Degree Project in Degree of Master of Science in Engineering

Second cycle, 30 credits

Detecting Wetlands Using Self-Supervised Convolutional Neural Networks With Data Normalization

LUCIA KARENS

Detecting Wetlands Using Self-Supervised Convolutional Neural Networks With Data Normalization

LUCIA KARENS

Master's Programme, Machine Learning, 120 credits
Date: July 15, 2026

Supervisor: Francisco J. Peña
Examiner: Danica Kragic Jensfelt
School of Electrical Engineering and Computer Science
Swedish title: Detektering av Våtmarker med Hjälp av Självövervakade Faltningsnätverk och Datanormalisering

Abstract

Wetlands are an important natural resource that needs to be preserved and monitored. Monitoring them using remote sensing and machine learning is fast, accurate, and cheap, especially while using self-supervised deep learning. The current state-of-the-art in monitoring wetlands in Sweden is the DeepAqua convolutional neural network model from the article "DeepAqua: Semantic segmentation of wetland water surfaces with SAR imagery using deep neural networks without manually annotated data". The model uses satellite imagery, radar imagery, and a U-Net based model to detect water on the radar images. However, DeepAqua performs poorly on data from years it has not been trained on due to changes in the data characteristics. In this paper we explore two data normalization methods to improve DeepAqua's robustness to new data. These methods are popular in medical imaging: Nyúl normalization, also known as histogram matching, and Z-score normalization. Each normalization method is implemented and tested with the base DeepAqua model. Both methods improve the model's prediction on years it has not been trained on. Z-score normalization is best suited for wetlands where water extent changes considerably. Nyúl normalization is slightly more powerful to generalise to unseen years, but is mostly suited for wetlands where water extent changes slightly. Methods used in medical imaging applications can thus be transferred to remote sensing applications, and data normalization is a strong option to improve DeepAqua's robustness.

Keywords

Machine Learning, Convolutional Neural Network, Self-Supervised Learning, Semantic Segmentation, Normalization, Remote Sensing, Wetlands

Sammanfattning

Våtmarker är en viktig naturresurs som behöver värnas om och övervakas. Övervakning av våtmarker med fjärranalys och maskininlärning är snabbt, noggrant, och billigt, särskilt med självövervakad djupinlärning. Den nuvarande bästa modellen för att övervaka våtmarker i Sverige är DeepAqua-modellen från "DeepAqua: Semantic segmentation of wetland water surfaces with SAR imagery using deep neural networks without manually annotated data". Modellen använder satellitbilder, radarbilder, och en U-net baserad modell för att identifiera vatten på radarbilderna. DeepAqua presterar dock sämre på år den inte har tränats på, på grund av ändringar i datans egenskaper. I den här rapporten utforskar vi två metoder för datanormalisering. Dessa metoder används inom medicinsk avbildning: Nyúl-normalisering och Z-score-normalisering. Varje normaliseringsmetod implementeras och testas med DeepAqua modellen som bas. Båda metoder förbättrar modellens förutsägelse på osedd data från senare år. Z-score-normalisering passar bäst för våtmarker där vattenutbredning förändras avsevärt. Nyúl-normalisering är något kraftfullare för att generalisera till osedd data, men är mest lämpad för våtmarker där vattenutbredning inte förändras mycket. Metoder som används i medicinska avbildningstillämpningar kan således överföras till fjärranalystillämpningar, och datanormalisering är ett starkt alternativ för att förbättra DeepAquas robusthet.

Nyckelord

Maskininlärning, Faltningsnätverk, Självledd Inlärning, Semantisk Segmentering, Normalisering, Fjärranalys, Våtmarker

Résumé

Les zones humides constituent une ressource naturelle importante qui doit être préservée et surveillée. Leur surveillance par télédetection et apprentissage automatique est rapide, précise et économique, notamment avec l'usage de l'apprentissage profond auto-supervisé. En Suède, l'état de l'art en surveillance de zones humides est représenté par le modèle DeepAqua, présenté dans l'article "DeepAqua : Semantic segmentation of wetland water surfaces with SAR imagery using deep neural networks without manually annotated data". Ce modèle utilise des données satellites et radar et un modèle basé sur U-Net pour détecter l'eau sur les images radar. Le modèle est cependant peu performant sur des données issues de d'années non utilisées pour son entraînement. Ceci est dû à un changement des charactéristiques des données. Dans ce rapport, nous explorons deux méthodes de normalization de données afin d'améliorer la robustesse de DeepAqua face à de nouvelles données. Ces méthodes sont utilisées couramment en imagerie médicale : la normalisation de Nyúl et la normalisation z-score (aussi appelée normalisation standard). Chaque méthode est implémentée et testée avec le modèle DeepAqua. Les deux méthodes mènent à une amélioration de la prédiction du modèle pour les données inédites d'années plus récentes. La normalisation z-score est particulièrement adaptée aux zones humides où la superficie en eau varie considérablement. La normalisation de Nyúl, légèrement plus performante pour la généralisation aux années non observées, convient surtout aux zones humides où la superficie en eau varie peu. Les méthodes utilisées en imagerie médicale peuvent ainsi être transposées à la télédétection, et la normalisation des données constitue est ainsi une option pertinente pour améliorer la robustesse de DeepAqua.

Mots-clés

Apprentissage Automatique, Réseau de Neurones Convolutif, Apprentissage Auto-Supervisé, Segmentation Sémantique, Normalization, Télédétection, Zones Humides

Acknowledgments

I would like to thank my supervisor Francisco Peña, for his steady support and guidance during the thesis. Thank you for the many fruitful meetings and kind reviewing of my work. Many thanks to Ioannis Iakovidis, for providing me with technical knowledge, troubleshooting, and help for data acquisition and insights into model training. Thank you to my fellow Master students in the DeepAqua project group, Carolina Zetterblad and Xu Zuo for their insights and comments during our weekly meetings, as well as moral support during our common work on mapping wetlands.

Stockholm, July 2026
Lucia Karens

Contents

List of Figures

List of Tables

Listings

List of acronyms and abbreviations

CNN	Convolutional Neural Network
DL	Deep Learning
ESA	European Space Agency
GEE	Google Earth Engine
IoU	Intersection over Union
ML	Machine Learning
MR	Magnetic Resonance
MSI	Multispectral Imaging
NDWI	Normalized Difference Water Index
RS	Remote Sensing
SAR	Synthetic Aperture Radar
SDG	Sustainable Development Goal
SU	Stockholm University
UN	United Nations

Chapter 1

Introduction

Wetlands are a vital part of the ecosystem. They are parts of land that are partially or completely covered by water, such as bogs or marshes. They act as natural barriers to floods and erosion, sequestrate carbon dioxide, have a rich biodiversity, and purify water. Preserving them is thus very important both for the planet and for humans [1]. Unfortunately, wetlands across the world are endangered and in decline [2]. Wetlands need to be preserved, and this cannot be done without monitoring them, their area, and their water extent.

Wetland monitoring, while historically done mostly through fieldwork, has in the past few decades increasingly been done via Remote Sensing (RS) [3], because it is both faster and cheaper. RS is the technique of observing and acquiring data about objects or phenomenons without being in physical contact with them, for instance through satellite observation. It is faster and cheaper than monitoring in the field because travelling in person to wetlands is a challenge since many wetlands are located in remote areas that are difficult to access. Additionally, there are many wetlands in the world, with more than 35 000 wetlands in Sweden alone [4]. Monitoring every single one of them in person would be a monumental task, both time-consuming and costly.

Alongside the rise of RS in wetland monitoring, automation techniques have also become popular [3]. Algorithms and techniques from Machine Learning (ML), and in particular more advanced models from Deep Learning (DL) have successfully been used to classify, monitor and map wetlands [3], [5], [6].

The current state-of-the-art model in wetland monitoring in Sweden is DeepAqua [7], a specialized model that maps wetlands based on water

detection. This model uses optical imagery from satellites combined with radar imagery in order to train without human-annotated data. DeepAqua can detect water under low vegetation and has very high detection scores for mapping wetlands in Sweden. However, it suffers from a lack of generalizability. When trained on data from specific years, it performs worse on data from other years, even though the geographical location is the same.

In this thesis, we aim to address the problem of generalizability of DeepAqua by normalizing the data before and during training. We investigate mathematical data normalization methods. We combine these methods with the pre-existing model and evaluate which method leads to the best mapping results on data from unseen years.

1.1 Problem

In order to efficiently and accurately monitor wetlands over several years, it is crucial to have a model that is robust to changes in satellite sensors. Changes in calibrations of the sensors that collect the data, as well as natural changes, should be managed by the model without need of retraining. Retraining the model costs time and resources, because the necessary conditions to collect data for the model are not easy to meet: the optical data needs to come from daylight images without any clouds. It is thus better to have a single robust model that uses already available data but can still produce accurate results and be used in research settings for years to come.

Currently, DeepAqua is not robust to satellite sensor adjustments. The data that is used to train DeepAqua, and in this thesis, consists partly of Synthetic Aperture Radar (SAR) imagery. However, this technique produces variable data: SAR images and their corresponding pixel intensity histograms can vary significantly for the same geographical region. For instance, the pixel intensity histograms of SAR images taken at the same location, on the same day, but on different years, show variation (see Figures 1.1 and 1.2).

This inter-year variability of the SAR data is the cause of DeepAqua's worse performance on data from years it was not trained on, and is a significant problem for the model's robustness. In this paper, we attempt to account for this data variation to improve DeepAqua's performance. We choose to investigate data normalization to improve its robustness. This leads to the following research question:

- How can we improve wetland detection using state-of-the-art networks with data normalization?

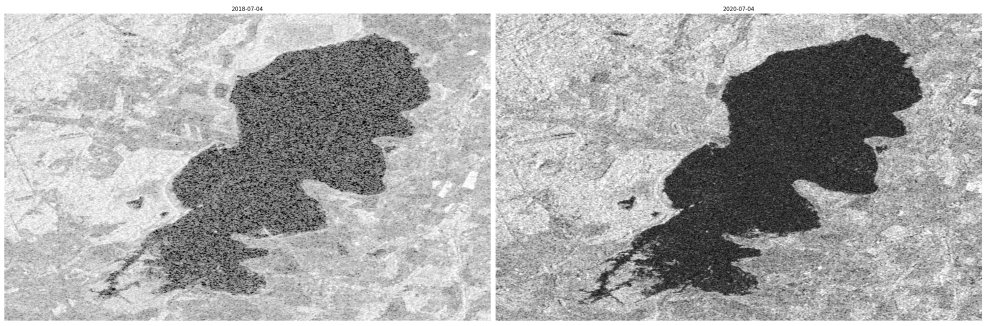

Figure 1.1: SAR images of lake Hornborga (*Hornborgasjön*), on 4th of July 2018 and 4th of July 2020

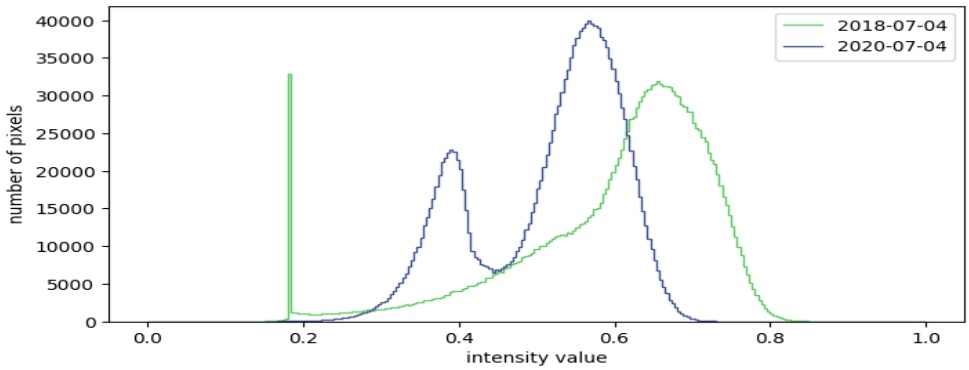

Figure 1.2: Pixel intensity histograms of lake Hornborga (*Hornborgasjön*), on 4th of July 2018 and 4th of July 2020

1.2 Purpose

The purpose of this thesis is to further research into wetland detection. We aim to improve existing wetland detection in order to create robust and reliable tools for the research community. In particular, with the created model, we aim to aid the Department of Physical Geography at Stockholm University (SU) in performing their research in hydrology and wetland preservation. They will be able to use the model as a tool for monitoring the water extent level of wetlands in Sweden.

We also aim to contribute to research into data normalization and its effects on robustness of specialized DL models.

1.3 Goals

The goal of this project is to develop a DL model that monitors water extent changes in wetlands by integrating multiple data sources from satellite images, radars, and other sensors. This has been divided into the following sub-goals:

1. Collect data for training, evaluating and testing of the model.
2. Obtain a model that is robust across variable data and wetland configurations. This model should perform better than DeepAqua on the same datasets.

1.4 Contributions

With this thesis, we aim to do the following contributions to science:

- Improve current state-of-the-art for wetland mapping in Sweden.
- Investigate the use of normalization for SAR data.

1.5 Ethics and Sustainability

The project is in direct line with the United Nations (UN) Sustainable Development Goal (SDG) number fifteen, life on land [8]:

> Protect, restore and promote sustainable use of terrestrial ecosystems, sustainably manage forests, combat desertification, and halt and reverse land degradation and halt biodiversity loss.

As mentioned earlier, monitoring of wetlands is necessary when working towards their preservation. This thesis will provide part of the solution to protect wetlands and preserve their unique biodiversity.

There are no major ethical concerns in the collection and use of data for the training and evaluation of the model. The data used is publicly available and free to use for research and academic purposes [9]. The data does not have a resolution high enough to detect human individuals or other private information.

Finally, the model that we aim to develop will be specialized in detecting water using radar satellite imagery. Since it will be specifically trained and used for this purpose, there are little to no risks of misappropriation of the model for unethical purposes.

1.6 Delimitations

This project uses the same datasets as the DeepAqua model in order to compare the two models. Additionally, we base our research on a similar architecture to DeepAqua, also for comparison.

This project focuses only on data normalization techniques. While there are more techniques one can use to improve a model's robustness, the scope of this project is limited to normalization.

When building the model, we use Python ML libraries such as PyTorch [10] to assist.

Finally, this project's approach to detecting wetlands is based on water detection for open water and shallow water below vegetation. The model does not detect water below thick vegetation, and does not differentiate between open water and water below vegetation.

1.7 Structure of the thesis

Chapter 2 presents relevant background information about RS, DL, data normalization, and current state-of-the-art in wetland detection using DL. Chapter 3 presents the methodology and method used to solve the problem. In chapter 4, we present and discuss our results. Finally, we reflect on our work and future avenues of research in chapter 5.

Chapter 2

Background

This chapter provides an overview about remote sensing, as well as DL basics and background of the model architecture we are using. Additionally, this chapter presents data normalization techniques and describes the current state-of-the-art in wetland detection.

2.1 Remote Sensing

A majority of the data used for wetland monitoring using ML is RS data, in particular satellite data. It gives relevant information such as wetland location, size, the vegetal and water composition, and more. In addition, a lot of satellite data is free to use and spans decades of information. Indeed, government programs such as European Space Agency (ESA)'s Copernicus program [11] have made large amounts of recorded satellite data publicly available on platforms such as Google Earth Engine (GEE). Two types of remote sensing data are used most often in research on wetlands: optical imagery and SAR imagery [3]. Both types have different advantages and drawbacks and are briefly presented below.

2.1.1 Optical imagery

A common type of remote sensing data is optical imagery acquired by satellites. While human eyes mostly process visible light in three

electromagnetic wavelength bands (red, green, blue), optical imaging separates light into more discrete bands (Multispectral Imaging (MSI)) or even into up to hundreds of continuous bands (hyperspectral imaging). The bands are simply predefined ranges of wavelength.

Separating light into more bands makes it possible to gain additional information about types of land cover, materials, and aid object detection. This is useful for detecting or classifying wetlands.

There are several types of satellites that record optical imagery and provide data to the public, but one of the commonly used ones is the European Space Agency's Sentinel-2 satellite [12], because it operates at 13 spectral bands, produces images of the entire surface of the globe, and has a high revisit frequency of 5 days at the Equator, which makes it suitable for tracking changes in land cover, such as wetland water extent changes.

2.1.2 Synthetic Aperture Radar

Another common type of remote sensing is SAR imaging. Unlike optical imagery, which is a passive technique to collect data, SAR is an active technique. The technique consists of sending a pulse of energy at Earth and recording the reflected pulse, which will be different based on what physical structures it has interacted with [13]. This pulse is a signal at wavelengths that make it possible to acquire data regardless of cloud coverage, day and night. The most common wavelength used in SAR is C-band, which is a wavelength of around 5,6 cm that penetrates through clouds and shallow vegetation [14]. While SAR data can be less high-definition than optical imagery, the latter is dependent on cloud-free days to capture images of the surface of the Earth, while SAR is not. In addition, since C-band SAR can penetrate through shallow vegetation, areas with water at the surface that is hidden by grass or moss can be detected and used to classify those areas as wetlands, something that would be more difficult with only optical imagery. Combining optical imagery and SAR is thus increasingly done to improve wetland mapping results [3], [6].

Using the two data types together is made easier thanks to the European Space Agency's Sentinel-1 satellites [15]. These satellites (Sentinel-1A and Sentinel-1B) generate C-band SAR images of resolution down to 5 meter resolution, with a revisit frequency of 12 days for each satellite, and 6 days when both are in use, which is similar enough to the Sentinel-2 satellite's frequency that data from the two satellites can be combined and compared. In addition to the good resolution and high revisit frequency, the Sentinel-1 can

emit signals in either vertical (V) or horizontal (H) polarization, and receive in both polarizations. The direction of polarization refers to the orientation of the emitted wave's electric field. Different materials and surfaces react differently to V or H polarization. Sending and receiving signals in both polarizations gives information about the object being observed, and can improve monitoring of wetlands for instance [14].

Figure 2.1 shows an example of a multispectral optical satellite image and SAR image of the same location.

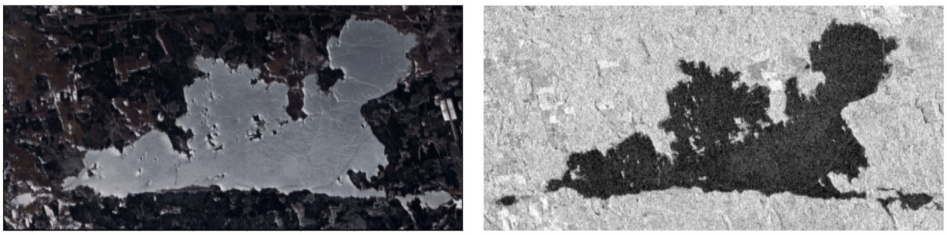

Figure 2.1: Lake Sottern. Left: multispectral image. Right: SAR image.

2.2 Deep Learning

In recent years, DL has increasingly been used to classify and map wetlands [6], because of its high performance on visual and multidimensional data obtained from remote sensing.

DL is a subset of machine learning techniques where multiple-layer models called artificial neural networks are built to learn features from raw data [16]. These networks are inspired by the functioning of biological brains, and are composed of layers of simple units, neurons. Each neuron performs a weighted sum of its inputs and passes it through an activation function. The activation function is most often a nonlinear function centered around zero, such as the Rectified Linear Unit (ReLU) [16]. If the activation function reaches a certain threshold, the neuron will pass on information. Training neural networks consists of updating the weights and biases for each of the neurons, with methods such as Stochastic Gradient Descent or Adam optimization [17].

These neural networks have different characteristics and performances based on how the neurons are connected, how many neurons they have, and

how many layers of neurons they have. The layers that are neither input or output layers are called hidden layers. Deep models can have several hundred hidden layers or more [18]. These networks can learn more advanced and high-level features than shallower networks and can be used in a wide range of applications, from natural language processing to object classification and segmentation [16], the latter of which is useful in wetland detection.

This section gives a summary of DL basics as well as background information about models and methods useful for detecting wetlands.

2.2.1 Convolutional Neural Networks

Convolutional Neural Networks (CNNs) are a type of deep neural network. While basic neural networks consist of fully-connected layers, where each neuron in a layer is connected to all the neurons in the previous layer and the next layer, CNNs also (sometimes exclusively) consist of convolutional and pooling layers. A convolutional layer applies a convolution to the input, which consists in sliding a small filter over the input data and computing a convolution: the dot product of the input and the filter value. This convolution enables the layer to learn features from the previous layer. Pooling layers then downsample the feature representations to merge semantically similar features [16].

These types of layers enable CNNs to work with multidimensional input, which makes them particularly well-suited for handling images. They have broken records in image classification and segmentation tasks ([19], [18], [20]) and have been used widely in image classification, object detection and semantic segmentation tasks [16]. Semantic segmentation is of particular interest for this project.

2.2.1.1 Semantic Segmentation

When automatically monitoring wetlands, one of the main tasks is to delineate where the wetland actually is on optical or SAR imagery. This is done by semantic segmentation, which consists of identifying an object in an image on a pixel-by-pixel basis. Each pixel is classified by the model as being part of the object of interest or not. Semantic segmentation requires deep learning models that can retain fine-grained details as well as separate general classes of objects.

2.2.1.2 U-Net

One of the best and most powerful models for semantic segmentation is U-Net [20], a CNN that has a U-shaped architecture. It was introduced in 2015 in the medical imaging field, but is today commonly used in many applications, one of them being monitoring wetlands. U-Net consists of a contracting path followed by an expanding path. The contracting path consists of convolutional layers and pooling layers that downsample the data. The expanding path upsamples the inputs through up-convolution in order to come back to the dimension of the original data [20]. Additionally, some layers of the contracting path are directly connected to layers of the expanding path. Feature maps from the contracting layers are copied, cropped, and concatenated to the expanding layers. These skip connections and the U-shaped architecture are the key to the model's powerful image segmentation [20]. Figure 2.2 illustrates the original U-Net architecture. Each green box corresponds to a feature map with multiple filters (their number is written on top). The spatial dimensions of the box are written at the lower left edge of the box. The white boxes represent cropped and copied feature maps.

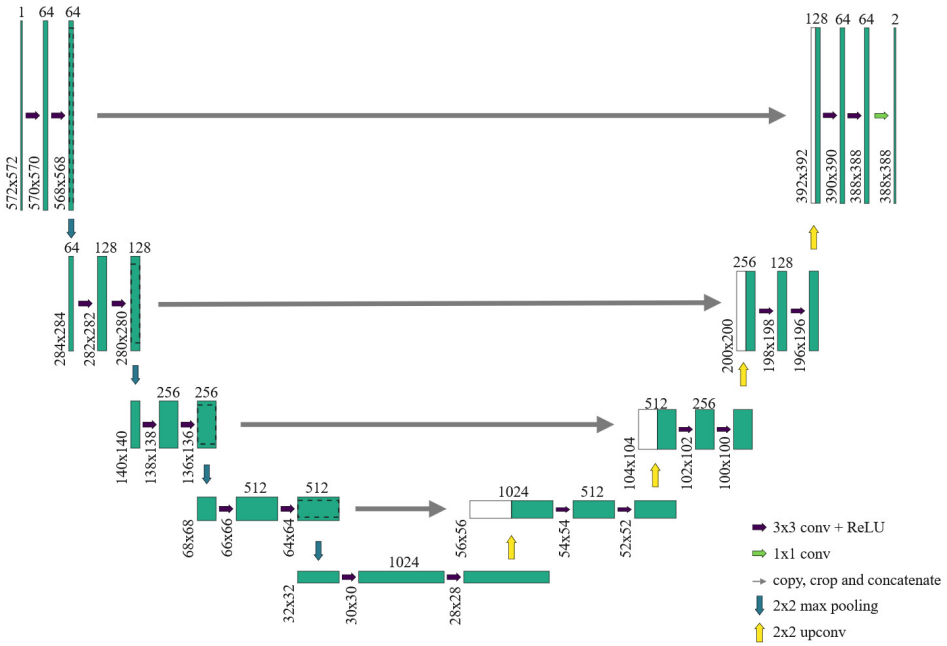

Figure 2.2: U-Net architecture

2.2.2 Learning Without Annotated Data

One of the biggest challenges of obtaining well-performing models is the quality and quantity of training data. These models, the U-Net included, need annotated data for training, because the training relies on having a ground truth to compare model predictions to (also called supervised learning). However, in fields such as remote sensing, there is a lack of big labelled datasets [21], [22], [6]. This is partly because labelling large amounts of data is very time-consuming.

However, there exists a DL framework that relies on the principle of learning from unlabelled data: unsupervised learning. Unsupervised learning is used for tasks such as data clustering or image generation.

Self-supervised learning is a subset of unsupervised learning. In self-supervised learning, the models create labels from the unlabelled data. There are several types of self-supervision, and they can be categorized into three groups: Generative, Predictive, and Contrastive [23]. Generative models learn representations from the data by recreating the input data. Predictive models predict specific properties of the data, by auto-generating labels and then training to predict these labels. Finally, contrastive learning consists of giving two similar inputs to a model and training the model to give them similar representations. Conversely, two dissimilar inputs should produce dissimilar representations. These image pairs are called positive and negative image pairs, respectively. A famous contrastive learning model is introduced in [24], where positive image pairs are automatically generated by performing data augmentation on one image. Contrastive learning is a powerful technique that can be used on its own or can provide models that have learned features that can then be used in downstream tasks. In this thesis, we focus on a particular case of contrastive learning: knowledge distillation.

Knowledge distillation, introduced by Hinton et al. [25], is the concept of transferring knowledge from one machine learning model to another. This is also referred to as teacher-student training, where the teacher is a pre-existing, often big model, and the student is a new, smaller model. Knowledge distillation makes use of the teacher model's predictions to guide the learning of the student model, instead of having the student model learn directly from data. The advantage of this method is that the student model learns the patterns and features that the bigger model has learned and then predicts similarly to the teacher model, but for a much smaller computational cost. Indeed, after knowledge distillation, the much smaller, simpler, and faster student model has an accuracy that is similar to the teacher model.

What makes distillation knowledge a type of contrastive learning is that the teacher model can be unsupervised, and thus create the labels that the student model needs to learn. This idea was introduced by Hu et al. [26], who showed that predictions from an unsupervised teacher model can be used to improve the accuracy of a supervised student model using cross-modal knowledge distillation. Thus, instead of distilling knowledge from a bigger model to a smaller model, it is from an unsupervised model to a supervised model. This method has successfully been used to monitor wetlands and remove the need of labelled training data by Peña et al. [7].

2.3 Data Normalization

When training a DL model, good data is essential to get good results. Data is usually chosen and collected according to the needs of the model. What constitutes ”good” data can vary from application to application, but most DL models benefit from modifying the raw data into data that is ready for training. This preprocessing can take several forms. For instance, data augmentation can artificially increase training data for a better robustness of the model; data filtering ensures that only the necessary aspects of the data are used; and data normalization homogenizes data that presents high variation.

2.3.1 Normalization in Medical Imaging

Normalization of data has several uses. It serves to homogenize and balance data. For multidimensional data, it can ensure that every dimension has equal weight during training. In RS, atmospheric conditions, incidence angle, and sensor quality and calibrations are all variables that influence the data. Satellite images and SAR images taken of the same location at similar dates may still be significantly different [27], [7], [28].

In SAR imagery, images depend on physical characteristics such as the polarization of the signal sent and received and the angle of incidence of the signal. These physical variations are normalized through standard preprocessing workflows [29]. However, system errors and other variations may still be present in the data even after preprocessing, which leads to heterogeneous data. Additional intensity normalization may be necessary depending on the task needed.

Research on additional normalization of SAR data is, to the best of our knowledge, very sparse, perhaps because SAR imagery contains important information that might be altered when normalizing the images. However,

in this thesis, we are only interested in separating water pixels from non-water pixels, and thus essentially treating the SAR images as two-dimensional greyscale pictures, which justifies using additional normalization methods to preprocess the data. We thus turn our attention to other fields were normalization is an important topic of research. For instance, in medical imaging, normalization is necessary for harmonization of images across patients and machines. Annotated datasets are often small, and harmonizing data from different centres is necessary to improve deep learning methods on medical imaging [30]. Some of the normalization methods in medical imaging can be applied to other domains. For instance, Magnetic Resonance (MR) images present similar characteristics to SAR images: they are obtained by using magnetic fields and radio waves that interact differently depending on the tissue, produce greyscale images, have varying intensities ranging from very low to very high, and can significantly differ based on calibration. Several methods used to preprocess and normalize MR images can thus theoretically be applied to SAR images. While deep learning based normalization is gaining in traction, the most commonly used normalization techniques are mathematical. These methods include min-max scaling, Z-score normalization, white stripe normalization, and histogram matching (or Nyúl normalization) [30], [31]. Comparisons of these methods across studies mark Z-score normalization and the Nyúl method as the most effective normalization methods [32], [30].

2.3.1.1 Z-score Normalization

Z-score normalization is a basic standardization technique that consists of standardizing the data to a mean of zero and standard deviation of one. The Z-score is calculated by

$$Z = \frac{X - \mu}{\sigma},$$

where Z is the Z-score normalized value, X is the original pixel value, and μ and σ are the mean and standard deviation of all the pixel values of the image.

Z-score normalization is commonly used in machine learning to ensure that different features are given the same importance for learning.

2.3.1.2 Nyúl Normalization

Histogram matching was introduced by Nyúl et al. to normalize intensities by using histogram landmarks such as percentiles [33], [34]. The normalization algorithm consists of two phases: training and transformation.

During the training phase, a set of training images is used to calculate a standard scale, which is the mean of so-called landmark values for each training image mapped to a predefined range. The choice of landmark values is up to the user, but a list that works well is $[1, 10, 20, ..., 80, 90, 99]$ where the numbers are the first and ninety-ninth percentiles along with the deciles [35]. During the transformation phase, an image is normalized by calculating the landmark values for that image, then performing piece-wise linear interpolation for each value of the image using the standard scale. The intensity values in the image that are below the first percentile and above the ninety-ninth percentile are discarded, to remove outliers.

Transformation is performed on any image that needs to be processed, and is fast (a few seconds at most). The improved Nyúl method is widely used for multi-centric data and image segmentation [36].

2.4 State of the Art

This section provides a brief overview of publications in the field of monitoring wetlands that use the tools discussed above: deep learning, remote sensing, and methods to deal with the challenge of the lack of large labelled datasets.

Indeed, during the last decades, the advent of massive RS data coupled with the difficulty of monitoring wetlands in person has led to a significant increase in studies on wetlands that make use of remote sensing and machine learning [5], [6], [3]. For the last ten years, efforts have shifted to use deep learning to classify, map, and monitor wetlands using a variety of models. Mahdianpari et al. [37] performed a study investigating the performance of various CNN models on wetland mapping and compared them to traditional ML techniques such as Support Vector Machines and Random Forests. They found that DL methods worked better than traditional ML methods, and that using more spectral bands for training improves results, likely due to the high intra-class and low inter-class variability of wetlands. Since then, DL and CNNs have been widely used for punctual mapping and classification of wetlands [3], with several studies making use of U-Net architectures [7], [38]. In recent years, other models such as vision Transformer networks or hybrid models have been used([39], [40], [41]). Finally, when tracking changes in

wetlands, or any other analysis that includes multitemporal data, recurrent neural networks have been used to account for temporal dependencies in the data [42].

Regardless of the type of DL architecture used, most researchers use similar data for the training of the models, mostly remote sensing data. While most studies use MSI or SAR data for classification [6], [3] there has been a rise in studies combining several types ([7] [42] [43] [44] [45]). The advantage of combining several types of data is that model accuracy can be improved, and that the limitations from each data type can be balanced by the other data types.

For instance, Hosseiny et al. developed WetNet [42], an ensemble learning method that took advantage of spatial, spectral, and temporal information of both MSI and SAR data to classify wetlands in Canada. They combined three submodels: a 3D CNN, a 2D CNN, and a Recurrent Neural Network. The Recurrent Neural Network can capture temporal dependencies. The three submodels and the use of multimodal data made it possible to perform a more complex and diverse feature extraction. The diverse feature extraction led to an improvement in classification accuracy over previous state-of-the-art models.

Finally, while the lack of large labelled wetland datasets is a hurdle to overcome for wetland monitoring on larger scales, there have been attempts to bypass the need for annotated data. Although these attempts are still in an early stage of research, some notable publications include Jamali et al. [39], Scheibenreif et al. [41], Ai et al. [46], and Peña et al. [7].

Jamali et al. developed a hybrid model to improve wetland classification accuracy in Canada. For wetland classes that had a low amount of samples, they used a Generative Adversarial Network to generate synthetic Sentinel-1/2 images, with a U-Net as discriminator. The synthetic images were combined with real images and sent into a transformer network that classified wetlands. The synthetic images inflated the training set and led to better classification scores of the model. While their paper proved the strength of synthetic data generation to improve model classification, their model is both costly in time and resources. Additionally, the model still requires annotated data as a base for the synthetic data generation.

Scheibenreif et al. explored self-supervision with transformers for segmentation and classification of land covers. They used large unlabelled datasets of Sentinel-1 and Sentinel-2 images to pretrain a self-supervised shifting-window vision (Swin) Transformer architecture with contrastive learning. The pretrained transformer was then finetuned with a hybrid Swin U-Net for segmentation and classification. The finetuning was done with a

labelled dataset that was ten times smaller than the unlabelled dataset. The combination of pretraining and finetuning showed better results than training models from scratch with only a labelled dataset. The paper proved that it is possible to obtain high classification and segmentation results without the need for a large annotated dataset. However, the model had worse results for wetland and grassland classes than other land cover classes, and still requires an annotated dataset for finetuning.

Ai et al. used transfer learning to develop a model for wetland vegetation mapping. They constructed a fully-connected deep neural network specialized in mapping lake-floodplain wetland vegetation with few training samples, and then used transfer learning to map wetland vegetation annually. They compared their deep neural network to Support Vector Machines and Random Forests. They found that transfer learning and using deep neural networks improves the mapping, and solves the problem of scarce input data for annual mapping, since their model requires much fewer training samples than CNNs. However, their model did not reach mapping accuracy levels of CNNs such as U-Net. This paper illustrates that even though other machine learning models can be trained on little data, the best performing ones are still CNNs, and need sufficiently large training sets.

Finally, Peña et al. used knowledge distillation where the unsupervised teacher model automatically generated training data with labels for the student model, which learned through self-supervision (see Figure 2.3). The teacher model generated ground truth masks for water pixels from MSI Sentinel-2 data. The ground truth masks were then used as labels for the Sentinel-1 SAR data used to train the student model. The resulting model, DeepAqua, had very high mapping scores for wetlands in Sweden and is state-of-the-art for self-supervised wetland detection. The model proved that learning without annotated data is possible and leads to high accuracy scores. However, due to minimal data preprocessing and a limited training set, it did not generalize well to unseen data from years it was not trained on.

To summarize, the best wetland mapping model that uses fully self-supervised learning and truly unlabelled data is DeepAqua. It is the only wetland mapping model that does not require any human-labelled data, and the only model that uses training data from Swedish wetlands. This model is thus best suited for research on wetlands in Sweden. However, it suffers from a lack of generalizability, which is what this thesis aims to solve.

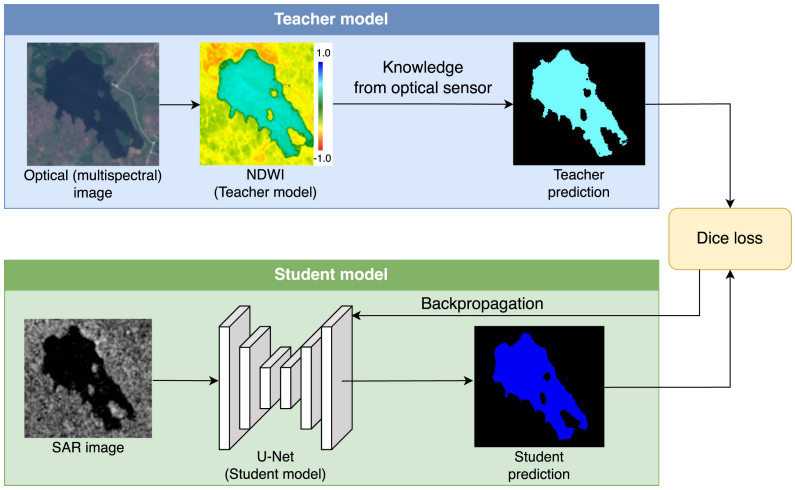

Figure 2.3: DeepAqua model architecture. Image taken from [7] and used under CC BY 4.0 DEED license.

Chapter 3

Method and Experiments

The aim of this project is to modify the existing DeepAqua model in order to make it more robust to unseen data. The research was done in an explorative manner, by implementing and evaluating several normalization methods. During the exploration phase, the data was examined and several mathematical normalization methods were tested, as well as using a DL model for normalization. Experiments with data augmentation were also performed. However, due to time limitations, we finally focused only on the most promising method: mathematical normalization. The entire research was done in several steps: collecting the SAR images and ground truth masks, building the student model, and testing the normalization methods. We trained and evaluated the models to verify that the normalization methods worked on the 2020 data. Finally, we tested all models on unseen data from a wider span of years to further test robustness.

In this section, we go through each step of the process. First, we describe the data that is collected and how. Second, we explain the model architecture in depth. Third, we present the evaluation metrics that were used to compare models. Fourth, we explain which normalization methods we tested as well as how they were implemented. Finally, we provide an overview of the experiments that were performed.

3.1 Data

3.1.1 Data Collection

Data collection is done using scripts on GEE [9]. VH-polarized SAR Sentinel-1 and multispectral Sentinel-2 images of the same region and same day are downloaded. The day and region are chosen carefully to ensure that there is less than 1% cloud cover on the area of interest. This restriction ensures that the ground truth masks can be generated correctly. The ground truth masks are generated by calculating the Normalized Difference Water Index (NDWI) on the Sentinel-2 images.

NDWI is an index to measure water content. It was introduced in 1996 by McFeeters [47] to detect water in RS imagery. The formula is as follows:

$$NDWI = \frac{(GREEN - NIR)}{(GREEN + NIR)},$$

where $GREEN$ is the green light wavelength of the object of interest, and NIR is the near-infrared wavelength of the object of interest. The index produces values from -1 to 1. In this thesis, a threshold of 0 is used: pixels with positive values are classified as water. The mask is thus a binary image. This ground truth mask is automatically generated from the green and near-infrared bands of the Sentinel-2 images while downloading the data from GEE. This step corresponds to passing the multispectral image through the teacher model. The process is illustrated in Figure 3.1.

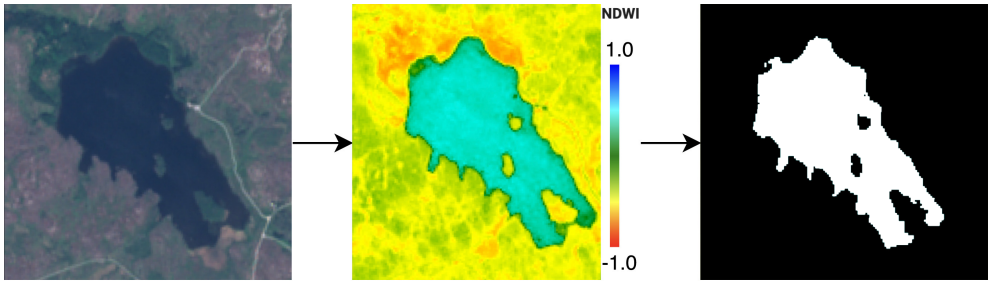

Figure 3.1: Ground truth mask generation using NDWI. The left image is a multispectral image. The middle image is generated using the NDWI formula. The right image shows the binary NDWI mask where pixels with positive values are white, and the rest are black. Image taken and adapted from [7] and used under CC BY 4.0 DEED license.

For each image, the resolution is 10 meters per pixel. For the DeepAqua paper, training data was collected from the entire county of Örebro, on dates 2018-07-04 and 2020-06-23. We use the same datasets for this thesis.

3.1.2 Data Processing

After collecting the data, it must be preprocessed before being fed to most of the student models.

The SAR images are preprocessed by removing outlier pixel values: values below the 1^{st} and above the 99^{th} percentiles are discarded. The remaining pixel values are min-max normalized to a $[0, 1]$ scale. Each pixel in the processed SAR images thus has a singular intensity value. For the Nyúl normalization, the SAR images are not preprocessed, since the Nyúl normalization algorithm handles outliers already.

The NDWI masks and corresponding SAR images are then split into 64x64 pixel tiles that are used to train the student model. Each dataset consists of 45 498 tile pairs.

The tile pairs are randomly separated into training and validation datasets: 80% of the pairs is used for training and 20% for validation. The split is the same for the different years, in order to compare the validation results across years.

Figure 3.2 shows an example of SAR and NDWI mask tiles from the 2018 dataset.

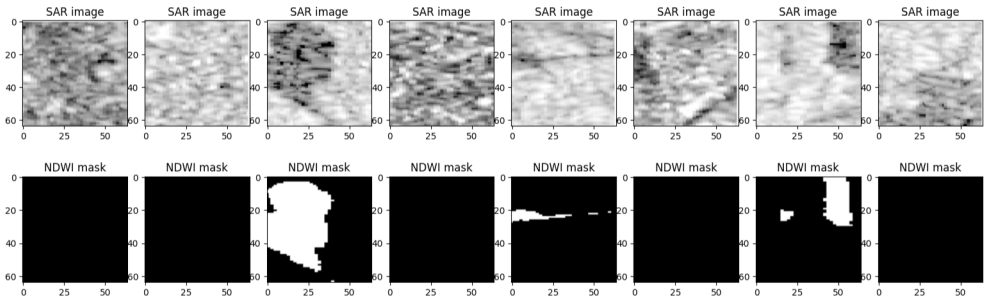

Figure 3.2: Random tile pairs from the 2018 dataset

For the entire training dataset, the proportion of water pixels is 11%.

3.1.3 Test data

Test data from the original DeepAqua paper is used, for comparison and convenience. The test dataset consists of 99 image pairs of three wetlands

located in Sweden: Svartådalen, Hjälstaviken and Hornborgasjön with manually created ground truth masks. Each wetland image set consists of monthly images taken between 2018 and 2022, excluding winter months (November-March) to avoid images with ice and snow. We preprocess the SAR images by discarding the 1^{st} and 99^{th} percentiles (taken from the 2018 and 2020 training sets) and min-max normalizing the values to lie between 0 and 1. Then, we split the SAR images and ground truth masks into 64x64 pixel tiles. As for the training set, we keep the raw SAR images for the model that uses Nyúl normalization. The final amount of tile pairs is 17 210. For the entire test dataset, the proportion of water pixels is 23%.

3.2 Model

3.2.1 Model Implementation

The model is implemented from scratch using PyTorch. The main part of the model is the student model, as mentioned in 2.4. The student model is based on U-Net, with several key changes. Firstly, we introduce batch normalization after each convolution during training. This enhances the learning and performance of the model. Secondly, we add zero-padding with 1 pixel border in the convolutions to preserve the spatial size of the image. As a consequence, it is not necessary to crop the encoding layers before concatenation. Finally, we use images of size 64x64 pixels and start with 32 filters in the first encoding layer. The details of the architecture are shown in Figure 3.3. Note the differences in dimensions and operations compared to the original U-Net model from Figure 2.2.

The base model is trained using the Adam optimizer and with a sigmoid function as the last activation function.

The model output is a segmentation mask. This mask is compared to the ground truth NDWI mask with the Dice Loss [48]. The Dice Loss is a loss that is useful for evaluating segmentations, especially when there is a class imbalance, as it prioritizes overlap between prediction and ground truth. In this project, the classes are water and land, which are represented by values 1 and 0, and the water class is significantly smaller than the land class. All pixels from the prediction output that have an intensity value over 0.5 are considered water, and all pixels with an intensity value under 0.5 are considered land. The Dice Loss is calculated as follows:

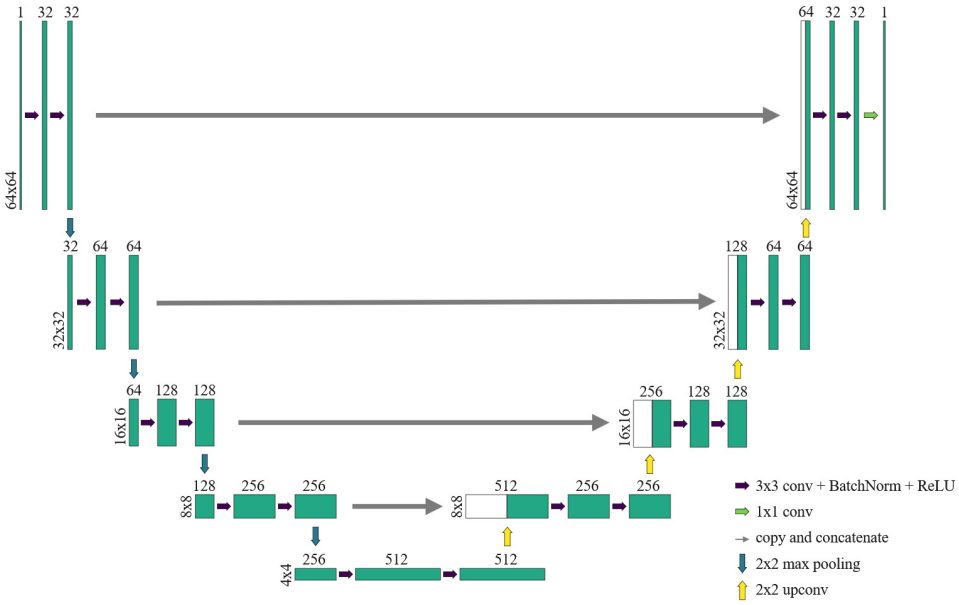

Figure 3.3: DeepAqua student model architecture

$$L_{Dice} = 1 - \frac{2 \cdot |Y_P \cap Y_T| + \epsilon}{|Y_P| + |Y_T| + \epsilon}$$

where ϵ is a small constant to avoid zero-division, and Y_P and Y_T are, respectively, the prediction output set and the teacher output set. $|Y_P|$ and $|Y_T|$ are the cardinalities of the sets.

3.3 Evaluation Metrics

Several evaluation metrics are used to judge the performance of the model. Since there is a class imbalance, it is important to use metrics that can account for that imbalance and give meaningful information. The primary metric that is used for evaluation is thus the Intersection over Union (IoU) metric, which measures how much overlap there is between the predicted segmentation area and the ground truth segmentation area.

$$IoU = \frac{TP}{TP + FP + FN},$$

where TP are the true positives, FP are the false positives, and FN are the false negatives. In this project, IoU is calculated for each image and then averaged

over the entire dataset.

The other metrics that we use are the same as the ones used in [7], in order to compare our model to DeepAqua.

Pixel accuracy gives a general overview of the model's performance, by counting correctly classified pixels. However, in the case of this project, the class imbalance means that using solely pixel accuracy is not a good metric.

$$\text{Pixel Accuracy} = \frac{TP + TN}{TP + TN + FP + FN},$$

where TP and TN are the true positives and negatives respectively, FP and FN are the false positives and negatives respectively.

Precision measures how many of the positive predictions were actually correct.

$$\text{Precision} = \frac{TP}{TP + FP}$$

Recall measures how many of the actual water pixels were correctly identified.

$$\text{Recall} = \frac{TP}{TP + FN}$$

The F1-score combines Precision and Recall to give a balanced measure of the model's performance:

$$\text{F1-score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

3.4 Normalization Techniques

3.4.1 Mathematical Normalization Techniques

The two mathematical methods for normalization are chosen according to the literature review, because they have been proven to work well for medical imaging [32], [30], and because they can theoretically be used in a RS context.

The algorithm for Nyúl normalization is separated into two parts, as explained in 2.3.1: training and inference.

Let $[m_{1,i}, p_{10,i}, p_{20,i}, \ldots, p_{90,i}, m_{99,i}]$ be the landmarks for an image i. The algorithms are explained in high level pseudocode below.

Listing 3.1: Training the Nyúl algorithm

Input: A set $S = \{X_1, X_2, ..., X_N\}$ of N images,
the range for the standard scale: $[m_{min}, m_{max}]$,
the list of desired landmarks.

for $\{i = 1, ..., N\}$ **do**
Obtain the landmarks of the image.
for $\{j = 10, 20, 30, ..., 90\}$ **do**
//Map the landmarks to the standard range
$$p'_{j,i} = m_{min} + (p_{j,i} - m_{1,i}) \left(\frac{m_{max} - m_{min}}{m_{99,i} - m_{1,i}} \right)$$
end for
Add the mapped landmarks to the standard scale.
end for
standard scale ← mean of the landmarks

Output: The standard scale $[m_1^s, p_{10}^s, p_{20}^s, ..., p_{90}^s, m_{99}^s]$

Listing 3.2: Nyúl normalization algorithm

Input: An image X, the standard scale, the list of desired landmarks.

Obtain the landmarks of the image.
for *{every intensity value x in the image}* **do**
Linearly map x to the standard scale by mapping from $[m_{i,X}, m_{j,X}]$ **to** $[m_i^s, m_j^s]$, where $i, j \in \{1, 10, 20, .., 80, 90, 99\}$, j is the next value in the set that is greater than i, and $m_{i,X}$, $m_{j,X}$ are the closest lower and upper decile values to x.
end for

Output: The normalized image X'

In medical imaging, the Nyúl normalization process is usually done on entire images of the same body part. In the case of this thesis, we are looking at

small highly variable tiles of one big image (Örebro county). For that reason, the algorithm is slightly modified: the raw 2018 tiles are used to obtain the standard scale, and each corresponding 2020 tile is normalized to match the histogram of the 2018 tile. The 2018 tile is normalized as well, to account for the slight distortion that the piece-wise linear interpolation of the pixel values adds to the images. The 2018 and 2020 tiles are thus both equally transformed.

For the Z-score normalization, the training process is similar. We collect the mean and standard deviation of the entire Örebro training set and use these values to normalize the tiles using the formula presented in 2.3.1. The 2018 tiles are thus all normalized according to the entire 2018 set, and the 2020 tiles are all normalized according to the entire 2020 set.

Below, we illustrate how both normalization techniques alter the intensity histograms of SAR data. We take four images of lake Sottern (Figure 3.4 and perform Nyúl normalization (Figure 3.5) as well as Z-score normalization (Figure 3.6) on the images.

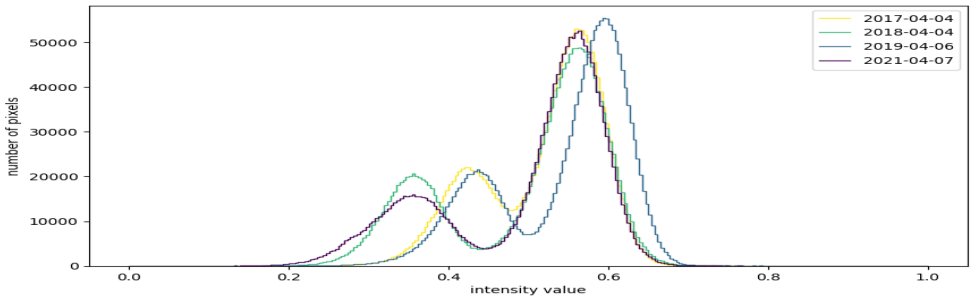

Figure 3.4: Original intensity histograms

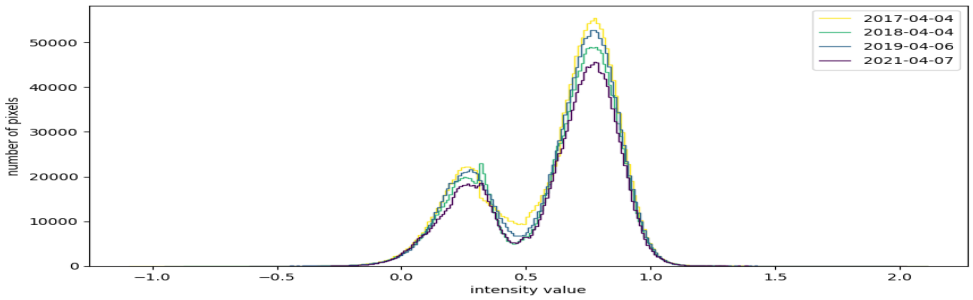

Figure 3.5: Nyúl normalized intensity histograms

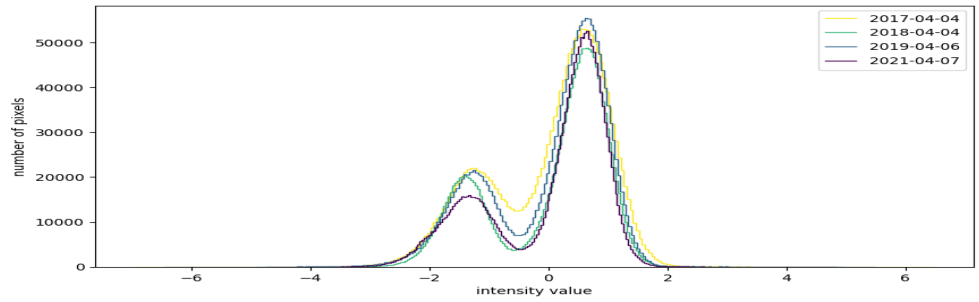

Figure 3.6: Z-score normalized intensity histograms

3.5 Experimental Design

The experiments are conducted in two stages: first, testing of two mathematical normalization techniques: Z-score normalization and Nyúl normalization. A comparison of their effectiveness compared to the DeepAqua model is made, using the same hyperparameters for each model. Then, hyperparameter sweeps are performed for the models with normalization in order to find the best performing models.

All experiments are conducted using NVIDIA T4 GPUs from the Alvis cluster provided by NAISS [49], and a personal laptop with NVIDIA GeForce RTX 4070 GPU. Base DeepAqua model training takes approximately 25 minutes for 20 epochs.

3.5.1 Initial Experiments

The first experiments compare the base DeepAqua with two other models: DeepAqua with Z-score normalization and DeepAqua with Nyúl normalization. All three models are trained on data from 2018. A second DeepAqua model is trained on the 2020 dataset, as an additional benchmark.

All four models are trained for 20 epochs, with batch size 32, 32 input feature channels, and starting learning rate $5 \cdot 10^{-5}$, according to the specifications of DeepAqua in [7].

After the models are trained, validation with data from 2020 is performed, except for the DeepAqua trained on the data from 2020, where we perform validation with 2018 data instead.

3.5.2 Hyperparameter Sweep

To find the best combination of hyperparameters for the models using normalization techniques (we refer to them as Nyúl DeepAqua and Z-score DeepAqua), we then perform a hyperparameter sweep. In order to optimize time and resources used, we choose to perform a random hyperparameter sweep. Indeed, random hyperparameter sweeps are more efficient than grid searches, especially when parameter importance is unequal [50].

We vary three hyperparameters: learning rate, batch size, and the number of input filters (also called feature maps, or channels in this thesis) in the first layer. Learning rate mainly impacts how fast the model trains, but a wrong learning rate might mean that the loss is only locally minimized instead of globally. Batch size impacts how many images the model can see at once, and impacts how stable the convergence towards a minimum loss is, as well as how well the model converges. The amount of filters impacts how many diverse features the model can learn at each level. However, a bigger amount of filters increases the number of parameters of the model and thus increases training time. We randomly vary the hyperparameters around the values chosen for DeepAqua in [7], based on the assumption that since the models share the same architecture, the optimal hyperparameters will not be very different from the values chosen for DeepAqua.

We vary the learning rate randomly between values $[10^{-6}, 5{\cdot}10^{-6}, 10^{-5}, 5{\cdot}10^{-5}, 10^{-4}, 5 \cdot 10^{-4}, 10^{-3}]$. We vary the batch size randomly between values $[8, 16, 32, 64, 128]$. We vary the number of filters randomly between values $[4, 8, 16, 32, 64]$.

We choose one random split into training and validation and perform 50 iterations on that fixed split. This ensures that the differences in model performance do not depend on the random split into training and validation, and only on the hyperparameter changes.

We perform the hyperparameter sweep on Python with the help of the WandB library for ease of vizualisation and grouping of the runs [51]. According to the sweep results, the best hyperparameter combinations are chosen for the Nyúl DeepAqua and the Z-score DeepAqua.

3.5.3 Testing

Finally, we test all models on the entire test dataset and compare their results. This is done by separating the test dataset into images from before 2020 and images from 2020 and onwards, and comparing model results. This separation is chosen to mirror the separation noted in the DeepAqua paper [7] , where

DeepAqua trained on 2018 data performed well on 2018 and 2019 test data but worse on the other years. In the paper, a separate DeepAqua was trained on 2020 data to do mapping on years 2020-2022.

Chapter 4

Results and Discussion

In this chapter, we present the results of the experiments and discuss them.

4.1 Data

As mentioned in Chapter 1, the SAR data presents significant differences between years 2018 and 2020 due to the sensor calibration changes. Visually, the data from 2018 is much noisier, and the contrast between land and water is smaller than for the data from 2020. Figure 4.1 shows preprocessed SAR tiles from 2018 and 2020. While the tiles are from the exact same location, they are visually very different, and their intensity histograms are different as well. Figure 4.2 shows the intensity histograms of the tiles.

Further examination of random tiles with only water pixels and only land pixels shows a clear difference in the distribution of intensity for water pixels, while the difference is much less marked for land pixels (see Figures 4.3-4.6). We can see on Figure 4.5 that the intensity distribution of water pixels is much more concentrated for the 2020 dataset, and after preprocessing of the tiles, there is a left shift of the curve for the 2020 dataset (Figure 4.6). An explanation for this left shift is due to the preprocessing: the values of the 1^{st} and 99^{th} percentiles are different for 2018 and 2020. While the 99^{th} percentiles of both datasets are close, the 1^{st} percentiles are very different, with the 2020 1^{st} percentile being bigger (see Table 4.1). While preprocessing the SAR images, a left shift is thus introduced for the 2020 dataset compared to the 2018 dataset.

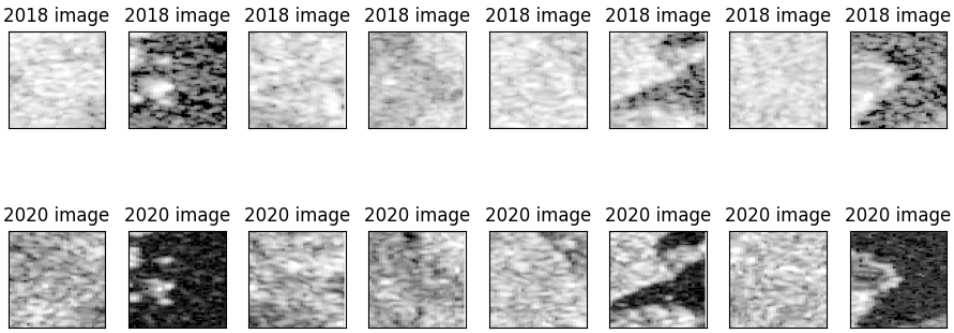

Figure 4.1: Random SAR tiles from training sets

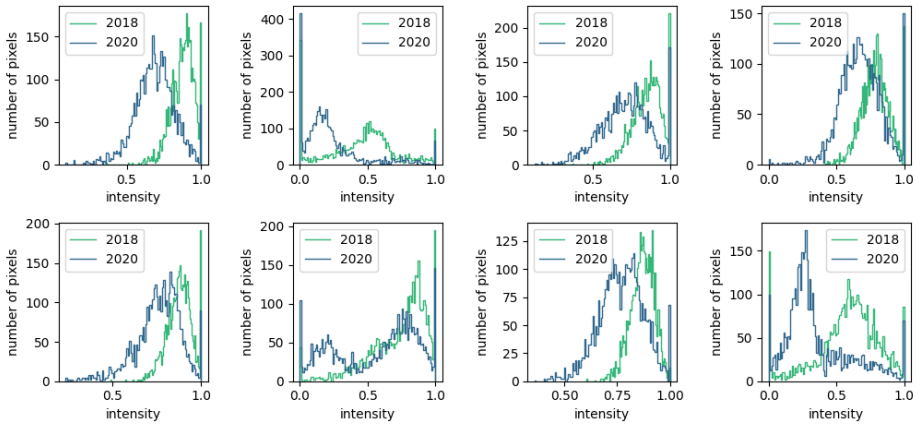

Figure 4.2: Intensity histograms of the SAR tiles

Table 4.1: First and ninety-ninth percentile values for raw SAR datasets

Datasets	1^{st} **percentile**	99^{th} **percentile**
2018	-48.59	-10.45
2020	-30.27	- 9.968

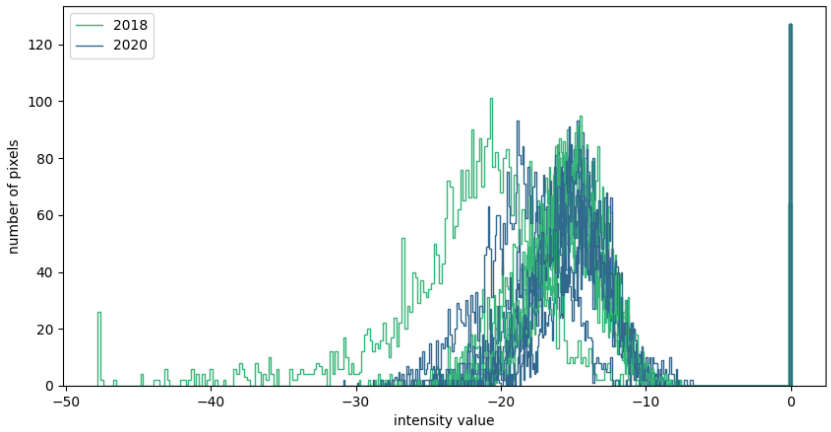

Figure 4.3: Intensity histograms of raw land tiles

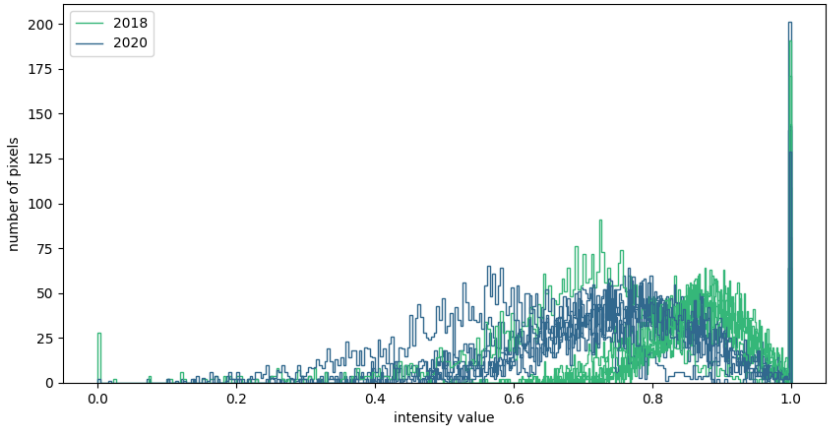

Figure 4.4: Intensity histograms of preprocessed land tiles

4.2 Initial Experiments

4.2.1 Comparison With Base Models

We run each model 10 times with different random division of the data into training and validation sets. Figures 4.7 - 4.14 show the validation loss and IoU of the four models: base DeepAqua, DeepAqua trained on 2020 data (DeepAqua 2020), DeepAqua with Nyúl-normalized data (Nyúl DeepAqua), and DeepAqua with Z-score normalized data (Z-score DeepAqua).

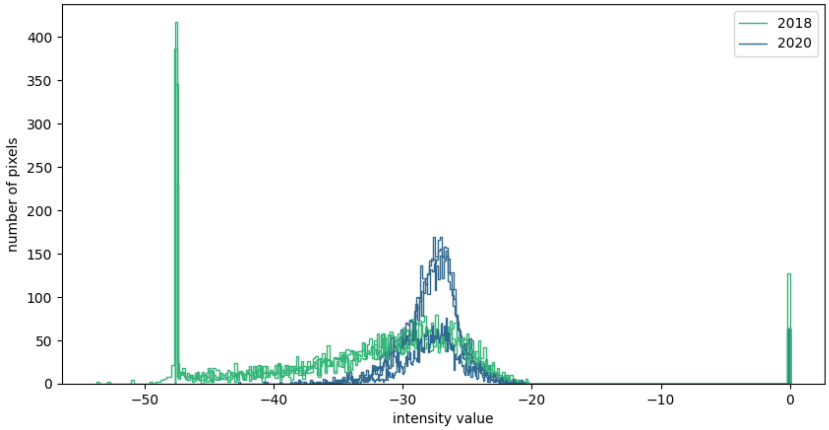

Figure 4.5: Intensity histograms of raw water tiles

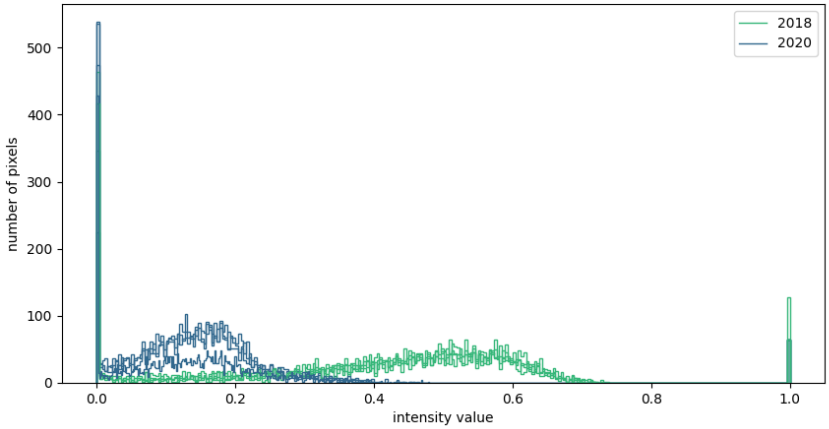

Figure 4.6: Intensity histograms of preprocessed water tiles

While all models converge within 20 epochs, there is a difference in stability between the base DeepAqua models and the models with normalization, especially for Nyúl DeepAqua. The loss and IoU curves are noisier for both Nyúl DeepAqua and Z-score DeepAqua, with Nyúl DeepAqua presenting the most variation for the IoU, both between and within runs.

After running each model 10 times, the evaluation metrics are calculated and averaged. The results of their performance on the 2018 dataset are presented in Table 4.2.

Table 4.3 shows their performance on the 2020 dataset.

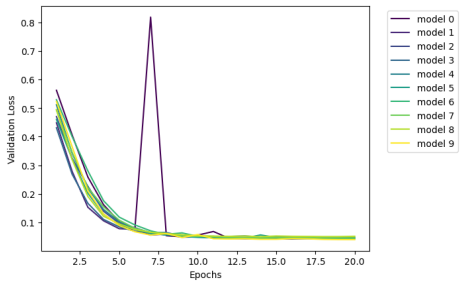

Figure 4.7: Validation loss for DeepAqua models (on 2018)

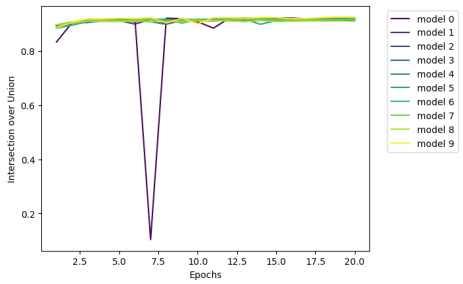

Figure 4.8: IoU for DeepAqua models (on 2018)

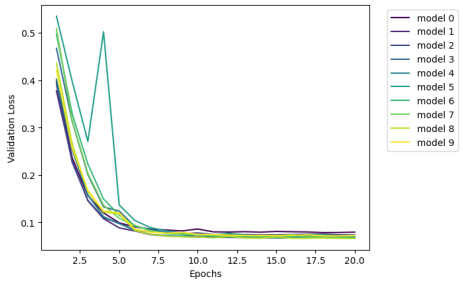

Figure 4.9: Validation loss for DeepAqua 2020 models (on 2020)

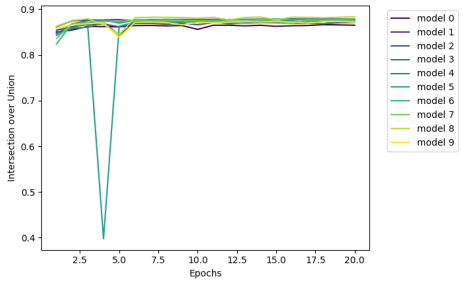

Figure 4.10: IoU for DeepAqua 2020 models (on 2020)

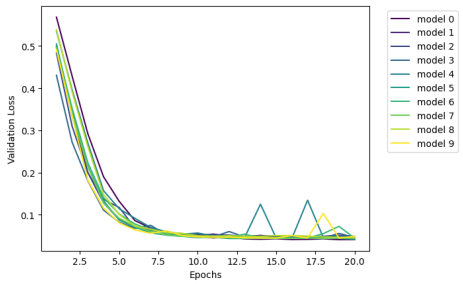

Figure 4.11: Validation loss for Nyúl DeepAqua models (on 2018)

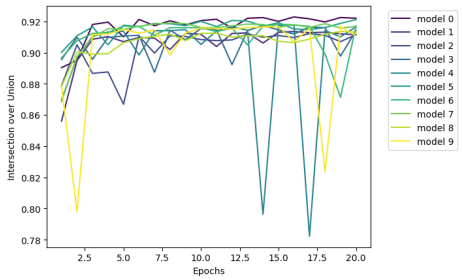

Figure 4.12: IoU for Nyúl DeepAqua models (on 2018)

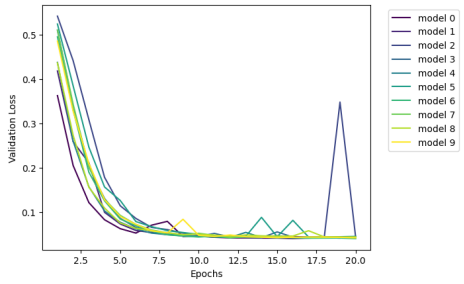

Figure 4.13: Validation loss for Z-score DeepAqua models (on 2018)

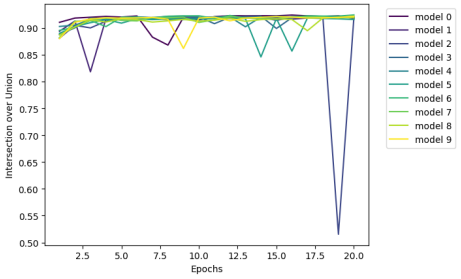

Figure 4.14: IoU for Z-score DeepAqua models (on 2018)

Table 4.2: Model performances on 2018 validation data (mean $\pm$ standard deviation)

Models	IoU	Precision	Recall	F1-score	Pixel Accuracy
DeepAqua 2018	0.93 ±0.0018	0.96 ±0.0018	0.97 ±0.0019	0.97 ±0.00094	0.99 ±0.00022
DeepAqua 2020	0.044±0.014	0.98 ±0.0071	0.044 ±0.014	0.084 ±0.014	0.90 ±0.0039
Nyúl DeepAqua	0.93 ±0.0019	0.97 ±0.0035	0.97 ±0.0027	0.96±0.0010	0.99 ±0.00020
Z-score DeepAqua	0.93 ±0.0010	0.97 ±0.0021	0.97 ±0.0017	0.97 ±0.00055	0.99 ±0.00012

Table 4.3: Model performances on 2020 validation data (mean $\pm$ standard deviation)

Models	IoU	Precision	Recall	F1-score	Pixel Accuracy
DeepAqua 2018	0.54 ±0.032	0.55±0.034	0.97 ±0.0025	0.70 ±0.028	0.91±0.012
DeepAqua 2020	0.89 ±0.0034	0.95 ±0.0032	0.95 ±0.0032	0.94 ±0.0019	0.99 ±0.00034
Nyúl DeepAqua	0.89 ±0.0028	0.94 ±0.0056	0.94 ±0.0041	0.94 ±0.0016	0.99 ±0.00040
Z-score DeepAqua	0.87 ±0.014	0.91 ±0.0053	0.95 ±0.018	0.93±0.0080	0.99 ±0.0014

As we can see in Tables 4.2 and 4.3, the pixel accuracy is not enough to judge whether a model detects water correctly. We instead focus on the IoU metric throughout the results section. The base DeepAqua model struggles significantly with correctly mapping water in the 2020 dataset, with an IoU that is below 60%. Inspecting the other metrics shows that the model's precision suffers, which means that the model has many false positives. Similarly, DeepAqua 2020 performs very poorly on the 2018 dataset (IoU below 1%). The model struggles with recall: there are too many false negatives. These drops in precision and recall for the base models is due to the difference in the 2018 and 2020 datasets, since the 2018 pixel values are generally higher compared to the 2020 pixel values. This leads to DeepAqua 2018 mistakenly classifying land pixels as water in the 2020 dataset, and DeepAqua 2020

mistakenly classifying water pixels as land in the 2018 dataset. However, both normalization methods work well: both the Nyúl-normalization and Z-score normalization lead to models that perform well on the 2018 and 2020 datasets. Both Nyúl DeepAqua and Z-score DeepAqua perform as well as DeepAqua on the 2018 validation set (IoU of 0.93). Nyúl DeepAqua reaches the same IoU score as DeepAqua 2020 on the 2020 validation set, while Z-score DeepAqua performs slightly worse (IoUs of 0.89 and 0.87 respectively).

To conclude, the base DeepAqua is outperformed by both normalization techniques and the 2020 DeepAqua model on the 2020 validation set. The maximum IoU score reached is lower for the 2020 dataset than for the 2018 dataset. There is no great difference between the normalization techniques in term of IoU results, which justifies continuing testing with both normalization techniques during the hyperparameter finetuning.

4.2.2 Hyperparameter Sweeps

4.2.2.1 Nyúl normalization results

Figure 4.15 represents the best validation IoUs for each combination of hyperparameters. A few trends can be seen: most models perform similarly, but batch size and number of input channels influences the final IoU score. The lowest batch size leads to worse performance. Too few channels (4, 8) and too many channels (64) lead to worse performance as well. However, the learning rate does not seem to matter as much as the two other hyperparameters. Runs with a very low learning rate and large batch size have a worse IoU, but examination of these particular runs leads to the conlusion that these runs did not have time to converge in 20 epochs. Detailed results of the sweep are shown in Tables 4.4, 4.5, and 4.6.

Table 4.4: Best validation IoU mean and standard deviation depending on batch size for the Nyúl DeepAqua sweep

Batch size	**IoU**
8	0.79 ±0.017
16	0.89 ±0.0058
32	0.82 ±0.27
64	0.85 ±0.23
128	0.81 ±0.31

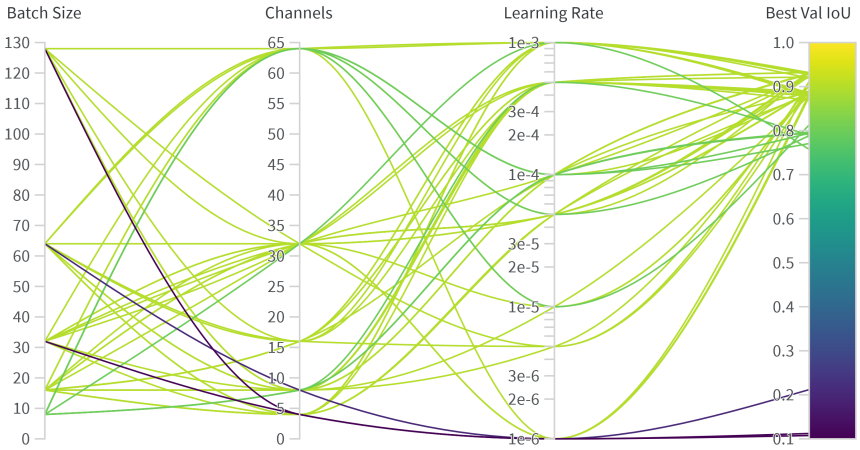

Figure 4.15: Parallel coordinates plot for the Nyúl DeepAqua Sweep

Table 4.5: Best validation IoU mean and standard deviation depending on input channels for the Nyúl DeepAqua sweep

Input channels	**IoU**
4	0.70 ±0.37
8	0.82 ±0.21
16	0.90 ±0.021
32	0.89 ±0.046
64	0.86 ±0.066

Table 4.6: Best validation IoU mean and standard deviation depending on learning rate for the Nyúl DeepAqua sweep

Learning rate	**IoU**
10^{-6}	0.52 ±0.41
$5 \cdot 10^{-6}$	0.90 ±0.010
10^{-5}	0.87 ±0.038
$5 \cdot 10^{-5}$	0.90 ±0.046
10^{-4}	0.86 ±0.067
$5 \cdot 10^{-4}$	0.89 ±0.043
10^{-3}	0.89 ±0.055

While the results confirm the intuitions gained from the parallel

coordinates plot, we can see that for batch size and and number of channels, there can still be a large variation from the mean IoU obtained. For the learning rate, the variation in IoU for each value is smaller, with both low learning rates and high learning rates leading to top results.

The best run had the hyperparameters: batch size 128, 32 input channels, and learning rate $5 \cdot 10^{-5}$, and achieved a validation IoU of 0.93, which is the same as the mean validation IoU obtained with the base DeepAqua hyperparameters. We conclude that while some hyperparameter choices lead to significantly worse performance, there are many combinations that yield the best results, which is an IoU of around 0.93. This might be due to the dataset being relatively small, as well as the strength of the U-Net model architecture. In addition, the IoU of 0.93 corresponds to the score of the base DeepAqua, which was optimized.

For the final testing, we choose to use the following hyperparameters: batch size 128, 32 input channels, and learning rate $5 \cdot 10^{-5}$. Ten models were trained with these hyperparameters. Table 4.7 shows evaluation metric scores on 2018 and 2020 validation data.

Table 4.7: Final Nyúl model performance on validation data (mean $\pm$ standard deviation)

Year	IoU	Precision	Recall	F1-score	Pixel Accuracy
2018	0.93 ±0.0021	0.96±0.0038	0.97 ±0.0028	0.96±0.0011	0.99±0.00025
2020	0.89±0.0034	0.94 ±0.0040	0.94 ±0.0037	0.94±0.0019	0.99 ±0.00045

The final metrics are very close to the original Nyúl DeepAqua run, with the standard deviation a little lower for precision and recall on the 2020 data.

4.2.2.2 Z-score normalization results

Figure 4.16 represents the best validation IoUs for each combination of hyperparameters. Similarly to the Nyúl DeepAqua sweep, a low batch size leads to worse performance. However, the highest batch size yields relatively worse results as well. Another trend is that a higher number of input channels leads to better results. The importance of learning rate is less clear, but a higher learning rate seems to lead to better results. Detailed results of the sweep are shown in Tables 4.8, 4.9, and 4.10.

Contrary to the Nyúl DeepAqua sweep, the Z-score sweep showcases less standard deviation of the IoU within hyperparameter category. We can see that a batch size of 32 or 64 yields best results. The highest number of input

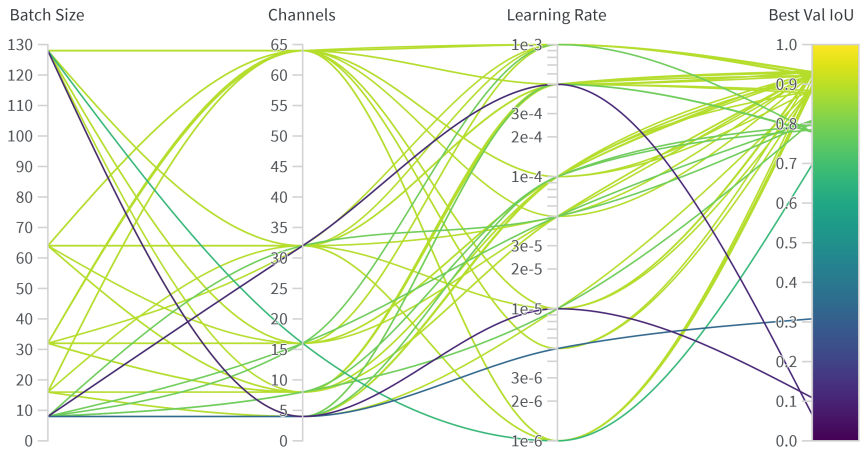

Figure 4.16: Parallel coordinates plot for the Z-score DeepAqua Sweep

Table 4.8: Best validation IoU mean and standard deviation depending on batch size for the Z-score DeepAqua sweep

Batch size	**IoU**
8	0.67 ±0.26
16	0.89 ±0.0049
32	0.92 ±0.0013
64	0.92 ±0.014
128	0.84 ±0.23

Table 4.9: Best validation IoU mean and standard deviation depending on input channels for the Z-score DeepAqua sweep

Input channels	**IoU**
4	0.69 ±0.31
8	0.89 ±0.054
16	0.87 ±0.091
32	0.82 ±0.25
64	0.91 ±0.017

channels yields the best results on average as well. Finally, a learning rate between $5 \cdot 10^{-5}$ and 10^{-3} yields the best results.

The best run had the hyperparameters: batch size 128, 32 input channels,

Table 4.10: Best validation IoU mean and standard deviation depending on learning rate for the Z-score DeepAqua sweep

Learning rate	IoU
10^{-6}	0.86 ±0.082
$5 \cdot 10^{-6}$	0.71 ±0.35
10^{-5}	0.76 ±0.32
$5 \cdot 10^{-5}$	0.89 ±0.059
10^{-4}	0.89 ±0.055
$5 \cdot 10^{-4}$	0.89 ±0.25
10^{-3}	0.90 ±0.055

and learning rate $5 \cdot 10^{-4}$, and achieved a validation IoU of 0.93. However, other runs achieved very similar scores. Once again, the IoU score is the same as for the base DeepAqua hyperparameters, and the same score than for the original run of Z-score DeepAqua. We draw the same conclusions as for the Nyúl sweep. We theorize that the IoU of 0.93 is likely the highest score that DeepAqua can attain with this training set.

For the final testing, we choose to use the following hyperparameters: batch size 32, 64 input channels, and learning rate $5 \cdot 10^{-5}$. Ten models were trained with these hyperparameters. Table 4.11 shows evaluation metric scores on 2018 and 2020 validation data.

Table 4.11: Final Z-score model performance on validation data (mean $\pm$ standard deviation)

Year	IoU	Precision	Recall	F1-score	Pixel Accuracy
2018	0.93±0.0035	0.96 ±0.0030	0.96±0.0030	0.96±0.0031	0.99 ±0.00031
2020	0.87±0.0057	0.91 ±0.0055	0.95 ±0.0033	0.93±0.0033	0.99±0.00061

The results are very close to the original Z-score run, with a slightly lower standard deviation for the metrics on the 2020 data.

4.3 Testing

Results with the test dataset are presented in Tables 4.12 and 4.13 below. Figures 4.17 and 4.18 show examples of the Nyúl and Z-score models predictions on the same test data tiles.

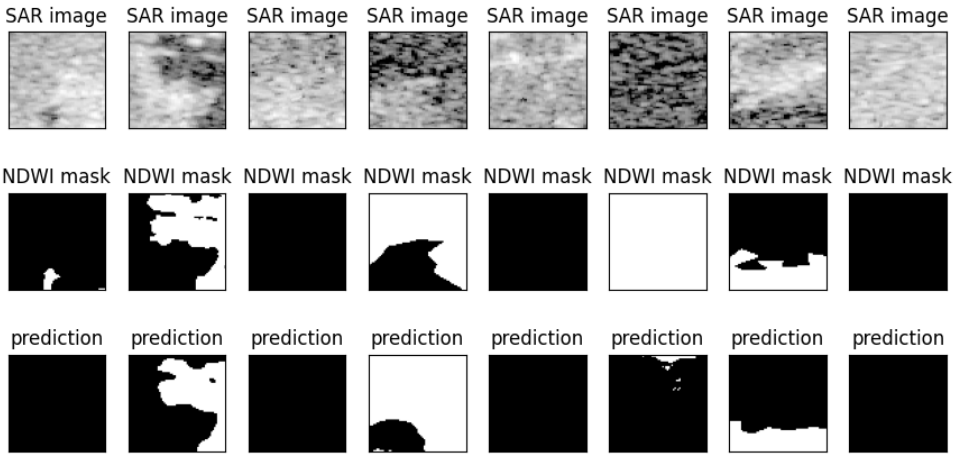

Figure 4.17: Nyúl DeepAqua model prediction examples

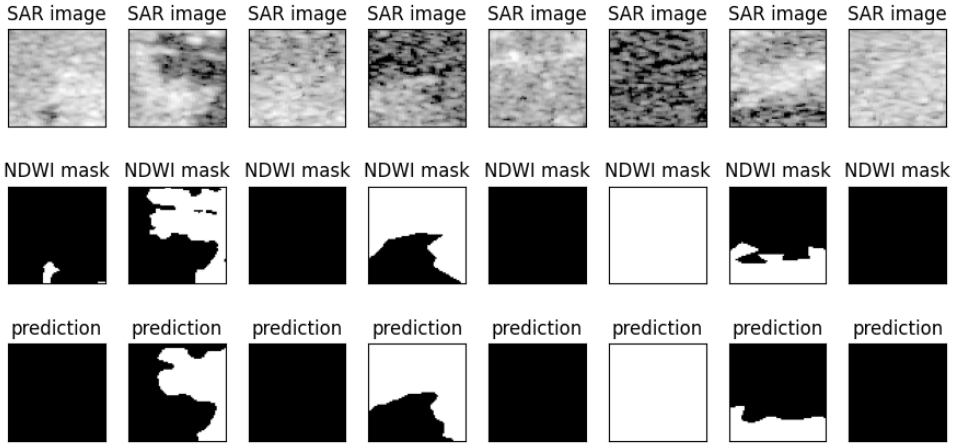

Figure 4.18: Z-score DeepAqua model prediction examples

Table 4.12: Model performances on 2018-2019 test data (mean $\pm$ standard deviation)

Models	**IoU**	**Precision**	**Recall**	**F1-score**	**Pixel Accuracy**
DeepAqua 2018	0.91 ±0.0092	0.97 ±0.0028	0.94 ±0.012	0.96 ±0.0050	0.98±0.0021
DeepAqua 2020	0.0047±0.0011	0.96±0.016	0.0047 ±0.0011	0.0094±0.0021	0.77 ±0.00025
Nyúl DeepAqua	0.83 ±0.013	0.93 ±0.013	0.88 ±0.023	0.91 ±0.0080	0.96 ±0.0030
Z-score DeepAqua	0.91 ±0.0095	0.97 ±0.0045	0.94 ±0.0134	0.96 ±0.0052	0.98 ±0.0022

Table 4.13: Model performances on 2020-2022 test data (mean ± standard deviation)

Models	IoU	Precision	Recall	F1-score	Pixel Accuracy
DeepAqua 2018	0.47±0.029	0.47 ±0.030	1.0 ±0.0013	0.64 ±0.027	0.73±0.032
DeepAqua 2020	0.92±0.0026	0.96 ±0.0039	0.95 ±0.0040	0.96 ±0.0014	0.98 ±0.00068
Nyúl DeepAqua	0.89 ±0.0094	0.94 ±0.0089	0.94±0.0067	0.94 ±0.00052	0.97 ±0.0025
Z-score DeepAqua	0.89 ±0.0097	0.91 ±0.013	0.98 ±0.0051	0.94 ±0.0054	0.97 ±0.0030

We can observe the same trend as for the validation data: the normalization techniques improve the evaluation metric scores over the base DeepAqua for the post-2020 test data. Z-score DeepAqua performs reliably well across all years, and is thus robust to changes in years and water extent, but there is a degradation in IoU from data before 2020 to data after 2020. We notice that the precision goes down while the recall goes up. The model thus tends to slightly over-predict water pixels on post-2020 data. This might be due to the fact that the intensity histograms of the data are different enough to degrade the model's capability to separate water and land pixels in post-2020 data, since the normalization method is basic.

On the other hand, Nyúl DeepAqua performs noticeably worse than it did on the validation set, and performs better on post-2020 test data than on pre-2020 test data. This can be explained by the difference between the test dataset and the training dataset. While the training datasets are taken on two similar dates, and with no great changes in water extent of the lakes and wetlands, the test dataset spans many different dates. From one month to the next, the water extent can change considerably. We see this mainly for both Hjälstaviken and Svartådalen in years 2018-2019. For instance, between October 2018 and April 2019, the water extent for Hjälstaviken is very different (see Figures 4.19 and 4.20). For the test set, the Nyúl normalization used a reference image that was close in water extent to the image of October 2018. The resulting Hjälstaviken IoU score for October 2018 is 0.83, while the IoU score for April 2019 is 0.15. We thus theorize that Nyúl normalization works less well when the water extents of the wetlands vary significantly. This hypothesis is in line with the fact that the Nyúl normalization method was originally meant for images of the same body part, with relatively small changes in the proportions types of tissue. It ensues that histogram matching works less well when intensity histogram profiles are very different. We can thus conclude that while Nyúl normalization is powerful when dealing with wetlands that show small variations in water extent, such as Hornborgasjön (overall IoU score of 0.90), it is less suited for wetlands that show large variations in water extent.

Figure 4.19: Ground truth mask for Hjälstaviken 2018-10-03

Figure 4.20: Ground truth mask for Hjälstaviken 2019-04-01

Chapter 5

Conclusions and Future work

5.1 Conclusions

To conclude, data normalization is a first, promising step to create a more robust model. The changes in SAR data from year to year are due to a difference in noise and contrast. These differences can be reduced by normalizing all the SAR images used for training and evaluation of the model. Of the two methods tested, both significantly improved model prediction for post-2020 images. A simple normalization technique such as Z-score is enough to significantly improve DeepAqua's performance, and is robust across months and water extent changes. However, the simplicity of the technique still leads to a slight degradation in model prediction on post-2020 data. A more complex Nyúl normalization is very powerful across years for datasets that showcase small water extent changes. However, when dealing with wetlands with varying water extent, the method leads to worse model prediction. When choosing which normalization method to use for water detection, it is thus important to know the characteristics of the data of interest, and inform the choice accordingly.

5.2 Limitations

Although data normalization improves the base model's performance, there is still a gap between the prediction results for test data before 2020 and after

2020. In addition, each normalization technique has its drawbacks. The Z-score normalization, while very simple to implement, does not perform as well on the post-2020 test set as DeepAqua 2020. However, the technique is easier to use than the Nyúl normalization, because it does not require a reference image or set of images, and is more robust to changes in water extent. Furthermore, further examination of the test set would be useful to determine where each normalization method performed worse.

5.3 Future work

On the long term, expanding the dataset used for training to encompass a broader array of days and regions would be beneficial for the model's accuracy. The model should also be tested on a wider year span to ensure that it can be used for studies that span several decades or more. These two expansions of the training set would perfect the model for use across the entire country of Sweden. Further expansion could be made to types of wetlands that are not present in Sweden, such as mangrove swamps.

Another robustness increase could likely be achieved by employing data augmentation during model training. Creating artificial data with varying levels of noise, contrast, and brightness, would help the model generalize better to unseen data. Additionally, expanding the model to differentiate open water from water under vegetation is necessary for more fine-grained analysis of changes in wetlands. In the same vein, exploration of other types of radar wavelengths, such as L-band, might lead to the model being able to detect water even under thick and tall vegetation.

5.4 Reflections

The thesis illustrates the benefits of interdisciplinary research. Normalization methods taken from medical imaging lead to good results in remote sensing applications. The thesis partly improves on the state-of-the-art results from DeepAqua, and thus furthers research into wetland mapping, especially in Sweden. With normalization, use of DeepAqua is easier, since only one model is required. This lowers the technology threshold for use of the model, which is meaningful for researchers that are not used to working with machine learning. For instance, hydrology researchers at SU will be able to use this model in their research, which fulfills the purpose of the thesis. The thesis is thus a meaningful contribution to research in hydrology, and to the UN SDG number

fifteen.

References

[1] T. C. on Wetlands. The importance of wetlands. [Online]. Available: https://www.ramsar.org/about/our-mission/importance-wetlands

[2] J. Thorslund, J. Jarsjo, F. Jaramillo, J. W. Jawitz, S. Manzoni, N. B. Basu, S. R. Chalov, M. J. Cohen, I. F. Creed, R. Goldenberg *et al.*, "Wetlands as large-scale nature-based solutions: Status and challenges for research, engineering and management," *Ecological Engineering*, vol. 108, pp. 489–497, 2017.

[3] H. Jafarzadeh, M. Mahdianpari, E. W. Gill, B. Brisco, and F. Mohammadimanesh, "Remote sensing and machine learning tools to support wetland monitoring: A meta-analysis of three decades of research," *Remote Sensing*, vol. 14, no. 23, 2022. doi: 10.3390/rs14236104. [Online]. Available: https://www.mdpi.com/2072-4292/14/23/6104

[4] U. Gunnarsson and M. Löfroth, "Våtmarksinventeringen – resultat från 25 års inventeringar," 2009.

[5] S. Adeli, B. Salehi, M. Mahdianpari, L. J. Quackenbush, B. Brisco, H. Tamiminia, and S. Shaw, "Wetland monitoring using sar data: A meta-analysis and comprehensive review," *Remote Sensing*, vol. 12, no. 14, 2020. doi: 10.3390/rs12142190. [Online]. Available: https://www.mdpi.com/2072-4292/12/14/2190

[6] D. Effah, A. Zia, M. Awrangjeb, Y. Gao, and K. Sarpong, "Advances in machine learning for wetland classification: a comprehensive survey of methods and applications," *Artificial Intelligence Review*, vol. 59, no. 1, p. 24, 2025.

[7] F. J. Peña, C. Hübinger, A. H. Payberah, and F. Jaramillo, "Deepaqua: Semantic segmentation of wetland water surfaces with sar imagery using

deep neural networks without manually annotated data," *International Journal of Applied Earth Observation and Geoinformation*, vol. 126, p. 103624, 2024. doi: https://doi.org/10.1016/j.jag.2023.103624. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S156984322300448X

[8] U. N. . D. of Economic and S. Affairs. Goal 15. [Online]. Available: https://sdgs.un.org/goals/goal15

[9] N. Gorelick, M. Hancher, M. Dixon, S. Ilyushchenko, D. Thau, and R. Moore, "Google earth engine: Planetary-scale geospatial analysis for everyone," *Remote Sensing of Environment*, 2017. doi: 10.1016/j.rse.2017.06.031. [Online]. Available: https://doi.org/10.1016/j.rse.2017.06.031

[10] J. Ansel, E. Yang, H. He, N. Gimelshein, A. Jain, M. Voznesensky, B. Bao, P. Bell, D. Berard, E. Burovski, G. Chauhan, A. Chourdia, W. Constable, A. Desmaison, Z. DeVito, E. Ellison, W. Feng, J. Gong, M. Gschwind, B. Hirsh, S. Huang, K. Kalambarkar, L. Kirsch, M. Lazos, M. Lezcano, Y. Liang, J. Liang, Y. Lu, C. Luk, B. Maher, Y. Pan, C. Puhrsch, M. Reso, M. Saroufim, M. Y. Siraichi, H. Suk, M. Suo, P. Tillet, E. Wang, X. Wang, W. Wen, S. Zhang, X. Zhao, K. Zhou, R. Zou, A. Mathews, G. Chanan, P. Wu, and S. Chintala, "PyTorch 2: Faster Machine Learning Through Dynamic Python Bytecode Transformation and Graph Compilation," in *29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2 (ASPLOS '24)*. ACM, Apr. 2024. doi: 10.1145/3620665.3640366. [Online]. Available: https://docs.pytorch.org/assets/pytorch2-2.pdf

[11] E. S. Agency. Introducing copernicus. [Online]. Available: https://www.esa.int/Applications/Observing_the_Earth/Copernicus/Introducing_Copernicus

[12] C. D. S. Ecosystem. Sentinel-2. [Online]. Available: https://dataspace.copernicus.eu/data-collections/copernicus-sentinel-data/sentinel-2

[13] N. Earthdata. Synthetic aperture radar (sar). [Online]. Available: https://www.earthdata.nasa.gov/learn/earth-observation-data-basics/sar

[14] SentiWiki. S1 mission. [Online]. Available: https://sentiwiki.copernicus.eu/web/s1-mission

[15] C. D. S. Ecosystem. Sentinel-1. [Online]. Available: https://dataspace.copernicus.eu/data-collections/sentinel-data/sentinel-1

[16] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, p. 436, 2015.

[17] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2017. [Online]. Available: https://arxiv.org/abs/1412.6980

[18] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015. [Online]. Available: https://arxiv.org/abs/1512.03385

[19] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 1097–1105. [Online]. Available: http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf

[20] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," 2015. [Online]. Available: https://arxiv.org/abs/1505.04597

[21] X. X. Zhu, D. Tuia, L. Mou, G.-S. Xia, L. Zhang, F. Xu, and F. Fraundorfer, "Deep learning in remote sensing: A comprehensive review and list of resources," *IEEE Geoscience and Remote Sensing Magazine*, vol. 5, no. 4, p. 8–36, Dec. 2017. doi: 10.1109/mgrs.2017.2762307. [Online]. Available: http://dx.doi.org/10.1109/MGRS.2017.2762307

[22] X. X. Zhu, S. Montazeri, M. Ali, Y. Hua, Y. Wang, L. Mou, Y. Shi, F. Xu, and R. Bamler, "Deep learning meets sar: Concepts, models, pitfalls, and perspectives," *IEEE Geoscience and Remote Sensing Magazine*, vol. 9, no. 4, pp. 143–172, 2021. doi: 10.1109/MGRS.2020.3046356

[23] Y. Wang, C. M. Albrecht, N. A. A. Braham, L. Mou, and X. X. Zhu, "Self-supervised learning in remote sensing: A review," *IEEE Geoscience and Remote Sensing Magazine*, vol. 10, no. 4, pp. 213–247, 2022. doi: 10.1109/MGRS.2022.3198244

[24] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *International conference on machine learning*. PmLR, 2020, pp. 1597–1607.

[25] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," 2015. [Online]. Available: https://arxiv.org/abs/1503.02531

[26] H. Hu, L. Xie, R. Hong, and Q. Tian, "Creating something from nothing: Unsupervised knowledge distillation for cross-modal hashing," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 06 2020. doi: 10.1109/CVPR42600.2020.00319 pp. 3120–3129.

[27] M. Canty, A. Nielsen, and M. Schmidt, "Automatic radiometric normalization of multitemporal satellite imagery," *Remote Sensing of Environment*, vol. 91, pp. 441–451, 06 2004. doi: 10.1016/j.rse.2003.10.024

[28] N. Leach, N. C. Coops, and N. Obrknezev, "Normalization method for multi-sensor high spatial and temporal resolution satellite imagery with radiometric inconsistencies," *Computers and Electronics in Agriculture*, vol. 164, p. 104893, 2019. doi: https://doi.org/10.1016/j.compag.2019.104893. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0168169919301243

[29] F. Filipponi, "Sentinel-1 grd preprocessing workflow," *Proceedings*, vol. 18, no. 1, 2019. doi: 10.3390/ECRS-3-06201. [Online]. Available: https://www.mdpi.com/2504-3900/18/1/11

[30] S. Seoni, A. Shahini, K. M. Meiburger, F. Marzola, G. Rotunno, U. R. Acharya, F. Molinari, and M. Salvi, "All you need is data preparation: A systematic review of image harmonization techniques in multi-center/device studies for medical support systems," *Computer Methods and Programs in Biomedicine*, vol. 250, p. 108200, 2024. doi: https://doi.org/10.1016/j.cmpb.2024.108200. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0169260724001962

[31] T. Islam, M. S. Hafiz, J. R. Jim, M. M. Kabir, and M. Mridha, "A systematic review of deep learning data augmentation in medical imaging: Recent advances and future research directions," *Healthcare Analytics*, vol. 5, p. 100340, 2024. doi: https://doi.org/10.1016/j.health.2024.100340. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S277244252400042X

[32] A. Carré, G. Klausner, M. Edjlali, M. Lerousseau, J. Briend-Diop, R. Sun, S. Ammari, S. Reuzé, E. Alvarez Andres, T. Estienne *et al.*, "Standardization of brain mr images across machines and protocols: bridging the gap for mri-based radiomics," *Scientific reports*, vol. 10, no. 1, p. 12340, 2020.

[33] L. G. Nyúl and J. K. Udupa, "On standardizing the mr image intensity scale," *Magnetic Resonance in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine*, vol. 42, no. 6, pp. 1072–1081, 1999.

[34] L. Nyul, J. Udupa, and X. Zhang, "New variants of a method of mri scale standardization," *IEEE Transactions on Medical Imaging*, vol. 19, no. 2, pp. 143–150, 2000. doi: 10.1109/42.836373

[35] J. C. Reinhold, B. E. Dewey, A. Carass, and J. L. Prince, "Evaluating the impact of intensity normalization on MR image synthesis," in *Medical Imaging 2019: Image Processing*, vol. 10949. International Society for Optics and Photonics, 2019, p. 109493H.

[36] M. Shah, Y. Xiao, N. Subbanna, S. Francis, D. L. Arnold, D. L. Collins, and T. Arbel, "Evaluating intensity normalization on mris of human brain with multiple sclerosis," *Medical Image Analysis*, vol. 15, no. 2, pp. 267–282, 2011. doi: https://doi.org/10.1016/j.media.2010.12.003. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1361841510001337

[37] M. Mahdianpari, B. Salehi, M. Rezaee, F. Mohammadimanesh, and Y. Zhang, "Very deep convolutional neural networks for complex land cover mapping using multispectral remote sensing imagery," *Remote Sensing*, vol. 10, no. 7, 2018. doi: 10.3390/rs10071119. [Online]. Available: https://www.mdpi.com/2072-4292/10/7/1119

[38] K. B. Dang, M. H. Nguyen, D. A. Nguyen, T. T. H. Phan, T. L. Giang, H. H. Pham, T. N. Nguyen, T. T. V. Tran, and D. T. Bui, "Coastal wetland classification with deep u-net convolutional networks and sentinel-2 imagery: A case study at the tien yen estuary of vietnam," *Remote Sensing*, vol. 12, no. 19, p. 3270, 2020.

[39] A. Jamali, M. Mahdianpari, B. Brisco, D. Mao, B. Salehi, and F. Mohammadimanesh, "3dunetgsformer: A deep learning pipeline for

complex wetland mapping using generative adversarial networks and swin transformer," *Ecological informatics*, vol. 72, p. 101904, 2022.

[40] A. Jamali and M. Mahdianpari, "Swin transformer and deep convolutional neural networks for coastal wetland classification using sentinel-1, sentinel-2, and lidar data," *Remote Sensing*, vol. 14, no. 2, p. 359, 2022.

[41] L. Scheibenreif, J. Hanna, M. Mommert, and D. Borth, "Self-supervised vision transformers for land-cover segmentation and classification," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 1422–1431.

[42] B. Hosseiny, M. Mahdianpari, B. Brisco, F. Mohammadimanesh, and B. Salehi, "Wetnet: A spatial–temporal ensemble deep learning model for wetland classification using sentinel-1 and sentinel-2," *IEEE transactions on geoscience and remote sensing*, vol. 60, pp. 1–14, 2021.

[43] H. Jafarzadeh, M. Mahdianpari, and E. W. Gill, "Wet-gc: A novel multimodel graph convolutional approach for wetland classification using sentinel-1 and 2 imagery with limited training samples," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 15, pp. 5303–5316, 2022. doi: 10.1109/JSTARS.2022.3177579

[44] M. Mahdianpari, B. Salehi, F. Mohammadimanesh, S. Homayouni, and E. Gill, "The first wetland inventory map of newfoundland at a spatial resolution of 10 m using sentinel-1 and sentinel-2 data on the google earth engine cloud computing platform," *Remote Sensing*, vol. 11, no. 1, 2019. doi: 10.3390/rs11010043. [Online]. Available: https://www.mdpi.com/2072-4292/11/1/43

[45] Y. Gao, W. Li, M. Zhang, J. Wang, W. Sun, R. Tao, and Q. Du, "Hyperspectral and multispectral classification for coastal wetland using depthwise feature interaction network," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–15, 2021.

[46] J. Ai, X. Han, L. Chen, H. He, X. Li, Y. Tan, T. Xie, and X. Tang, "Deep neural network and transfer learning for annual wetland vegetation mapping using sentinel-2 time-series data in the heterogeneous lake floodplain environment," *International Journal of Remote Sensing*, pp. 1–24, 2025.

[47] S. K. McFEETERS, "The use of the normalized difference water index (ndwi) in the delineation of open water features," *International Journal of Remote Sensing*, vol. 17, no. 7, pp. 1425–1432, 1996. doi: 10.1080/01431169608948714. [Online]. Available: https://doi.org/10.1080/01431169608948714

[48] L. R. Dice, "Measures of the amount of ecologic association between species," *Ecology*, vol. 26, no. 3, pp. 297–302, 1945.

[49] NAISS. National academic infrastructure for supercomputing in sweden. [Online]. Available: https://www.naiss.se/

[50] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization." *Journal of machine learning research*, vol. 13, no. 2, 2012.

[51] L. Biewald, "Experiment tracking with weights and biases," 2020, software available from wandb.com. [Online]. Available: https://www.wandb.com/

TRITA – XXX-EX 2026:0000
Stockholm, Sweden 2026

www.kth.se

€€€€ For DIVA €€€€

{
"Author1": { "Last name": "Karens",
"First name": "Lucia",
"Local User Id": "u1XXXXXX",
"E-mail": "lkarens@kth.se",
"organisation": {"L1": "",
}
},
"Cycle": "2",
"Course code": "DA233X",
"Credits": "30.0",
"Degree1": {"Educational program": "Master's Programme, Machine Learning, 120 credits"
,"programcode": "TMAIM"
,"Degree": "Degree of Master of Science in Engineering"
,"subjectArea": "Degree of Master of Science in Engineering"
},
"Title": {
"Main title": "Detecting Wetlands Using Self-Supervised Convolutional Neural Networks With Data Normalization",
"Language": "eng" },
"Alternative title": {
"Main title": "Detektering av Våtmarker med Hjälp av Självövervakade Faltningsnätverk och Datanormalisering",
"Language": "swe"
},
"Supervisor1": { "Last name": "Peña",
"First name": "Francisco J.",
"Local User Id": "u1XXXXXX",
"E-mail": "frape@kth.se",
"organisation": {"L1": "School of Engineering Sciences in Chemistry, Biotechnology and Health",
"L2": "School of Chemistry, Biotechnology and Health" }
},
"Examiner1": { "Last name": "Kragic Jensfelt",
"First name": "Danica",
"Local User Id": "u1ydsyln",
"E-mail": "dani@kth.se",
"organisation": {"L1": "School of Electrical Engineering and Computer Science",
"L2": "School of Electrical Engineering and Computer Science" }
},
"National Subject Categories": "10201, 10207",
"SDGs": "15",
"Other information": {"Year": "2026", "Number of pages": "xvii,55"},
"Copyrightleft": "copyright",
"Series": { "Title of series": "TRITA – XXX-EX" , "No. in series": "2026:0000" },
"Opponents": { "Name": "Johan Hedstrand Welander"},
"Presentation": { "Date": "2026-07-02 11:00"
,"Language":"eng"
,"Room": "via Zoom https://kth-se.zoom.us/j/65529857622"
,"City": "Stockholm" },
"Number of lang instances": "3",
"Abstract[eng]": €€€€

Wetlands are an important natural resource that needs to be preserved and monitored. Monitoring them using remote sensing and machine learning is fast, accurate, and cheap, especially while using self-supervised deep learning. The current state-of-the-art in monitoring wetlands in Sweden is the DeepAqua convolutional neural network model from the article ``DeepAqua: Semantic segmentation of wetland water surfaces with SAR imagery using deep neural networks without manually annotated data". The model uses satellite imagery, radar imagery, and a U-Net based model to detect water on the radar images. However, DeepAqua performs poorly on data from years it has not been trained on due to changes in the data characteristics. In this paper we explore two data normalization methods to improve DeepAqua's robustness to new data. These methods are popular in medical imaging: Nyúl normalization, also known as histogram matching, and Z-score normalization. Each normalization method is implemented and tested with the base DeepAqua model. Both methods improve the model's prediction on years it has not been trained on. Z-score normalization is best suited for wetlands where water extent changes considerably. Nyúl normalization is slightly more powerful to generalise to unseen years, but is mostly suited for wetlands where water extent changes slightly. Methods used in medical imaging applications can thus be transferred to remote sensing applications, and data normalization is a strong option to improve DeepAqua's robustness.

"Abstract[swe]": €€€€

Våtmarker är en viktig naturresurs som behöver värnas om och övervakas. Övervakning av våtmarker med fjärranalys och maskininlärning är snabbt, noggrant, och billigt, särskilt med självövervakad djupinlärning. Den nuvarande bästa modellen för att övervaka våtmarker i Sverige är DeepAqua-modellen från ``DeepAqua: Semantic segmentation of wetland water surfaces with SAR imagery using deep neural networks without manually annotated data". Modellen använder satellitbilder, radarbilder, och en U-net baserad modell för att identifiera vatten på radarbilderna. DeepAqua presterar dock sämre på år den inte har tränats på, på grund av ändringar i datans egenskaper. I den här rapporten utforskar vi

två metoder för datanormalisering. Dessa metoder används inom medicinsk avbildning: Nyúl-normalisering och Z-score-normalisering. Varje normaliseringsmetod implementeras och testas med DeepAqua modellen som bas. Båda metoder förbättrar modellens förutsägelse på osedd data från senare år. Z-score-normalisering passar bäst för våtmarker där vattenutbredning förändras avsevärt. Nyúl-normalisering är något kraftfullare för att generalisera till osedd data, men är mest lämpad för våtmarker där vattenutbredning inte förändras mycket. Metoder som används i medicinska avbildningstillämpningar kan således överföras till fjärranalystillämpningar, och datanormalisering är ett starkt alternativ för att förbättra DeepAquas robusthet.

"Abstract[fre]": €€€€

Les zones humides constituent une ressource naturelle importante qui doit être préservée et surveillée. Leur surveillance par télédetection et apprentissage automatique est rapide, précise et économique, notamment avec l'usage de l'apprentissage profond auto-supervisé. En Suède, l'état de l'art en surveillance de zones humides est représenté par le modèle DeepAqua, présenté dans l'article ``DeepAqua: Semantic segmentation of wetland water surfaces with SAR imagery using deep neural networks without manually annotated data". Ce modèle utilise des données satellites et radar et un modèle basé sur U-Net pour détecter l'eau sur les images radar. Le modèle est cependant peu performant sur des données issues de d'années non utilisées pour son entraînement. Ceci est dû à un changement des charactéristiques des données. Dans ce rapport, nous explorons deux méthodes de normalization de données afin d'améliorer la robustesse de DeepAqua face à de nouvelles données. Ces méthodes sont utilisées couramment en imagerie médicale: la normalisation de Nyúl et la normalisation z-score (aussi appelée normalisation standard). Chaque méthode est implémentée et testée avec le modèle DeepAqua. Les deux méthodes mènent à une amélioration de la prédiction du modèle pour les données inédites d'années plus récentes. La normalisation z-score est particulièrement adaptée aux zones humides où la superficie en eau varie considérablement. La normalisation de Nyúl, légèrement plus performante pour la généralisation aux années non observées, convient surtout aux zones humides où la superficie en eau varie peu. Les méthodes utilisées en imagerie médicale peuvent ainsi être transposées à la télédétection, et la normalisation des données constitue est ainsi une option pertinente pour améliorer la robustesse de DeepAqua.

€€€€,
"Keywords[eng]": €€€€
Machine Learning, Convolutional Neural Network, Self-Supervised Learning, Semantic Segmentation, Normalization, Remote Sensing, Wetlands €€€€,
€€€€,
"Keywords[swe]": €€€€
Maskininlärning, Faltningsnätverk, Självledd Inlärning, Semantisk Segmentering, Normalisering, Fjärranalys, Våtmarker €€€€,
€€€€,
"Keywords[fre]": €€€€
Apprentissage Automatique, Réseau de Neurones Convolutif, Apprentissage Auto-Supervisé, Segmentation Sémantique, Normalization, Télédétection, Zones Humides €€€€,
}

acronyms.tex

```
%%% Local Variables:
%%% mode: latex
%%% TeX-master: t
%%% End:
% The following command is used with glossaries-extra
\setabbreviationstyle[acronym]{long-short}
% The form of the entries in this file is \newacronym{label}{acronym}{phrase}
%                                      or \newacronym[options]{label}{acronym}{phrase}
% see "User Manual for glossaries.sty" for the  details about the options, one example is shown below
% note the specification of the long form plural in the line below

\newacronym{DiVA}{DiVA}{Digitala Vetenskapliga Arkivet}

\newacronym{LADOK}{LADOK}{Lokalt -adbbaserat dokumentationssystemt}

\newacronym{KTH}{KTH}{KTH Royal Institute of Technology}

% note the use of a non-breaking dash in the following acronym



\newacronym{UN}{UN}{United Nations}
\newacronym{SDG}{SDG}{Sustainable Development Goal}

\newacronym{ML}{ML}{Machine Learning}
\newacronym{DL}{DL}{Deep Learning}
\newacronym{CNN}{CNN}{Convolutional Neural Network}
\newacronym{SAR}{SAR}{Synthetic Aperture Radar}
\newacronym{RS}{RS}{Remote Sensing}
\newacronym{ESA}{ESA}{European Space Agency}
\newacronym{GEE}{GEE}{Google Earth Engine}
\newacronym{MSI}{MSI}{Multispectral Imaging}
\newacronym{MR}{MR}{Magnetic Resonance}
\newacronym{SU}{SU}{Stockholm University}
\newacronym{IoU}{IoU}{Intersection over Union}
\newacronym{NDWI}{NDWI}{Normalized Difference Water Index}
```