the chance of change caused by something other besides landslides increases. Also a larger timestep in between two single images will not make those images less noisy. Furthermore if a larger timestep is taken anyways, then temporal averaging over that time might as well be done. The total time spent from the pre-event image to the post-event image, while using a larger timestep between the two, will not increase when temporal averaging is done over that time. This is contrary to when the timestep of the ampslope image is taken as a consecutive timestep, as is done in this study and failed to result in landslide detection. So the first option, by just increasing the timestep without temporal averaging, will probably fail. Secondly, the first and second band of the ampslope representation could be temporally averaged pre-event image and post-event image instead of single pre-event and post-event images. The difference between this setup and A_{ratio} would be that the difference between this images is not taken, but that they are used as bands of the ampslope image. When temporal averaging is done on these two bands, the speckle will be reduced. Then the algorithm might

be able to understand the correlation between the bands and the occurrence of a landslide. Thirdly, more images can be produced by using multiple combinations of pre- and post-event images making up the first two bands of the ampslope image.

For the A_{ratio} the highest mIoU was 0.01, this did not result in accurate landslide detection and mapping. This is confirmed by figure 4.6, where the A_{ratio} averaged over one month showed limited detectable landslide scars. Handwerger et al. [28] found by using A_{ratio} with averaging pre- and post-event images of the Hiroshima database over one month, the true positive rate (TPR,see section 2.9.2) was 0.6350 by taking a threshold of the positive A_{ratio}. This threshold method, a lot simpler and faster than DL methods such as U-Net and cGAN as used in this study, is visualized in figure 4.6. The TPR is higher with more temporal averaging, as also confirmed by figure 4.6. More temporal averaging, especially of the post-event images, than two or three months is impractical and not desirable. Also during this time changes unrelated to landslides might have happened, which oppose the landslide detection. To do this research on these other possible changes must be done then for the area. This makes it difficult to generalize the model to multiple regions, since these possible changes differ per region. After two to three months the landslide might be overgrown with vegetation, and the information comes to late for emergency response and disaster mitigation. If the date of the event is the query, then a moving average can be applied. Furthermore the main reason to use SAR over optical imagery is the limited usability of the latter, due to its dependency on cloud free weather and light. If in need to use SAR temporal averaging of three months or more is necessary, the advantage over optical imagery vanishes. In that sense, it is more suitable to use optical imagery since it is easier to process and more intuitive. Also the need for temporal averaging and denoising of optical imagery is limited, as can be seen from figure 3.10.

The threshold method from Handwerger et al. [28] had better results for the A_{ratio} than the U-Net and cGAN models in this study. This cannot be directly compared since Handwerger et al. [28] only used the Hiroshima landslide database, where in this study in addition to that database three other databases were used. Additionally Handwerger et al. [28] removed small landslides with area of under 100 square meters prior to classification, which is not done in this study. For A_ratio images averaged over one month the TPR was 0.6350, over three months 0.7248 and over multiple years 0.8372 according to the threshold method used in Handwerger et al. [28]. So further research into the amount of temporal averaging necessary and the most suitable model to use in case SAR amplitude imagery is preferred for landslide detection is still needed.

5 CONCLUSION

In this chapter, the study will be summarized and an answer will be given to the main research- and sub questions. Furthermore recommendations for further research will be given.

- *How do landslides appear in Synthetic Aperture Radar (SAR) amplitude imagery?*

  Single SAR amplitude images do not clearly present landslides. The SAR amplitude is difficult to interpret due to speckle and geometric effects. Thereby landslides cannot be recognized from a single SAR amplitude image after a landslide event, in contrast to a Copernicus Sentinel-2 optical image after a landslide event. This is discussed in section 4.1.

- *How can landslides be represented by a combination of SAR amplitude imagery and other (satellite) data?*

  Three representations have been tried in this study which consist of a combination of SAR amplitude imagery and slope data derived from a Digital Elevation Map (DEM). Landslides generally occur on areas with a slope between 12 and 35 degrees, so they are dependent on slope. The necessity for speckle reduction by temporal averaging was researched by using three representations with and without temporal averaging. The first representation is a one-banded image featuring the log-based amplitude difference, here referred to as A_{ratio}. Images covering a period of one month pre- and post-event are averaged and subsequently the log-based amplitude ratio is taken. The resulting ratio image undergoes filtering by a slope mask, with a threshold of 0.5 degree slope. The second representation is a two-banded image, where the first band is previously used A_{ratio} image, referred to as ratioslope representation. The second band is the previously mentioned slope image. The third and final representation is a novel method from this study referred to as ampslope representation. This ampslope representation is a three-banded image, where the first image is a single pre event SAR amplitude image, and the second band is a single post-event SAR amplitude image and the third band is the previously mentioned slope image. In this way, data processing is limited and temporal averaging is not used. The pre- and post-event images of the ampslope image are the last and first available before and after the event respectively.

- *How can landslides be detected from suitable SAR amplitude image representation?*

  Landslides can be detected by using the log-based amplitude ratio, so-called A_{ratio}, with temporal averaging of the pre- and post-event SAR amplitude images. More temporal averaging increases the quality of the image by reducing the speckle. The combination of images of descending and ascending direction by averaging further reduces speckle and geometric effects. Therefore with more temporal averaging, the landslides can be detected better. This can be seen in figures 4.3,4.5 and 4.6. The positive A_{ratio} mostly corresponds to landslide scars. The DL models used in this study, a U-net and a Pix2Pix cGAN have not been able to detect landslides with A_{ratio} images. By taking the positive A_{ratio} with a threshold, as done in Handwerger et al. [28] and as shown in figures 4.5 and 4.6, landslide detection can be done.

- *How can success of landslide detection and mapping be evaluated?*

  Landslide detection classifies pixels which are landslides or not, and returns this classification. Landslide mapping refers to segmentation of the landslides, by which the segmented landslide object or polygon is returned. The latter is returned by two Deep Learning (DL) models used in this study, a U-Net and a conditional Generative Adversarial Network (cGAN), which are trained on the various SAR amplitude image representations. Intersection over Union (IoU) is an object-based metric which returns how well the segmentation of the landslide polygons is done. During training of the model the mean Intersection over Union mIoU of the images is returned. Additionally the loss metric of the model, which indicates how well the model fits to the data, during training is returned. The mIoU and loss are captured in a learning curve, which gives an indication of the quality of the fit of the model to the data during training. After training, images can be predicted based on a testing data set which is not used as a training data set during training. This shows how the model can predict unknown data, the predicted images can be visually checked. The IoU and mIoU of the predicted images can be calculated as well.

These sub questions lead to the main research question :

How can landslide detection and mapping be done on SAR amplitude satellite imagery?

Both models, U-net and Pix2Pix cGAN, did not succeed in accurate landslide detection and segmentation. For both the U-Net and the cGAN model, the mIoU of all three representations during various tests was less than 0.01. The predicted target images correspond to these low IoU values, since the predicted landslides are not accurate.

Landslides can be detected by use of human eye or via a threshold when using the log-based amplitude ratio, A_{ratio}, representation with temporal averaging. The detection becomes better with more temporal averaging and by averaging over images of ascending and descending direction. The DL models used in this study, are not suitable for detection of landslides in combination with the representations used. In theory the DL method should be better than the threshold method, but this cannot be concluded from the results of this study. Therefore a threshold method in combination with A_{ratio} of which images of ascending and descending direction are averaged and temporal averaging is done of at least one month. This A_{ratio} must be computed by using a moving average, such that the event time can be found.

5.1 RECOMMENDATIONS

In conclusion, if (relatively) cloud-free optical imagery is available within three months after a landslide event, then that is to be preferred over SAR amplitude imagery. This is since SAR amplitude imagery needs speckle reduction in the form of temporal averaging of at least more than a month. This introduces the need for a moving average which can be a lot of work, especially when different regions and events are taken into account. In this study it is found that optical imagery shows landslides more clearly than SAR amplitude data, as visualized in figure 4.2, and less processing is needed for it. Additionally, for averaging over one month's time the results of both DL models used in this study and the threshold model used in Handwerger et al. [28] was not higher than that of landslide detection studies with optical data [14; 67; 11].

Another possibility is to use polarimetric synthetic aperture radar PolSAR [61; 66] and interferometric synthetic aperture radar InSAR [40] instead of amplitude SAR products. In this study these methods have not been explored, due to the costly processing and limited availability of this data. However, these methods do have the advantage of SAR data over optical data, in the sense that they are usable during bad weather and at night. But they have the advantage over SAR amplitude data that they have a larger correlation with change detection than amplitude data does.

The combination of SAR and optical satellite imagery is worth looking into. Since these two techniques are in some ways opposites from each other, they also compliment each other. The advantages of optical satellite imagery are its spectral resolution which makes change detection relatively easy compared to SAR. Also it does not suffer from speckle and geometric effects like in SAR. The disadvantage is its is dependent on light and cannot pentrate clouds, such that is only usable in (relatively) cloud-free weather. For SAR this is almost the complete opposite, since it can be used during bad weather and at night, but it is difficult to work with due to speckle and geometric effects. Therefore these to techniques could be seen as complimentary, little studies have taken advantage of this. Studies which have used this combination have mainly use the fusion of these two data types, where heterogeneous data is transformed into the same feature space so that it becomes comparable [23; 52; 26; 64; 41; 48; 30; 42]. Only Benedetti et al. [5] has used heterogeneous data, in this case SAR and optical satellite data (Sentinel-1 and -2 resp.), without fusing them. They used a separate CNN for each type of data and then combined the output by a weighted average. Both could be a possibility to use in combination with cGAN or U-Net, possibly using little to no temporal averaging on the images. This is worth exploring further, since it has the potential to overcome the speckle and geometric effect of SAR imagery while using limited to no temporal averaging.

BIBLIOGRAPHY

[1] Adriano, B., Yokoya, N., Miura, H., Matsuoka, M., and Koshimura, S. (2020). A semiautomatic pixel-object method for detecting landslides using multitemporal ALOS-2 intensity images. *Remote Sens.*, 12(3).

[2] Al-Sharif, A. A., Pradhan, B., Hadi, S. J., and Mola, N. (2012). Revisiting Methods and Potentials of SAR Change Detection.

[3] An, Y., Long, J., and Mabu, S. (2020). A segmentation network with multiattention and its application to SAR image analysis. *IEEJ Trans. Electr. Electron. Eng.*, 15(4):570–576.

[4] Anonymous (2019). Viewing Geometry and Spatial Resolution. https://www.nrcan.gc.ca/maps-tools-publications/satellite-imagery-air-photos/remote-sensing-tutorials/microwave-remote-sensing/viewing-geometry-and-spatial-resolution/9341.

[5] Benedetti, A., Picchiani, M., and Frate, F. D. (2018). Sentinel-1 and sentinel-2 data fusion for urban change detection. In *Int. Geosci. Remote Sens. Symp.*, volume 2018-July.

[6] Bressan, P. O., Junior, J. M., Martins, J. A. C., Gonçalves, D. N., Freitas, D. M., Osco, L. P., de Andrade Silva, J., Luo, Z., Li, J., Garcia, R. C., and Gonçalves, W. N. (2021). Semantic Segmentation with Labeling Uncertainty and Class Imbalance.

[7] Brownlee, J. (2017). What is the Difference Between Test and Validation Datasets? https://machinelearningmastery.com/difference-test-validation-datasets/.

[8] Brownlee, J. (2018). How to Use ROC Curves and Precision-Recall Curves for Classification in Python. https://machinelearningmastery.com/roc-curves-and-precision-recall-curves-for-classification-in-python/.

[9] Brownlee, J. (2019a). How to Develop a Pix2Pix GAN for Image-to-Image Translation. https://machinelearningmastery.com/how-to-develop-a-pix2pix-gan-for-image-to-image-translation/.

[10] Brownlee, J. (2019b). How to Evaluate Generative Adversarial Networks. https://machinelearningmastery.com/how-to-evaluate-generative-adversarial-networks/.

[11] Bui, T. A., Lee, P. J., Lum, K. Y., Loh, C., and Tan, K. (2020). Deep Learning for Landslide Recognition in Satellite Architecture. *IEEE Access*, 8:143665–143678.

[12] Burrows, K., Walters, R. J., Milledge, D., and Densmore, A. L. (2020). A systematic exploration of satellite radar coherence methods for rapid landslide detection. *Nat. Hazards Earth Syst. Sci.*, 20(11).

[13] Burrows, K., Walters, R. J., Milledge, D., Spaans, K., and Densmore, A. L. (2019). A new method for large-scale landslide classification from satellite radar. *Remote Sens.*, 11(3).

[14] Catani, F. (2020). Landslide detection by deep learning of non-nadiral and crowdsourced optical images. *Landslides*.

[15] Chollet, F. (2019). Image segmentation with a U-Net-like architecture. https://keras.io/examples/vision/oxford_pets_image_segmentation/.

[16] Cruden, D. M. (1991). A simple definition of a landslide. *Bull. Int. Assoc. Eng. Geol. - Bull. l'Association Int. Géologie l'Ingénieur*, 43(1).

[17] Dilley, M., Chen, R. S., Deichmann, U., Lerner-Lam, A., Arnold, M., Agwe, J., Buys, P., Kjekstad, O., Lyon, B., and Yetman, G. (2005). -. *Nat. Disaster Hotspots A Glob. Risk Anal.*, pages 1–132.

[ESA] ESA. IW GRD Resolutions - Sentinel-1 SAR Technical Guide - Sentinel Online - Sentinel Online. https://sentinels.copernicus.eu/web/sentinel/technical-guides/sentinel-1-sar/products-algorithms/level-1-algorithms/ground-range-detected/iw.

[19] Ferrario, M. F. (2019). Landslides triggered by multiple earthquakes: insights from the 2018 Lombok (Indonesia) events. *Nat. Hazards*, 98(2).

[20] Fitch, J. P. (1988). *Synthetic Aperture Radar*. Springer New York.

[21] Froude, M. J. and Petley, D. N. (2018). Global fatal landslide occurrence from 2004 to 2016. *Nat. Hazards Earth Syst. Sci.*, 18(8).

[22] Garbin, C., Zhu, X., and Marques, O. (2020). Dropout vs. batch normalization: an empirical study of their impact to deep learning. *Multimed. Tools Appl.*, 79(19-20):12777–12815.

[23] Gargiulo, M., Dell'aglio, D. A. G., Iodice, A., Riccio, D., and Ruello, G. (2020). Integration of Sentinel-1 and Sentinel-2 Data for Land Cover Mapping Using W-Net. *Sensors*, 20(10).

[24] Ghorbanzadeh, O., Blaschke, T., Gholamnia, K., Meena, S. R., Tiede, D., and Aryal, J. (2019). Evaluation of different machine learning methods and deep-learning convolutional neural networks for landslide detection. *Remote Sens.*, 11(2).

[Gkioxari et al.] Gkioxari, G., Lin, T.-Y., Tech, C., Hariharan, B., and Dollár, P. Deep Learning for Instance-level Object Understanding CVPR 2017 Tutorial on Deep Learning for Objects and Scenes Ross Girshick in collaboration with Kaiming He.

[26] Gong, M., Niu, X., Zhan, T., and Zhang, M. (2019). A coupling translation network for change detection in heterogeneous images. *Int. J. Remote Sens.*, 40(9).

[27] Gu, F., Zhang, H., and Wang, C. (2020). A two-component deep learning network for SAR image denoising. *IEEE Access*, 8.

[28] Handwerger, A., Jones, S., Huang, M.-H., Amatya, P., Kerner, H., and Kirschbaum, D. (2020). Rapid landslide identification using synthetic aperture radar amplitude change detection on the Google Earth Engine. *Nat. Hazards Earth Syst. Sci.*

[29] Hay, G. J. and Castilla, G. (2008). Geographic object-based image analysis (GEOBIA): A new name for a new discipline. *Lect. Notes Geoinf. Cartogr.*, 0(9783540770572).

[30] He, W. and Yokoya, N. (2018). Multi-temporal Sentinel-1 and -2 Data Fusion for Optical Image Simulation.

[31] Highland, L. M. and Bobrowsky, P. (2008). The Landslide Handbook-A Guide to Understanding Landslides.

[32] Hong, H., Chen, W., Xu, C., Youssef, A. M., Pradhan, B., and Bui, D. T. (2017). Rainfall-induced landslide susceptibility assessment at the Chongren area (China) using frequency ratio, certainty factor, and index of entropy. *Geocarto Int.*, 32(2).

[33] Hu, K., Zhang, X., You, Y., Hu, X., Liu, W., and Li, Y. (2019). Landslides and dammed lakes triggered by the 2017 Ms6.9 Milin earthquake in the Tsangpo gorge. *Landslides*.

[34] Isola, P., Zhu, J.-Y., Zhou, T., and Efros, A. A. (2016). Image-to-Image Translation with Conditional Adversarial Networks. *Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017*, 2017-Janua:5967–5976.

[35] Jadon, S. (2020). A survey of loss functions for semantic segmentation.

[36] Jung, J. and Yun, S. H. (2020). Evaluation of coherent and incoherent landslide detection methods based on synthetic aperture radar for rapid response: A case study for the 2018 Hokkaido landslides. *Remote Sens.*, 12(2).

[37] Kabani, A. W. and El-Sakka, M. R. (2017). Ejection fraction estimation using a wide convolutional neural network. *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, 10317 LNCS:87–96.

[38] Kamali, M. E., Abuelgasim, A., Papoutsis, I., Loupasakis, C., and Kontoes, C. (2020). A reasoned bibliography on SAR interferometry applications and outlook on big interferometric data processing.

[39] Kamencay, P., Benco, M., Mizdos, T., and Radil, R. (2017). A new method for face recognition using convolutional neural network. *Adv. Electr. Electron. Eng.*, 15(4 Special Issue):663–672.

[40] Kamiyama, J., Noro, T., Sakagami, M., Suzuki, Y., Yoshikawa, K., Hikosaka, S., and Hirata, I. (2018). Detection of Landslide Candidate Interference Fringes in DInSAR Imagery Using Deep Learning.

[41] Kaplan, G. and Avdan, U. (2018). Sentinel-1 and Sentinel-2 data fusion for wetlands mapping: Balikdami, Turkey. In *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci. - ISPRS Arch.*, volume 42, pages 729–734. International Society for Photogrammetry and Remote Sensing.

[42] Khan, A., Govil, H., Kumar, G., and Dave, R. (2019). Synergistic use of Sentinel-1 and Sentinel-2 for improved LULC mapping with special reference to bad land class: a case study for Yamuna River floodplain, India. *Spat. Inf. Res.*, 28.

[43] Konishi, T. and Suga, Y. (2018). Landslide detection using COSMO-SkyMed images: A case study of a landslide event on Kii Peninsula, Japan. *Eur. J. Remote Sens.*, 51(1).

[44] Konishi, T. and Suga, Y. (2019). Landslide detection with ALOS-2/PALSAR-2 data using convolutional neural networks: a case study of 2018 Hokkaido Eastern Iburi earthquake. page 6. SPIE-Intl Soc Optical Eng.

[45] Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. In *Adv. Neural Inf. Process. Syst.*, volume 2.

[46] Kulkarni, A., Mohandoss, T., Northrup, D., Mwebaze, E., and Alemohammad, H. (2020). Semantic Segmentation of Medium-Resolution Satellite Imagery using Conditional Generative Adversarial Networks.

[47] Kurtz, C., Stumpf, A., Malet, J. P., Gançarski, P., Puissant, A., and Passat, N. (2014). Hierarchical extraction of landslides from multiresolution remotely sensed optical images. *ISPRS J. Photogramm. Remote Sens.*, 87.

[48] Kyriou, A. and Nikolakopoulos, K. (2020). Landslide mapping using optical and radar data: a case study from Aminteo, Western Macedonia Greece.

[49] Lacroix, P., Bièvre, G., Pathier, E., Kniess, U., and Jongmans, D. (2018). Use of Sentinel-2 images for the detection of precursory motions before landslide failures. *Remote Sens. Environ.*, 215.

[50] Lau, S. (2017). A Walkthrough of Convolutional Neural Network — Hyperparameter Tuning | by Suki Lau | Towards Data Science. https://towardsdatascience.com/a-walkthrough-of-convolutional-neural-network-7f474f91d7bd.

[51] Lin, T.-Y., Goyal, P., Girshick, R., He, K., and Dollár, P. (2017). Focal Loss for Dense Object Detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 42(2):318–327.

[52] Liu, J., Gong, M., Qin, K., and Zhang, P. (2018). A Deep Convolutional Coupling Network for Change Detection Based on Heterogeneous Optical and Radar Images. *IEEE Trans. Neural Networks Learn. Syst.*, 29(3).

[53] Liu, P., Wei, Y., Wang, Q., Chen, Y., and Xie, J. (2020). Research on post-earthquake landslide extraction algorithm based on improved U-net model. *Remote Sens.*, 12(5).

[54] Liu, S., Zhang, J., Chen, Y., Liu, Y., Qin, Z., and Wan, T. (2019). Pixel Level Data Augmentation for Semantic Image Segmentation Using Generative Adversarial Networks. *ICASSP, IEEE Int. Conf. Acoust. Speech Signal Process. - Proc.*, 2019-May:1902–1906.

[55] Luo, J., Huang, J., and Li, H. (2021). A case study of conditional deep convolutional generative adversarial networks in machine fault diagnosis. *J. Intell. Manuf.*, 32(2):407–425.

[56] Lv, Z., Liu, T., Kong, X. B., Shi, C., and Benediktsson, J. A. (2020). Landslide Inventory Mapping with Bitemporal Aerial Remote Sensing Images Based on the Dual-Path Fully Convolutional Network. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, 13:4575–4584.

[57] Ma, W., Xiong, Y., Wu, Y., Yang, H., Zhang, X., and Jiao, L. (2019). Change Detection in Remote Sensing Images Based on Image Mapping and a Deep Capsule Network. *Remote Sens.*, 11(6).

[58] Mabu, S., Nakayama, Y., and Kuremoto, T. (2020). Landslide Classification from Synthetic Aperture Radar Images Using Convolutional Neural Network with Multichannel Information. *Journal of Signal Processing*, 24(2):61–73.

[59] Mattia, F. D., Galeone, P., Simoni, M. D., and Ghelfi, E. (2019). A Survey on GANs for Anomaly Detection.

[60] Mirza, M. and Osindero, S. (2014). Conditional Generative Adversarial Nets.

[61] Mohammadimanesh, F., Salehi, B., Mahdianpari, M., Gill, E., and Molinier, M. (2019). A new fully convolutional neural network for semantic segmentation of polarimetric SAR imagery in complex land cover ecosystem. *ISPRS J. Photogramm. Remote Sens.*, 151:223–236.

[62] Mondini, A. C., Santangelo, M., Rocchetti, M., Rossetto, E., Manconi, A., and Monserrat, O. (2019). Sentinel-1 SAR Amplitude Imagery for Rapid Landslide Detection.

[63] Moosavi, V., Talebi, A., and Shirmohammadi, B. (2014). Producing a landslide inventory map using pixel-based and object-oriented approaches optimized by Taguchi method. *Geomorphology*, 204.

[64] Niu, X., Gong, M., Zhan, T., and Yang, Y. (2019). A Conditional Adversarial Network for Change Detection in Heterogeneous Images. *IEEE Geosci. Remote Sens. Lett.*, 16(1).

[of Japanese Geographers (AJG)] of Japanese Geographers (AJG), A. : 307. https://ajg-disaster.blogspot.com/2018/07/3077.html.

[66] Ohki, M., Abe, T., Tadono, T., and Shimada, M. (2020). Landslide detection in mountainous forest areas using polarimetry and interferometric coherence. *Earth, Planets Sp.*, 72(1).

[67] Piralilou, S. T., Shahabi, H., Jarihani, B., Ghorbanzadeh, O., Blaschke, T., Gholamnia, K., Meena, S. R., and Aryal, J. (2019). Landslide detection using multi-scale image segmentation and different machine learning models in the higher himalayas. *Remote Sens.*, 11(21).

[68] Prakash, N., Manconi, A., and Loew, S. (2020). Mapping landslides on EO data: Performance of deep learning models vs. Traditional machine learning models. *Remote Sens.*, 12(3).

[69] Qi, W., Wei, M., Yang, W., Xu, C., and Ma, C. (2020). Automatic mapping of landslides by the ResU-Net.

[70] Robinson, T. R., Rosser, N., and Walters, R. J. (2019). The Spatial and Temporal Influence of Cloud Cover on Satellite-Based Emergency Mapping of Earthquake Disasters. *Sci. Rep.*, 9(1).

[71] Ronneberger, O., Fischer, P., and Brox, T. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation.

[72] Rosebrook, A. (2016). Intersection over Union (IoU) for object detection - PyImageSearch. https://www.pyimagesearch.com/2016/11/07/intersection-over-union-iou-for-object-detection/.

[73] Salehinejad, H., Valaee, S., Dowdell, T., Colak, E., and Barfett, J. (2017). Generalization of Deep Neural Networks for Chest Pathology Classification in X-Rays Using Generative Adversarial Networks. *ICASSP, IEEE Int. Conf. Acoust. Speech Signal Process. - Proc.*, 2018-April:990–994.

[74] Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., and Chen, X. (2016). Improved Techniques for Training GANs. *Adv. Neural Inf. Process. Syst.*, pages 2234–2242.

[75] Sharma, S. (2017). Activation Functions in Neural Networks | by SAGAR SHARMA | Towards Data Science. https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6.

[76] Shi, W., Zhang, M., Zhang, R., Chen, S., and Zhan, Z. (2020). Change detection based on artificial intelligence: State-of-the-art and challenges.

[77] Shorten, C. and Khoshgoftaar, T. M. (2019). A survey on Image Data Augmentation for Deep Learning. *J. Big Data*, 6(1).

[78] Sivasankar, T., Ghosh, S., and Joshi, M. (2021). Exploitation of optical and SAR amplitude imagery for landslide identification: a case study from Sikkim, Northeast India. *Environ. Monit. Assess.*, 193(7):386.

[79] Sivasankar, T., Kumar, D., Srivastava, H. S., and Patel, P. (2018). Advances in radar remote sensing of agricultural crops: A review. *Int. J. Adv. Sci. Eng. Inf. Technol.*, 8(4):1126–1137.

[80] Stumpf, A. and Kerle, N. (2011). Object-oriented mapping of landslides using Random Forests. *Remote Sens. Environ.*, 115(10).

[81] Tang, T. W., Kuo, W. H., Lan, J. H., Ding, C. F., Hsu, H., and Young, H. T. (2020). Anomaly detection neural network with dual auto-encoders GAN and its industrial inspection applications. *Sensors (Switzerland)*, 20(12).

[82] Tiu, E. (2019). Metrics to Evaluate your Semantic Segmentation Model | by Ekin Tiu | Towards Data Science. https://towardsdatascience.com/metrics-to-evaluate-your-semantic-segmentation-model-6bcb99639aa2.

[83] Uemoto, J., Moriyama, T., Nadai, A., Kojima, S., and Umehara, T. (2019). Landslide detection based on height and amplitude differences using pre- and post-event airborne X-band SAR data. *Nat. Hazards*, 95(3):485–503.

[84] Yakubovskiy, P. (2019). segmentation models. https://github.com/qubvel/segmentation_models/blob/master/segmentation_models/losses.py.

[85] Yessou, H., Sumbul, G., and Demir, B. (2020). A Comparative Study of Deep Learning Loss Functions for Multi-Label Remote Sensing Image Classification. In *Int. Geosci. Remote Sens. Symp.*

[86] Yu, B., Chen, F., and Xu, C. (2020). Landslide detection based on contour-based deep learning framework in case of national scale of Nepal in 2015. *Comput. Geosci.*, 135.

[87] Yu, Z., Wang, W., Li, C., Liu, W., and Yang, J. (2018). Speckle Noise Suppression in SAR Images Using a Three-Step Algorithm. *Sensors (Basel).*, 18(11).

[88] Zenati, H., Foo, C.-S., Lecouat, B., Manek, G., and Chandrasekhar, V. R. (2018). Workshop track-ICLR 2018 EFFICIENT GAN-BASED ANOMALY DETECTION.

[89] Zhang, S., Li, R., Wang, F., and Iio, A. (2019). Characteristics of landslides triggered by the 2018 Hokkaido Eastern Iburi earthquake, Northern Japan. *Landslides*, 16(9):1691–1708.

A

A.1 SATELLITE IMAGES

The tables A.1, A.2,A.3 and A.4 show the Copernicus Sentinel-1 satellite images which were used for the respective regions of Hiroshima, Japan, Hokkaido, Japan, Lombok, Indonesia and Tsangpo Gorge, China.

descending, pre event	S1A_IW_GRDH_1SDV_20180528T211627_20180528T211656_022110_0263F2_7CF0
	S1A_IW_GRDH_1SDV_20180604T210817_20180604T210842_022212_02672C_A700
	S1A_IW_GRDH_1SDV_20180609T211628_20180609T211657_022285_02696A_A8C6
	S1A_IW_GRDH_1SDV_20180616T210818_20180616T210843_022387_026C91_2BFC
	S1A_IW_GRDH_1SDV_20180621T211629_20180621T211658_022460_026EBC_01A2
descending, post event	S1B_IW_GRDH_1SDV_20180709T211537_20180709T211606_011739_01598A_59DD
	S1A_IW_GRDH_1SDV_20180710T210820_20180710T210845_022737_0276D8_EE0D
	S1A_IW_GRDH_1SDV_20180715T211630_20180715T211659_022810_027920_C6F1
	S1B_IW_GRDH_1SDV_20180716T210720_20180716T210749_011841_015CB2_7EBC
	S1B_IW_GRDH_1SDV_20180716T210749_20180716T210817_011841_015CB2_F268
	S1A_IW_GRDH_1SDV_20180722T210820_20180722T210845_022912_027C48_FE6D
	S1A_IW_GRDH_1SDV_20180727T211631_20180727T211700_022985_027EA7_561D
	S1A_IW_GRDH_1SDV_20180803T210821_20180803T210846_023087_0281C5_E538
	S1A_IW_GRDH_1SDV_20180808T211632_20180808T211701_023160_02841E_B0BB
ascending, pre event	S1B_IW_GRDH_1SDV_20180529T090534_20180529T090559_011134_0146BD_79F9
	S1B_IW_GRDH_1SDV_20180603T091332_20180603T091406_011207_014915_C765
	S1B_IW_GRDH_1SDV_20180610T090535_20180610T090600_011309_014C3A_460F
	S1B_IW_GRDH_1SDV_20180615T091333_20180615T091407_011382_014E72_2193
ascending, post event	S1B_IW_GRDH_1SDV_20180529T090534_20180529T090559_011134_0146BD_79F9
	S1B_IW_GRDH_1SDV_20180603T091332_20180603T091406_011207_014915_C765
	S1B_IW_GRDH_1SDV_20180610T090535_20180610T090600_011309_014C3A_460F
	S1B_IW_GRDH_1SDV_20180615T091333_20180615T091407_011382_014E72_2193

Table A.1: S-1 SAR images used for Hiroshima region

descending, pre event	S1A_IW_GRDH_1SDV_20180912T203319_20180912T203347_023670_029470_9857
	S1A_IW_GRDH_1SDV_20180917T204113_20180917T204138_023743_0296C9_5A07
	S1A_IW_GRDH_1SDV_20180924T203319_20180924T203348_023845_029A25_4838
	S1A_IW_GRDH_1SDV_20180929T204113_20180929T204138_023918_029C78_1E7D
descending, post event	S1A_IW_GRDH_1SDV_20181011T204113_20181011T204138_024093_02A236_CC29
	S1A_IW_GRDH_1SDV_20181018T203320_20181018T203348_024195_02A592_E12B
	S1A_IW_GRDH_1SDV_20181023T204113_20181023T204138_024268_02A7E7_CF20
	S1A_IW_GRDH_1SDV_20181030T203320_20181030T203348_024370_02AB36_2290
	S1A_IW_GRDH_1SDV_20181104T204113_20181104T204138_024443_02ADD3_2E6A
ascending, pre event	S1B_IW_GRDH_1SDV_20180913T082635_20180913T082700_012694_0176CF_1486
	S1B_IW_GRDH_1SDV_20180913T082700_20180913T082725_012694_0176CF_9BA1
	S1B_IW_GRDH_1SDV_20180925T082635_20180925T082700_012869_017C34_A97C
	S1B_IW_GRDH_1SDV_20180925T082700_20180925T082725_012869_017C34_F6E9
ascending, post event	S1B_IW_GRDH_1SDV_20181007T082636_20181007T082701_013044_01818A_EB0D
	S1B_IW_GRDH_1SDV_20181007T082701_20181007T082726_013044_01818A_A8B8
	1B_IW_GRDH_1SDV_20181019T082636_20181019T082701_013219_0186F1_77DF
	S1B_IW_GRDH_1SDV_20181019T082701_20181019T082726_013219_0186F1_90C1
	S1B_IW_GRDH_1SDV_20181031T082653_20181031T082718_013394_018C6D_B89F

Table A.2: S-1 SAR images used for Hokkaido region

descending, pre event	S1B_IW_GRDH_1SDV_20180630T215223_20180630T215257_011608_015571_9ECE
	S1A_IW_GRDH_1SDV_20180706T215300_20180706T215328_022679_027513_AB0F
	S1B_IW_GRDH_1SDV_20180712T215224_20180712T215258_011783_015ADD_EADB
	S1A_IW_GRDH_1SDV_20180718T215301_20180718T215329_022854_027A6F_92D5
	S1B_IW_GRDH_1SDV_20180724T215225_20180724T215259_011958_016031_8087
descending, post event	S1A_IW_GRDH_1SDV_20180823T215303_20180823T215331_023379_028B27_91C6
	S1B_IW_GRDH_1SDV_20180829T215227_20180829T215300_012483_01705C_AA11
	S1A_IW_GRDH_1SDV_20180904T215304_20180904T215332_023554_0290B8_771C
	S1B_IW_GRDH_1SDV_20180910T215227_20180910T215301_012658_0175BE_3BA0
	S1A_IW_GRDH_1SDV_20180916T215304_20180916T215332_023729_029655_F231
ascending, pre event	S1A_IW_GRDH_1SDV_20180628T103306_20180628T103336_022555_02717D_EC04
	S1A_IW_GRDH_1SDV_20180703T104116_20180703T104145_022628_0273A4_9BA8
	S1B_IW_GRDH_1SDV_20180709T104031_20180709T104100_011732_015959_D30E
	S1A_IW_GRDH_1SDV_20180710T103307_20180710T103337_022730_0276A3_893F
	S1A_IW_GRDH_1SDV_20180715T104116_20180715T104145_022803_0278E6_FCD6
	S1B_IW_GRDH_1SDV_20180721T104052_20180721T104121_011907_015EB5_2334
	S1A_IW_GRDH_1SDV_20180722T103308_20180722T103338_022905_027C0F_2890
	S1A_IW_GRDH_1SDV_20180727T104117_20180727T104146_022978_027E6E_D0ED
ascending, post event	S1A_IW_GRDH_1SDV_20180820T104118_20180820T104147_023328_028992_CEAA
	S1B_IW_GRDH_1SDV_20180826T104034_20180826T104103_012432_016ED0_FA43
	S1A_IW_GRDH_1SDV_20180827T103310_20180827T103340_023430_028CCD_947E
	S1A_IW_GRDH_1SDV_20180901T104119_20180901T104148_023503_028F20_8CB2
	S1A_IW_GRDH_1SDV_20180908T103311_20180908T103340_023605_029267_0EED
	S1A_IW_GRDH_1SDV_20180913T104120_20180913T104149_023678_0294B6_0AE0

Table A.3: S-1 SAR images used for Lombok region

descending, pre event	S1A_IW_GRDH_1SDV_20171106T233728_20171106T233753_019151_02068B_7197
	S1A_IW_GRDH_1SDV_20171106T233753_20171106T233818_019151_02068B_DA74
descending, post event	S1A_IW_GRDH_1SDV_20171130T233728_20171130T233753_019501_02117D_3FDB
	S1A_IW_GRDH_1SDV_20171130T233753_20171130T233818_019501_02117D_4C04
	S1A_IW_GRDH_1SDV_20171212T233727_20171212T233752_019676_0216FB_1DC2
	S1A_IW_GRDH_1SDV_20171212T233752_20171212T233817_019676_0216FB_A7B6
ascending, pre event	S1A_IW_GRDH_1SDV_20171018T114115_20171018T114140_018867_01FDED_406D
	S1A_IW_GRDH_1SDV_20171030T114115_20171030T114140_019042_02033F_F2D4
	S1A_IW_GRDH_1SDV_20171111T114115_20171111T114140_019217_0208A2_DDC4
ascending, post event	S1A_IW_GRDH_1SDV_20171123T114114_20171123T114139_019392_020E29_DD81
	S1A_IW_GRDH_1SDV_20171205T114114_20171205T114139_019567_0213A7_D327
	S1A_IW_GRDH_1SDV_20171217T114114_20171217T114139_019742_021916_FCAD

Table A.4: S-1 SAR images used for Tsangpo Gorge region

B

B.1 DEFINITION OF LARGE LANDSLIDES

Figure B.1 shows the minimum sizes of the landslides defined as 'large' for each region, TLR: Hiroshima, Japan, Hokkaido, Japan, Lombok, Indonesia and Tsangpo Gorge, China respectively.

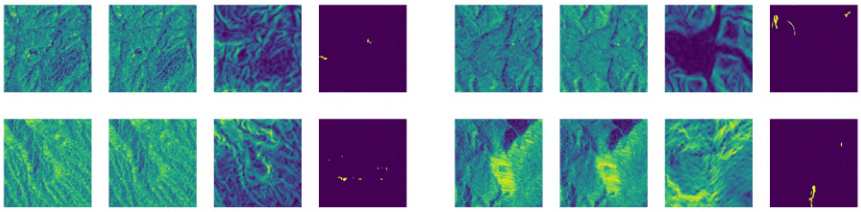

Figure B.1: The minimum sizes of the landslides defined as 'large' for each region, TLR: Hiroshima, Japan, Hokkaido, Japan, Lombok, Indonesia and Tsangpo Gorge, China.

B.2 ADDITIONAL TESTS WITH AMPSLOPE REPRESENTATION, FEATURING LARGE LANDSLIDES ONLY

Figures B.2 and B.3 show a random subset of the source, generated target and target images during training of the cGAN, with as input the ampslope source images and the corresponding target images. These correspond to the training set in section 4.4, but here for descending direction and a combination of ascending and descending direction.

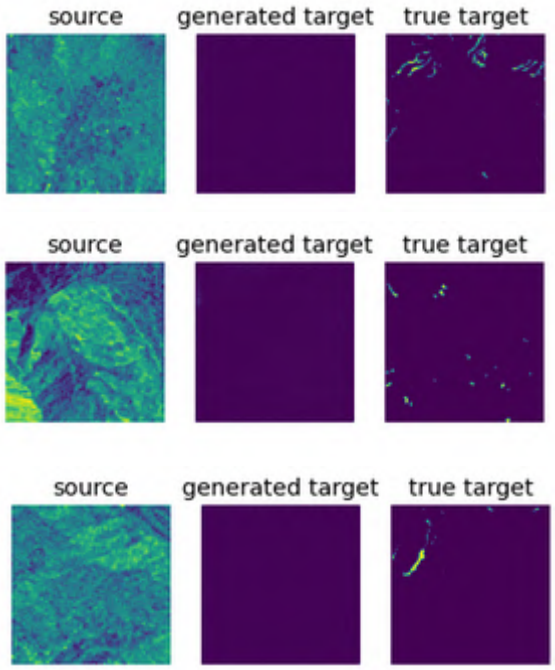

Figure B.2: Source, target and generated target images for the ampslope representation featuring large landslides only of descending direction from the cGAN, with batch size 32.

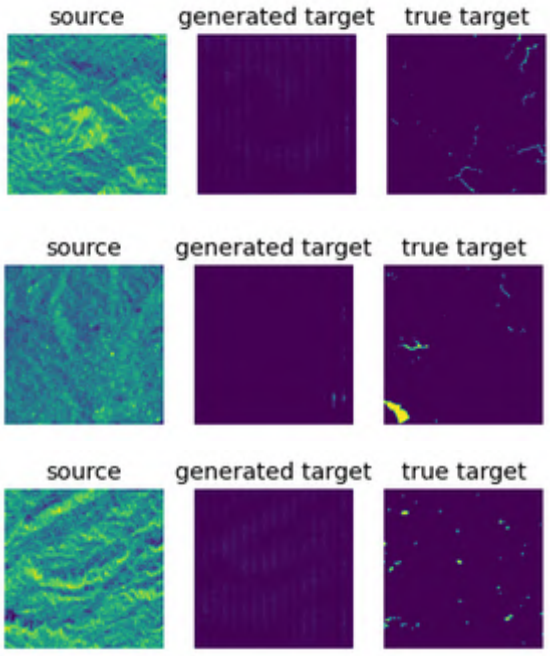

Figure B.3: Source, target and generated target images for the ampslope representation featuring large landslides only of both ascending and descending direction combined from the cGAN, with batch size 32.

Figures B.4 and B.5 show the predicted target and target images for the ampslope representations featuring large landslides only, of descending and a combination of descending and ascending direction respectively.

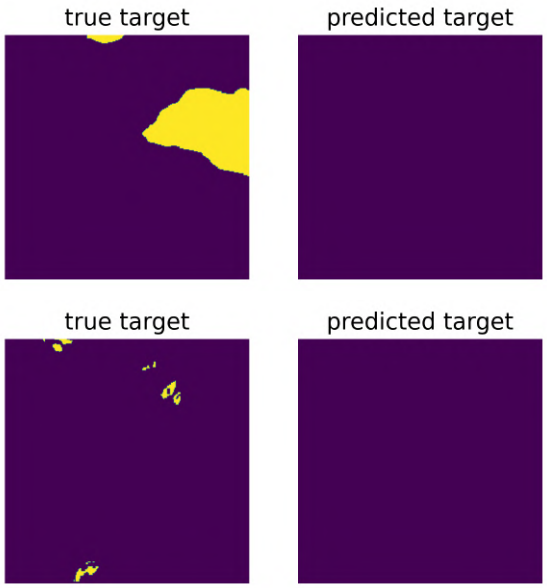

Figure B.4: True target and predicted target images for the ampslope representations featuring large landslides only from the cGAN, of descending direction with batch size 32.

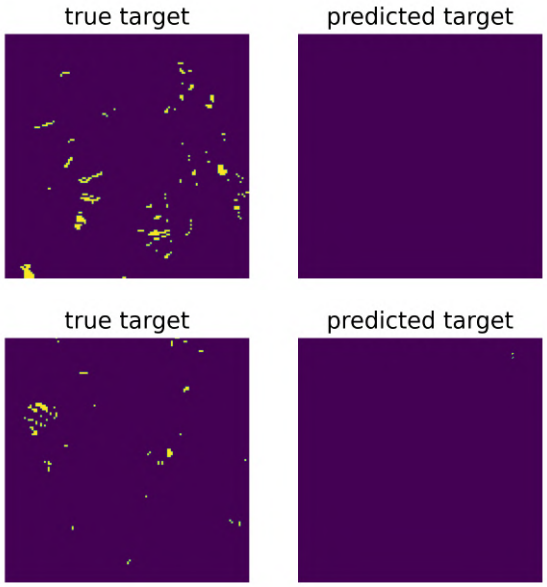

Figure B.5: True target and predicted target images for the ampslope representations featuring large landslides only from the cGAN, of both ascending and descending direction combined with batch size 32.

B.3 ADDITIONAL TESTS WITH AMPSLOPE REPRESENTATION, FEATURING ALL LANDSLIDES

Figure B.6 shows the learning curve based on the loss functions of the cGAN for the ampslope representation featuring all landslides.

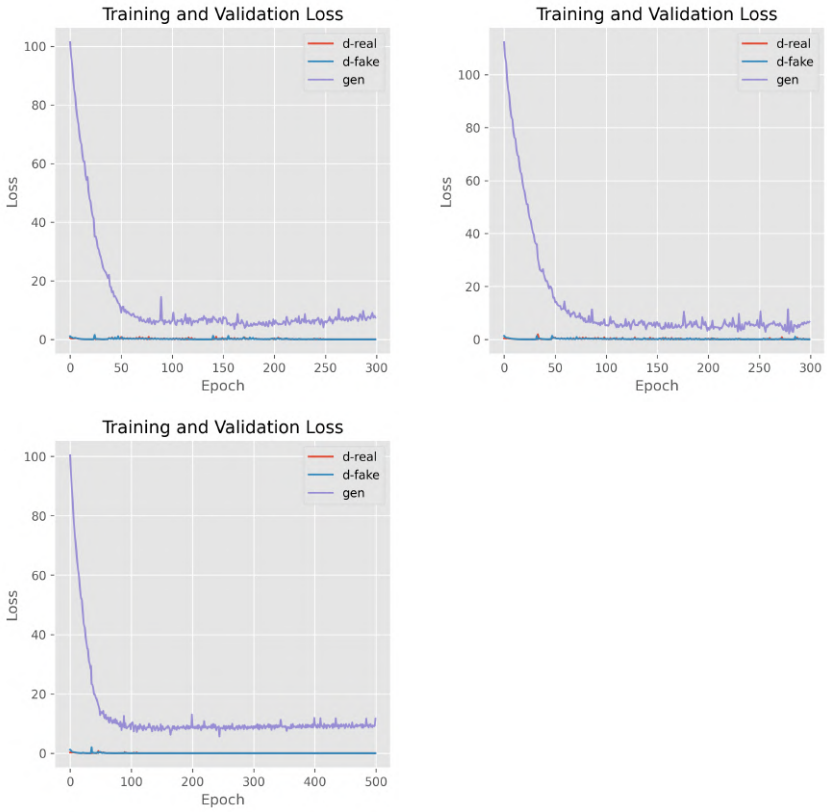

Figure B.6: Learning curves showing the loss for the ampslope representation featuring all landslides from the cGAN. The 'gen' line is the loss of the generator, the 'd-real' shows the loss of the discriminator with regards to the real targets and the 'd-fake' shows the loss of the discriminator with regards to the generated fake targets. The top figure shows the learning curve for the data set of ascending orbit, the middle for that of descending orbit, and the last a combination of the two. The batch size of these models are 64. The x-axis shows the amount of iterations, e.g. the amount of epochs * the amount of batches per epoch. The y-axis shows the loss value.

Figure B.7 shows the learning curves based on the mIoU from the cGAN for the ampslope representation featuring all landslides.

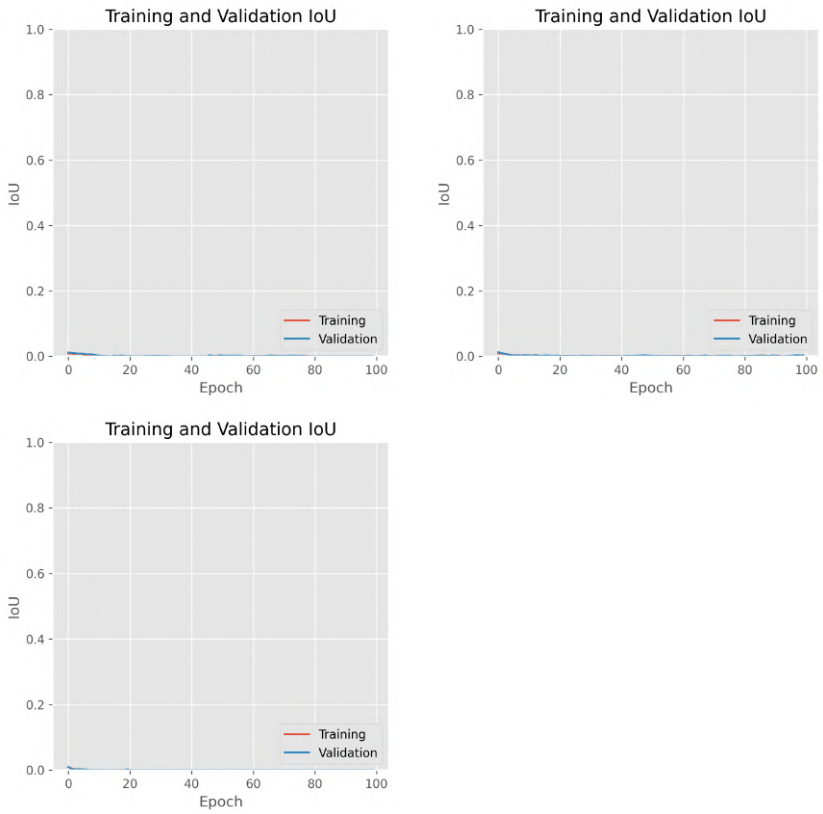

Figure B.7: Learning curves showing the mean Intersection over Union mIoU for the ampslope representation featuring all landslides from the cGAN. The top figure shows the learning curve for the data set of ascending orbit, the middle for that of descending orbit, and the last a combination of the two. With batch size 64.

FigureB.8 shows the loss and mIoU curve of the U-net for the ampslope representation featuring all landslides.

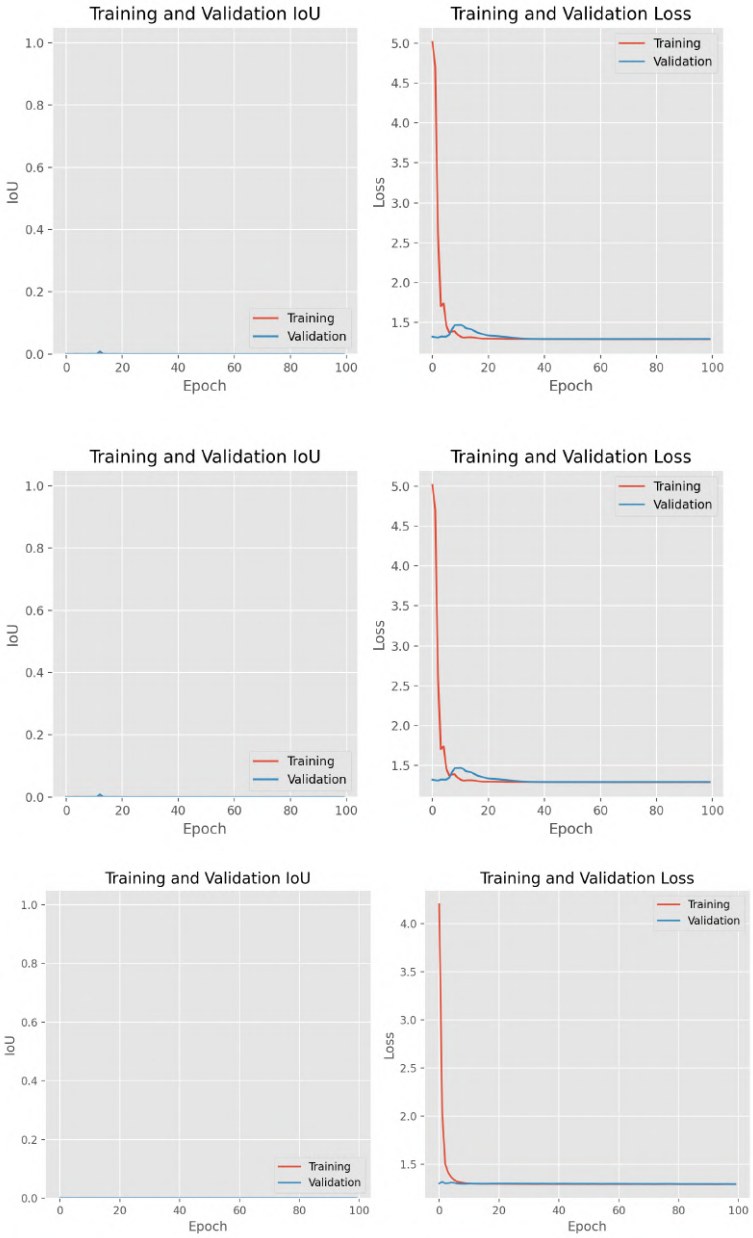

Figure B.8: Learning curves showing the loss and mean Intersection over Union mIoU for the ampslope representation featuring all landslides from the U-Net. The top figure shows the learning curve for the data set of ascending orbit, the middle for that of descending orbit, and the last a combination of the two. With batch size 64.

Figures B.9,B.10 and B.11 show a random subset of the source, generated target and target images during training of the cGAN, with as input the ampslope representation featuring all landslides and the corresponding target images.

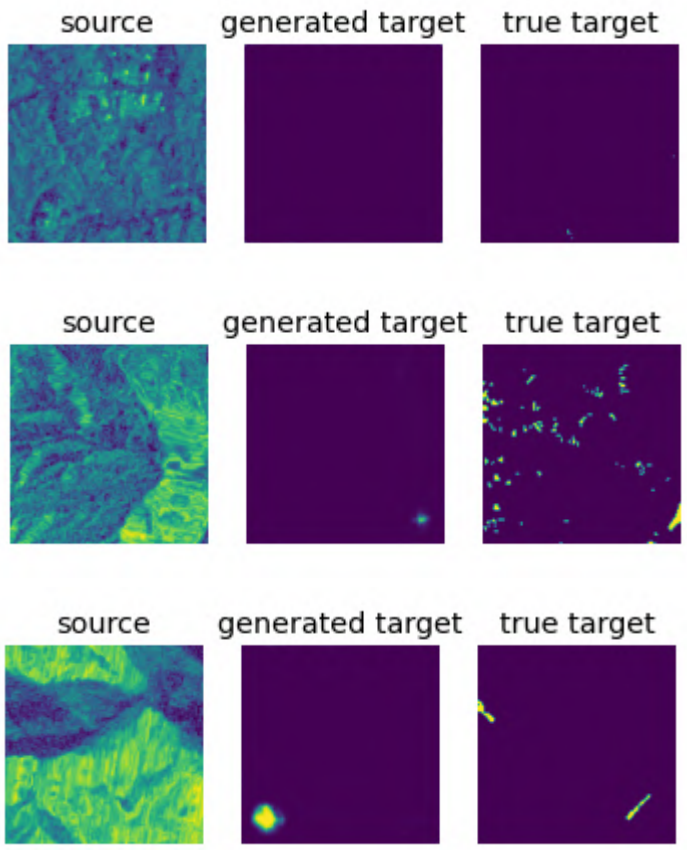

Figure B.9: Source, target and generated target images for the ampslope representation featuring all landslides of ascending direction from the cGAN, with batch size 64.

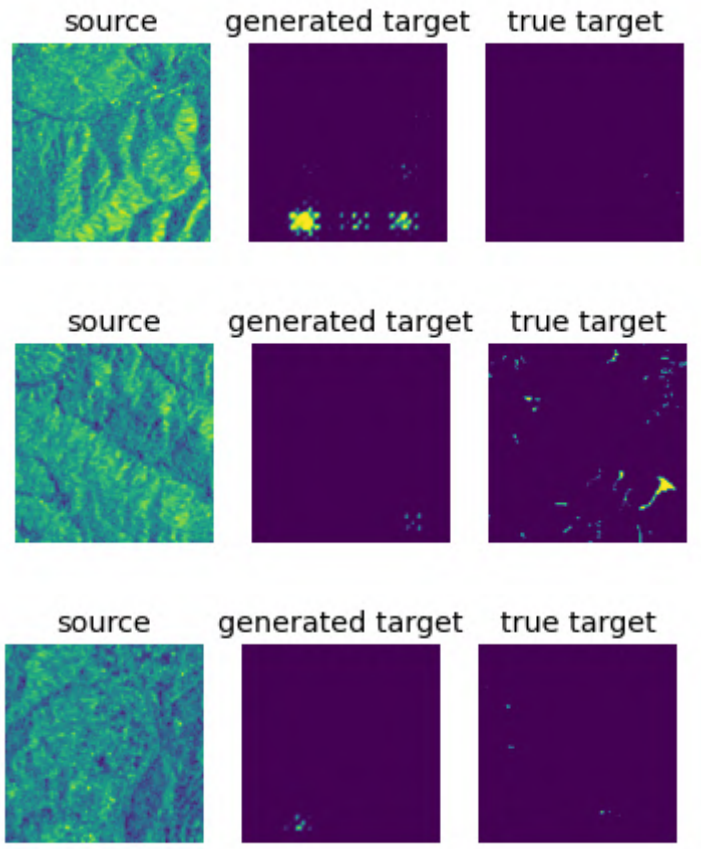

Figure B.10: Source, target and generated target images for the ampslope representation featuring all landslides of descending direction from the **Cgan!**, with batch size 64.

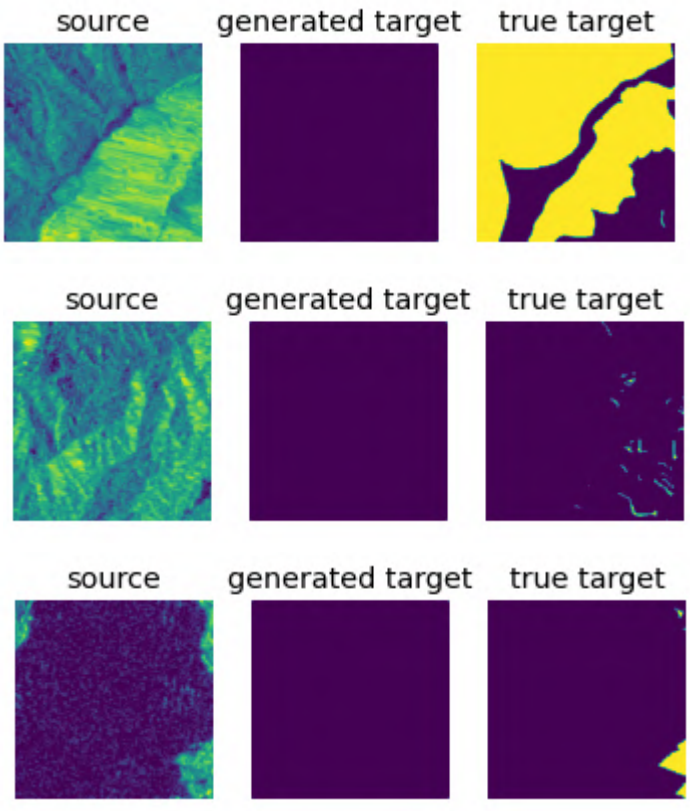

Figure B.11: Source, target and generated target images for the ampslope representation featuring all landslides with a combination of ascending and descending direction from the cGAN,with batch size 64.

Figure B.12, B.13 and B.14 show a subset of true target and predicted target images, for the ampslope representation featuring all landslides the testing data set. The testing data set is unseen data which is not used as training data by the model before.

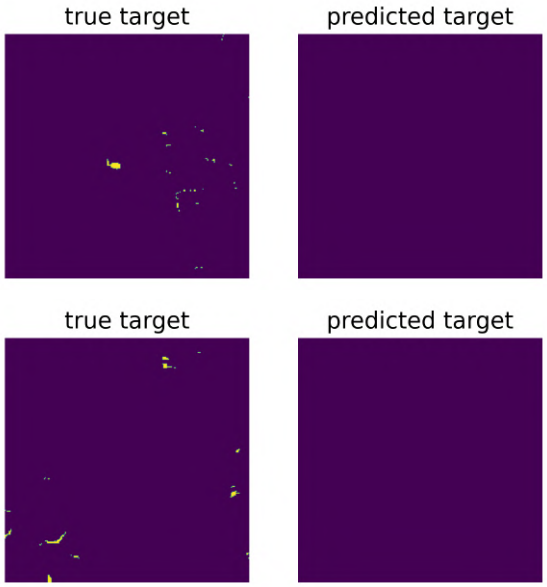

Figure B.12: True target and predicted target of the testing data set of the ampslope representation featuring all landslides from the cGAN, of ascending direction with batch size 64.

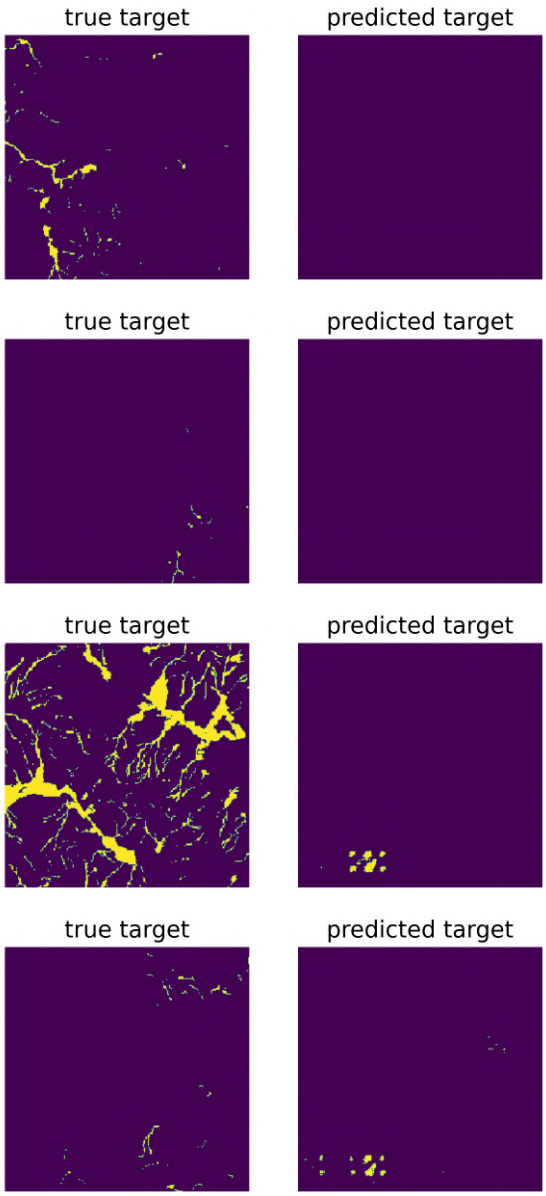

Figure B.13: True target and predicted target of the testing data set of the ampslope representation featuring all landslides from the cGAN, of descending direction with batch size 64..

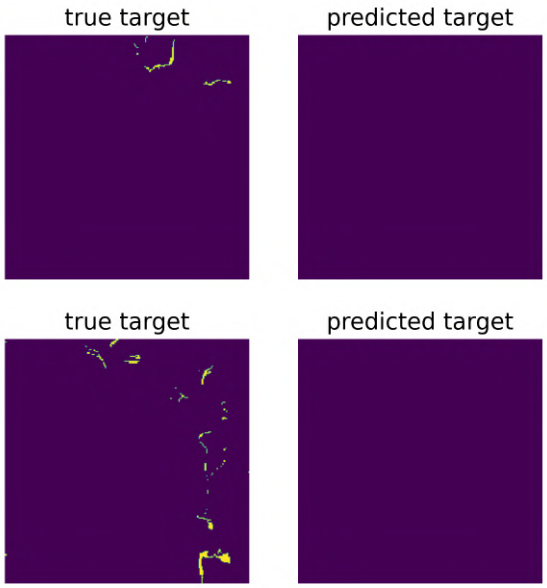

Figure B.14: True target and predicted target of the testing data set of the ampslope representation featuring all landslides from the cGAN, with a combination of descending and ascending direction with batch size 64.