KTH ROYAL INSTITUTE
OF TECHNOLOGY

Degree Project in Machine Learning

Second cycle, 30 credits

Using Satellite Images and Deep Learning to Detect Water Hidden Under the Vegetation

A cross-modal knowledge distillation-based method to reduce manual annotation work

EZIO CRISTOFOLI

Stockholm, Sweden, 2024

Using Satellite Images and Deep Learning to Detect Water Hidden Under the Vegetation

A cross-modal knowledge distillation-based method to reduce manual annotation work

EZIO CRISTOFOLI

Degree Programme in Computer Science and Engineering
Date: January 9, 2024

Supervisor: Francisco Peña Natgeo
Examiner: Amir H. Payberah
School of Electrical Engineering and Computer Science
Swedish title: Användning Satellitbilder och Djupinlärning för att Upptäcka Vatten Gömt Under Vegetationen
Swedish subtitle: En tvärmodal kunskapsdestillationsbaserad metod för att minska manuellt anteckningsarbete

Abstract

Detecting water under vegetation is critical to tracking the status of geological ecosystems like wetlands. Researchers use different methods to estimate water presence, avoiding costly on-site measurements.

Optical satellite imagery allows the automatic delineation of water using the concept of the Normalised Difference Water Index (NDWI). Still, optical imagery is subject to visibility conditions and cannot detect water under the vegetation, a typical situation for wetlands. Synthetic Aperture Radar (SAR) imagery works under all visibility conditions. It can detect water under vegetation but requires deep network algorithms to segment water presence, and manual annotation work is required to train the deep models.

This project uses DEEPAQUA, a cross-modal knowledge distillation method, to eliminate the manual annotation needed to extract water presence from SAR imagery with deep neural networks. In this method, a deep student model (e.g., UNET) is trained to segment water in SAR imagery. The student model uses the NDWI algorithm as the non-parametric, cross-modal teacher. The key prerequisite is that NDWI works on the optical imagery taken from the exact location and simultaneously as the SAR. Three different deep architectures are tested in this project: UNET, SegNet, and UNET++, and the Otsu method is used as the baseline.

Experiments on imagery from Swedish wetlands in 2020-2022 show that cross-modal distillation consistently achieved better segmentation performances across architectures than the baseline. Additionally, the UNET family of algorithms performed better than SegNet with a confidence of 95%. The UNET++ model achieved the highest Intersection Over Union (IOU) performance. However, no statistical evidence emerged that UNET++ performs better than UNET, with a confidence of 95%.

In conclusion, this project shows that cross-modal knowledge distillation works well across architectures and removes tedious and expensive manual work hours when detecting water from SAR imagery. Further research could evaluate performances on other datasets and student architectures.

Keywords

Computer Vision, Deep Networks, Semantic Image Segmentation, Knowledge Distillation, UNET, UNET++, SegNet, Satellites, Synthetic Aperture Radar (SAR), Sentinel-1, Sentinel-2, Google Earth Engine, Automatic Annotation, Normalised Difference Water Index (NDWI).

Sammanfattning

Att upptäcka vatten under vegetation är avgörande för att hålla koll på statusen på geologiska ekosystem som våtmarker. Forskare använder olika metoder för att uppskatta vattennärvaro vilket undviker kostsamma mätningar på plats.

Optiska satellitbilder tillåter automatisk avgränsning av vatten med hjälp av konceptet Normalised Difference Water Index (NDWI). Optiska bilder fortfarande beroende av siktförhållanden och kan inte upptäcka vatten under vegetationen, en typisk situation för våtmarker. Synthetic Aperture Radar (SAR)-bilder fungerar under alla siktförhållanden. Den kan detektera vatten under vegetation men kräver djupa nätverksalgoritmer för att segmentera vattennärvaro, och manuellt anteckningsarbete krävs för att träna de djupa modellerna.

Detta projekt använder DEEPAQUA, en cross-modal kunskapsdestillationsmetod, för att eliminera det manuella annoteringsarbete som behövs för att extrahera vattennärvaro från SAR-bilder med djupa neurala nätverk. I denna metod tränas en djup studentmodell (t.ex. UNET) att segmentera vatten i SAR-bilder semantiskt. Elevmodellen använder NDWI, som fungerar på de optiska bilderna tagna från den exakta platsen och samtidigt som SAR, som den icke-parametriska, cross-modal lärarmodellen. Tre olika djupa arkitekturer testas i detta examensarbete: UNET, SegNet och UNET++, och Otsu-metoden används som baslinje.

Experiment på bilder tagna på svenska våtmarker 2020-2022 visar att cross-modal destillation konsekvent uppnådde bättre segmenteringsprestanda över olika arkitekturer jämfört med baslinjen. Dessutom presterade UNET-familjen av algoritmer bättre än SegNet med en konfidens på 95%. UNET++-modellen uppnådde högsta prestanda för Intersection Over Union (IOU). Det framkom dock inga statistiska bevis för att UNET++ presterar bättre än UNET, med en konfidens på 95%.

Sammanfattningsvis visar detta projekt att cross-modal kunskapsdestillation fungerar bra över olika arkitekturer och tar bort tidskrävande och kostsamma manuella arbetstimmar vid detektering av vatten från SAR-bilder. Ytterligare forskning skulle kunna utvärdera prestanda på andra datamängder och studentarkitekturer.

Nyckelord

Computer Vision, Deep Networks, Semantic Image Segmentation, Knowledge Distillation, UNET, UNET++, SegNet, Satellites, Synthetic Aperture Radar

(SAR), Sentinel-1, Sentinel-2, Google Earth Engine, Automatic Annotation, Normalised Difference Water Index (NDWI).

Acknowledgments

I would like to thank Francisco Peña Natgeo, for the precious supervision and support provided to me during the development of this study. Thanks also to Johanna Hansen and Ioannis Iakovidis for their ideas and joint effort in annotating the data.

Stockholm, January 2024
Ezio Cristofoli

Contents

List of Figures

List of Tables

List of acronyms and abbreviations

AC	Atrous Convolution
CNN	Convolutional Neural Network
DC	Dilated Convolution
FCN	Fully Convolutional Network
FN	False Negative
FP	False Positive
GAN	Generative Adversarial Network
GPU	Graphics Processing Unit
IOU	Intersection Over Union
MRI	Magnetic Resonance Imaging
NAISS	National Academic Infrastructure for Supercomputing in Sweden
NDWI	Normalised Difference Water Index
PA	Pixel Accuracy
RNN	Recurrent Neural Networks
SAR	Synthetic Aperture Radar
TN	True Negative
TP	True Positive

Chapter 1

Introduction

In this thesis, I examine the use of deep neural network algorithms to extract information about water presence in satellite imagery. The study's main objective is to compare performances of selected CNN-based architectures when utilizing DEEPAQUA, a cross-modal knowledge distillation method (described in [1]). The method uses satellite optical imagery to replace the manual annotation of satellite Synthetic Aperture Radar (SAR) imagery.

1.1 Background

Tracking reliably and cost-effectively climate change impacts on the natural world is a significant activity nowadays. The localization and measurement of different water ecosystems in the natural environment required in the past costly on-site presence. Today, on-site activities are replaced by semantic segmentation deep algorithms applied to satellite imagery [1].

In this context, the Department of Geology of Stockholm University utilizes Deep Learning algorithms to identify, locate, size, and visualize the extension of wetlands in Sweden over the years.

Wetlands are a specific ecosystem of geological interest where "water covers the soil or is present either at or near the surface of the earth all year or for varying periods during the year, including during the growing season" [2]. One peculiar characteristic of wetlands is that often, the water might be hidden under vegetation.

The primary types of satellite imagery used to analyze wetlands include optical and SAR systems:

- Optical satellites are passive systems since they detect the sunlight terrestrial bodies reflect. Their imagery (e.g., Sentinel-2) is very similar to aerial photos and supports well water detection but is strongly affected by visibility conditions (like clouds, fog, and night) and has difficulties in detecting water under vegetation;

- SAR systems are active since they transmit signals and detect their reflection on the Earth's surface. Since the transmitted signal can pass through clouds or vegetation, SAR imagery overcomes the limitations of day/night, clouds, and covering vegetation and is the preferred satellite source for wetlands analysis.

Deep neural networks have proven to be accurate in the semantic segmentation of images. Semantic segmentation consists of associating a label or category with every pixel in an image, and it is used to recognize a collection of pixels that form distinct categories. In particular, Convolutional Neural Network (CNN) based architectures like Fully Convolutional Network (FCN) [3], Dilated Convolution (DC) [4], Atrous Convolution (AC) [5], and U-Net [6] have been successfully used in segmentation tasks in different areas such as medical, geoscience and autonomous vehicles to mention a few.

As shown in Figure 1.1, this approach requires time-consuming annotation work to manually identify water instances in the SAR imagery and produce the pixel label information needed to train the deep network.

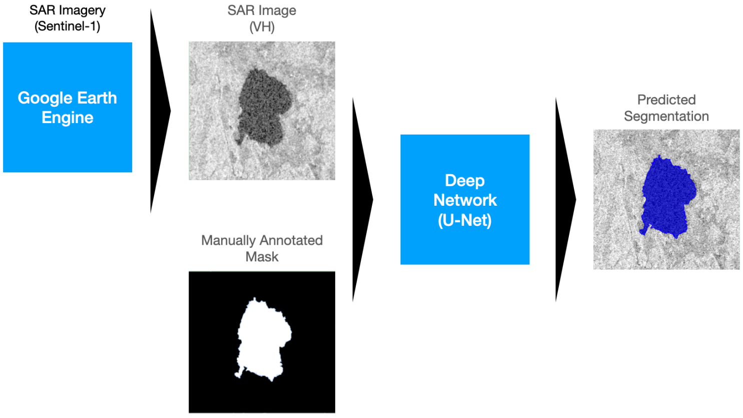

Figure 1.1: Current approach for SAR imagery segmentation requires manual annotation of imagery for training the algorithm

In this study, we use DEEPAQUA, a method to replace the manual annotation work of the SAR imagery by utilizing the optical remote sensing indicator Normalised Difference Water Index (NDWI) [7] to generate masks as shown in Figure 1.2 automatically. While DEEPAQUA utilizes a UNET architecture, in this study, we assess the prediction accuracy performances across a selection of deep architectures.

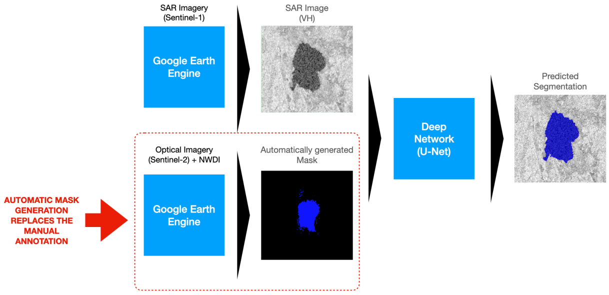
SAR Imagery (Sentinel-1)
Google Earth Engine
SAR Image (VH)
Deep Network (U-Net)
Predicted Segmentation
AUTOMATIC MASK GENERATION REPLACES THE MANUAL ANNOTATION
Optical Imagery (Sentinel-2) + NWDI
Google Earth Engine
Automatically generated Mask

Figure 1.2: Automatic mask generation NDWI-based approach replaces the manual annotation work

1.2 Problem

The key research question addressed by this project is: ”What performances can be obtained by a selection of CNN-based architectures when segmenting water in SAR imagery utilizing DEEPAQUA, the automatic annotation process based on the optical concept of NDWI described in [1]?”

The first step of this project consists of replicating the DEEPAQUA method with a new from-scratch implementation of UNET (the same architecture used in DEEPAQUA). The segmentation accuracy is expected to match the performances obtained in [1]. In the second step of this project, we apply DEEPAQUA to other CNN-based architectures. We test the segmentation accuracy performances and expect to align with the state-of-the-art architectures for water detection.

1.3 Purpose

The purpose of this project is to replicate the results of the methodology described in [1] and verify performances with additional deep architectures. The output of this project is relevant for geoscience researchers engaged in water detection. This project utilized publicly available data and established scientific methods to contribute to preserving the natural ecosystem's health.

1.4 Goals

This project aims to replicate DEEPAQUA performances with a new from-scratch UNET implementation and to verify performances over a selection of CNN architectures. This has been divided into the following three sub-goals:

1. Verify the viability of replacing the manual annotation of SAR imagery with DEEPAQUA;
2. Replicate the accuracy that a UNET architecture can achieve;
3. Identify which architecture performs best over a selection of CNN architectures.

1.5 Research Methodology

The project uses the publicly available satellite optical (Sentinel-2) and SAR (Sentinel-1) imagery of Swedish wetlands in the Google Earth Engine, and the cross-modal knowledge distillation method described in [1].

A quantitative research method is adopted throughout the work. The deductive approach is adopted to prove the research hypothesis.

The viability of replacing the manual annotation of SAR imagery with the automatic NDWI-based approach is tested by training the architectures with the cross-modal knowledge distillation method described in paragraph 2.3 and comparing the obtained results with publicly available sources ([1]).

The performances of the different architectures are assessed on a manually annotated SAR satellite imagery test set. Comparisons are performed considering 95% confidence intervals.

1.6 Delimitations

This project focuses on verifying the feasibility and assessing the performances of an automated annotation approach based on cross-modal knowledge distillation.

3D segmentation architectures are outside the scope of this project. Other alternative methods to replace manual annotation with automatic processing, e.g., self-supervised learning or autoencoders, are also outside this project's scope.

The Swedish wetlands in scope include Örebro (for training) and Svartådalen (for testing), and the considered satellite data refer to the period 2020-2022.

1.7 Structure of the thesis

Chapter 2 presents the background information about the semantic segmentation deep architectures in scope, the NDWI concept, and the adopted knowledge distillation approach to replace the manual annotations of SAR imagery. Chapter 3 illustrates the methods I used to perform the experiments. Chapter 4 presents and discusses the experiments' results. Finally, Chapter 5 summarises the findings, highlights limitations, and sketches possible future work.

Chapter 2

Background

This chapter provides background information about water detection using satellite optical sensors (Section 2.1) and water detection using satellite radar sensors (Section 2.2). Additionally, this chapter describes the three architectures used in this study for semantic segmentation and the concepts of self-supervised learning through knowledge distillation (Section 2.3). Finally, the chapter provides an overview of the relevant related work (Section 2.4) and concludes with a concise summary (Section 2.5).

2.1 Water Detection with Optical Sensors

Satellite optical imagery (e.g., Sentinel-2) utilizes sensors to capture the reflected sunlight from the earth's surface. NDWI [7] is a popular method to identify water presence in optical imagery. The founding principles of NDWI are:

- Water reflects green light;
- Water reflects poorly Near Infrared (NIR) frequencies;
- Elements with no water (soil and vegetation) reflect very well NIR frequencies.

Equation 2.1 shows how the NDWI value is calculated for each pixel of an optical satellite image, where G is the energy reflected in the Green Band, and NIR is the energy reflected in the Near Infrared Band. Considering the properties of water and soil described above, the values of NDWI are defined in the range from -1 to 1, with negative values corresponding to “no water” and positive values to ”water”.

$$NDWI = \frac{G - NIR}{G + NIR} \tag{2.1}$$

Figure 2.1 shows how NDWI segments water in optical imagery. Equation 2.1 can automatically generate the NDWI image, which is then transformed into the binary NDWI water mask shown in the figure.

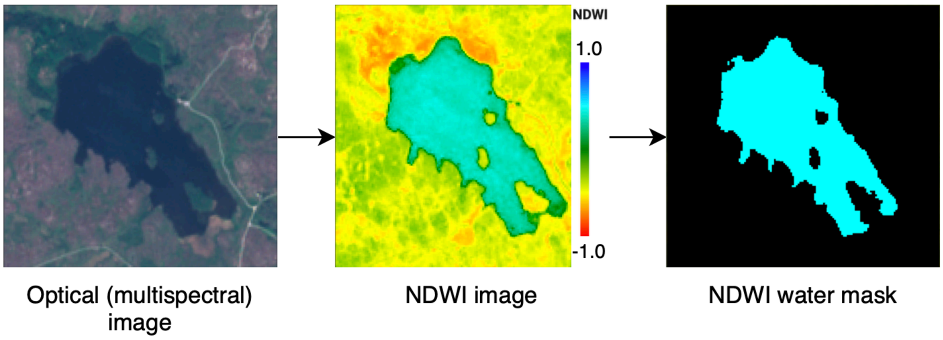

Figure 2.1: Water segmentation with NDWI [1]

The NDWI-based water segmentation approach can be executed in a completely automatic way. The primary limitations include:

- The optical satellite imagery is only usable in clear daylight conditions with no clouds and no night darkness;
- The optical satellite imagery does not detect water when it is hidden below the vegetation (in this case, the reflectivity capabilities of the vegetation win over the water).

Both of the above are substantial disadvantages when aiming to identify vegetation-rich wetlands.

2.2 Water Detection in Radar Imagery

Synthetic Aperture Radar Systems (SAR) (e.g., Sentinel-1) transmit power at frequencies below the light spectrum and capture the reflected energy from the earth's surface. The frequency band, different from optical, allows this satellite imagery to work independently of local light and weather conditions. Also, it permits identifying water surfaces covered by vegetation [1]. On the other hand, this type of imagery is more sensitive than optical to noise and speckles

[1]. In the case of SAR imagery, there is, though not a straightforward concept like NDWI, which can be used to segment water presence. The approach researchers adopt consists of using semantic segmentation deep algorithms to extract information about water presence. Semantic segmentation of water in SAR imagery has to satisfy the following simple main requirements:

- Can perform 2D binary segmentation;
- Can reach good accuracy performances (possibly even with limited training data).

Different Convolutional Neural Networks (CNN) based architectures meet these requirements [8]. This study focused on the following deep models, all based on an encoder-decoder architecture principle [9] :

- SegNet [9] is one of the first encoder-decoder networks, often used in autonomous vehicle and medical imagery use cases. Described in the following Section 2.2.1.
- UNET [6], originally developed for medical imagery use cases but commonly used by geo-researchers for water segmentation tasks. Described in the following Section 2.2.2.
- UNET++ [10] enhanced version of UNET designed to increase accuracy in medical imagery use cases. Described in the following Section 2.2.3.

2.2.1 SegNet

SegNet [9] is a convolutional neural network [8] architecture designed for semantic segmentation in computer vision. It is often used for self-driving vehicles and analysis of medical imagery like Magnetic Resonance Imaging (MRI) use cases. It is designed to take an image as input and produce a pixel-wise label map as output.

Figure 2.2 shows the architecture of SegNet based on an encoder-decoder architecture (respectively highlighted in the picture with blue-green and red-light blue colors).

The encoding part captures high-level features by applying 13 convolutional and pooling layers. The encoder network performs convolutions with a filter bank to produce feature maps. These are then batch normalized. Then, an element-wise rectified linear non-linearity (ReLU) is applied. Then,

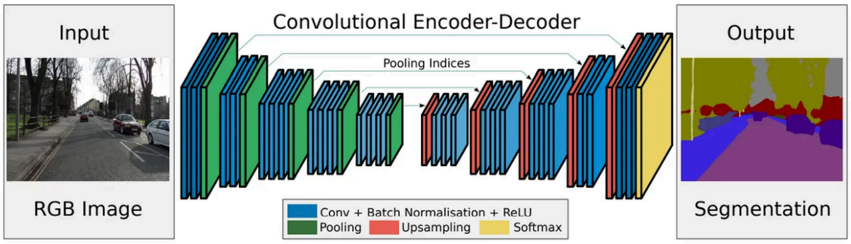

Figure 2.2: SegNet architecture [9]

max-pooling with a 2×2 window and stride 2 (non-overlapping window) is performed, and the resulting output is sub-sampled by a factor of 2.

A specific aspect of SegNet consists in the sub-sampling stage, where Max-pooling is used to achieve translation invariance over small spatial shifts in the image. Sub-sampling leads to each pixel governing a larger input image context, contributing to achieving high classification accuracy at the price of reducing the feature map size.

The output image resolution should be the same as the input image. To achieve this, up-sampling is performed on the decoder side. The SegNet decoder network up-samples its input feature map using the memorized max-pooling indices from the corresponding encoder feature map(s), as shown in figure 2.3.

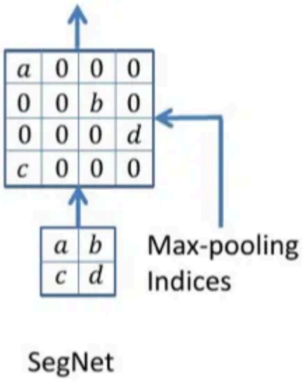

Figure 2.3: Max-pooling mechanism in SegNet decoders [9]

Key characteristics of the SegNet decoder include:

- For each of the 13 encoders, there is a corresponding decoder that up-samples the feature map using memorized max-pooling indices;

- Sparse feature maps of higher resolutions are produced;
- Sparse maps are fed through a trainable filter bank to produce dense feature maps;
- The last decoder is connected to a softmax classifier, which classifies each pixel.

The main advantages offered by SegNet in comparison with other encoder-decoder architectures for semantic segmentation include the following:

- The reuse of pooling indices for decoding offers a relatively more computationally efficient solution versus the solutions adopted in the other two architectures in the scope of this project;
- The model can be trained with a limited amount of labeled data;
- The model has proven in multiple use cases to produce high-resolution segmentation maps [9].

2.2.2 UNET

Figure 2.4 shows UNET's architecture which consists of a sequence of CNN modules of descending (coding path) and ascending (decoding path) feature dimensionality.

In the coding path, the width and heights of the feature maps are shrunk while the channel expands by a factor of 2 until it reaches 1024 (typically the maximum recommended level for CNNs). The feature maps' widths and heights are expanded to the mask's dimension in the decoding path.

The coding path aims to capture context, while the decoding path enables precise image feature localization. The connections (”skip connections”) between the coding and the decoding modules in each hierarchical level of the paths provide the detail to reconstruct accurate shapes of the segmentation boundaries.

Substantially, while SegNet brings to the decoder only the max-pooling indexes from the encoding side, in UNET, the skip connection brings to the decoding side the complete feature map from the encoder in the same hierarchical position.

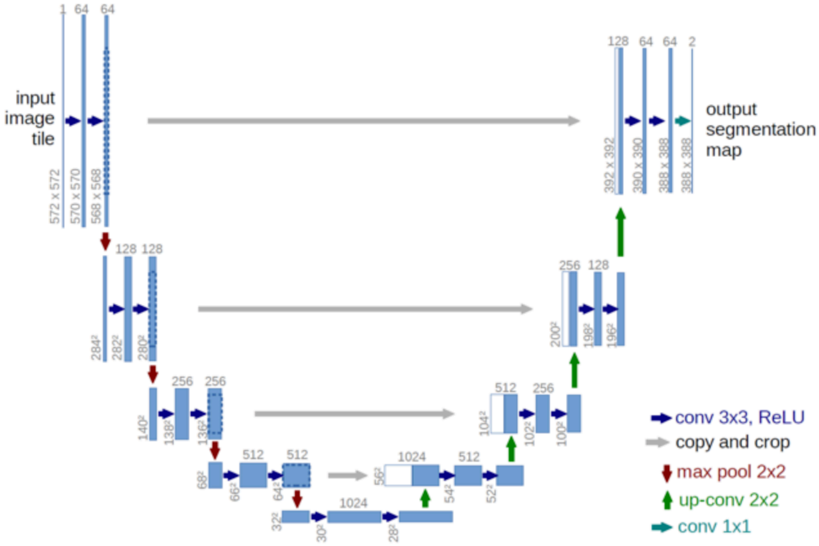

Figure 2.4: Reference UNET architecture [6]

UNET was initially developed in the context of medical diagnostic tasks. Still, given its simplicity and accuracy, it has also been successfully used in other use cases, including satellite imagery segmentation.

2.2.3 UNET++

UNET++ is an evolution of UNET developed with a focus on medical imagery [10]. Figure 2.5 provides a schematic overview of the UNET++ architecture where the nodes in black color are the modules of an original UNET backbone.

A substantial change in UNET++ is the replacement of the UNET's skip connections between the coders and the decoders with more dense convolutional blocks. Those blocks use information from the module in the same and below levels. Figure 2.6 shows a schematic view of how dense convolutional blocks work:

- In the formula shown at the top of Figure 2.6, H is the DenseNet's composite function that combines Batch Normalization, ReLU activation, and a 3x3 convolution;
- The elements inside [] are concatenated as the H composite function inputs;

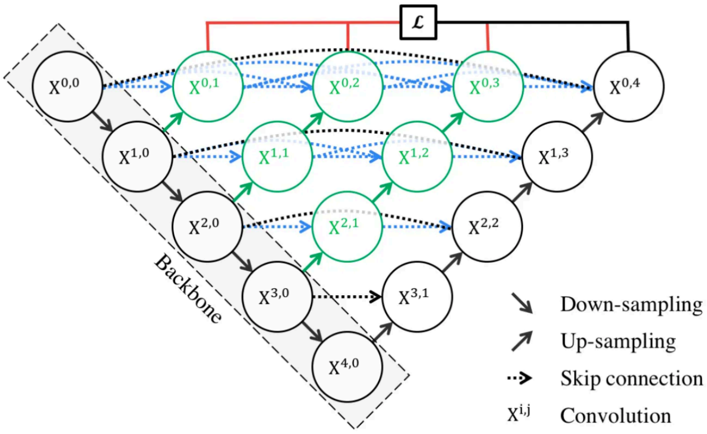

Figure 2.5: Reference UNET++ architecture [10]

- U is the UNETs composite function, which consists of two 3x3 convolutions with ReLU activations.

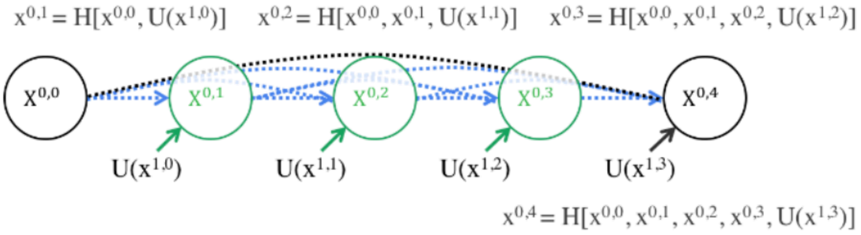
x0,1 = H[x0,0, U(x1,0)]
x0,2 = H[x0,0, x0,1, U(x1,1)]
x0,3 = H[x0,0, x0,1, x0,2, U(x1,2)]
X0,0
X0,1
X0,2
X0,3
X0,4
U(x1,0)
U(x1,1)
U(x1,2)
U(x1,3)
x0,4 = H[x0,0, x0,1, x0,2, x0,3, U(x1,3)]

Figure 2.6: UNET++ dense block [10]

Replacing the skip connections with more complex convolutional blocks aims to reduce the semantic gap between the encoder's and decoder's feature maps so the model has an easier learning task.

An additional change compared to UNET is introduced by a mechanism called by the authors "deep supervision." It substantially allows the UNET++ to operate in two different modes:

- Accurate Mode: in this modality, the final output of the model is obtained by averaging the outputs from all branches in level 0;

- Fast mode: in this modality, there is the possibility to increase the speed of the model, reducing the number of blocks in the coder and the decoder used to generate the output. Figure 2.7 indicates how the deep supervision mechanism in Fast Mode works for different values of the supervising parameter L.

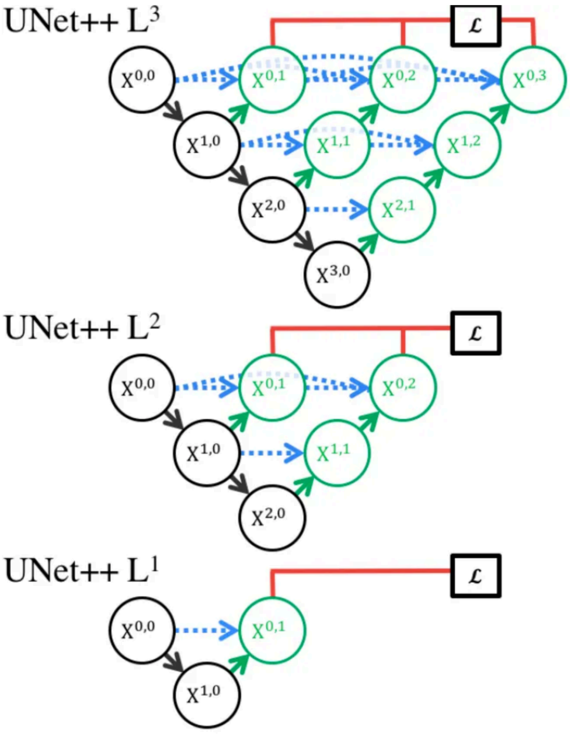

Figure 2.7: UNET++ deep supervision functionality [10]

Finally, it has to be noted that, differently from the other two architectures, in their paper [10], the authors propose a slightly more complex loss function consisting of the combination of the Dice Loss and the Binary Cross Entropy.

2.3 Self-supervised Learning through Knowledge Distillation

Self-supervised learning is a machine learning method [11] that allows algorithms to learn without needing human-annotated samples. This paragraph discusses a method to achieve self-supervised learning for SAR imagery semantic segmentation using knowledge distillation [12].

Knowledge distillation [12] is a methodology, frequently adopted in neural networks, of using a “teacher” model, often, but not always necessarily, complex, to teach to a “student” model, usually simpler, to obtain “teacher” level performances from the “student” despite of their different complexity.

In [1], Francisco Peña and others propose DEEPAQUA. It is a method to apply a “reversed” distillation process to perform water segmentation. In this scenario, the teacher is an optical-based (NDWI-based, see Section 2.1) model, and the student is a radar-based model (e.g., U-Net on SAR imagery). This process of distilling knowledge from different domains (optical and SAR in this case) is also known as cross-modal knowledge distillation [13].

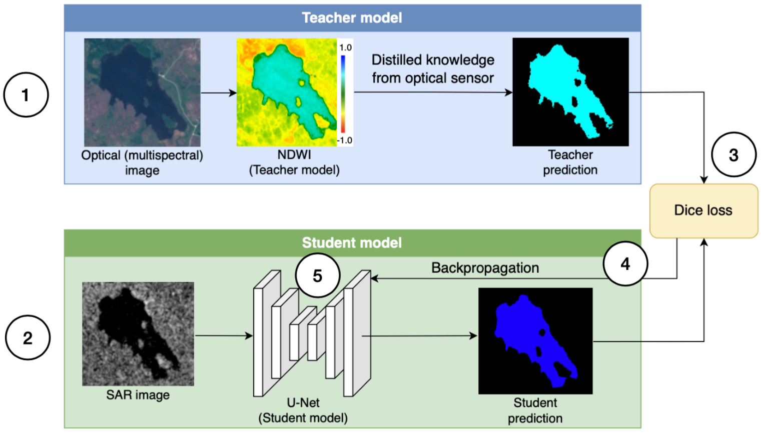

Figure 2.8: Cross-modal knowledge distillation steps from the optical NDWI-based teacher model to the UNET student model working on SAR [1]

The advantage of this approach is that it allows the transferring of the simplicity of the teacher (no manual annotation work required by the optical model) to train the student automatically. As shown in Figure 2.8, in such a distillation process, the teacher algorithm generates predictions, which are then used to train the student according to the following steps [1]:

Step 1 Given a pair of optical and SAR images referring to the same area and taken at the same time, feed the optical image to the teacher model and obtain the (NDWI-based) segmentation mask as its output;

Step 2 Feed the SAR image to the student model (e.g., UNET) and get the predicted segmentation mask by the student;

Step 3 Optimise the student model computing the Dice Loss (see Section 2.3.1) between the obtained teacher and the student segmentation masks;

Step 4 Compute the gradient of the Loss to the weights of the student model;

Step 5 Update the student weights using a regularisation optimizer (e.g., Adam);

Step 6 Repeat steps 1-5 for all pairs of optical and SAR images in the training set until convergence.

This project replicated the above method with a UNET (Section 2.2.2) student model in the test region of Svartådalen and experimented with the two additional CNN-based student architectures: SegNet (Section 2.2.1) and UNET++ (Section 2.2.3).

2.3.1 Dice Loss

The Dice Loss [14] is commonly used for semantic segmentation tasks. It is defined as:

$$L_{Dice} = 1 - \frac{2 \times |Y_T * Y_S| + \epsilon}{|Y_T| + |Y_S| + \epsilon} \tag{2.2}$$

where:

- Y_T is the teacher output;
- Y_S is the student output;
- $|\cdot|$ denotes the sum of all elements in a matrix;
- $*$ denotes element-wise multiplication;
- ϵ is a small constant to avoid division by zero.

The Dice loss ranges from 0 to 1, where lower values show higher similarity. By minimizing the Dice loss, the student model learns to mimic the teacher model's output and thus segment SAR images without requiring annotations [1].

2.4 Related Work

This section provides an overview of related work in the research areas of 2D Semantic Segmentation (Section 2.4.1) and wetland detection (Section 2.4.2).

2.4.1 2D Semantic Segmentation

Deep architectures like CNNs [8] have, since the introduction of AlexNet [15], proven to deliver higher accuracy performances than traditional machine learning models like random forests or support vector machines in computer vision tasks. Compared to the traditional models, CNN networks self-learn the relevant representative features within the images and better generalize.

Semantic image segmentation is an area of computer vision that aims to classify each image pixel into a predefined set of classes. Various use cases have driven interest in semantic image segmentation, including autonomous vehicles, medical imaging, and remote sensing. The Fully Convolutional Network (FCN) [3] has been one of the first models proposed for 2D image segmentation. It uses convolutional layers to reduce the dimensionality of the input images (encoder). It then makes a class prediction at a reduced level of granularity. Finally, it uses upsampling and deconvolution layers to resize the image to the original dimensions (decoder). However, because the encoder module reduces the input's resolution, the decoder module struggles to produce fine-grained segmentation.

The SegNet architecture [9], described in Section 2.2.1 also adopts an encoder-decoder architecture and addresses the FCN's drawback using in the decoder the memorized max-pooling indices from the corresponding encoder feature map(s), while the UNET architecture [6], as described in Section 2.2.2, addresses the same FCN's drawback adding skip connections from earlier layers in the encoder and summing their feature map to the feature maps in the decoder.

Further improvements to the UNET architectures are UNET++ [10], and UNET3+ [16]. As described in Section 2.2.3, UNET++ is a modification of the UNET architecture that uses nested dense skip connections to reduce the semantic gap between the feature maps of the encoder and decoder sub-networks, simplifying the optimizer's job. UNET3+ further improves the dense skip connection architecture to simplify the model without penalizing the performance.

One benefit of downsampling a feature map is broadening the receptive field for the following filter (given a constant filter size). However, the broader context comes at the cost of reduced spatial resolution. Dilated convolutions provide an alternative approach to gaining a wide field of view while preserving the full spatial dimension [5]. Some architectures replace the last few pooling layers with dilated convolutions with successively higher dilation rates at higher computational costs, for example, in the DeepLab family of models [17].

2.4.2 Wetland Detection

Wetland detection from satellite imagery is not a trivial task, as water bodies reflect radiations in different ways depending on the weather and light conditions, the water haziness, and vegetation.

Many researchers use satellite optical imagery, e.g., Sentinel-2, to identify wetlands using deep learning algorithms. Various authors proposed the utilization of different architectures such as AlexNet [15], ResNet [18], and DenseNet [19]. However, these approaches are sensitive to weather and daylight conditions and, even more importantly, can not detect water hidden under vegetation, a common situation for wetlands.

To overcome this limitation, researchers have proposed to use satellite radar imagery, e.g., Sentinel-1, which operates in the C-band and can pass through vegetation and clouds [20]. Different algorithms have been used with satellite SAR imagery, including random forests [21], the WetNet model [22] an ensemble of 2D CNN, 3D CNN and Recurrent Neural Networks (RNN) models [23], 3D UNET [24] models utilizing a Generative Adversarial Network (GAN) [25] to generate synthetic data with similar characteristics to the ground-truth. However, all these approaches share the limitation of requiring manually annotated data to train the model, which is a time-consuming and costly activity.

Self-supervised semantic segmentation from SAR images using knowledge distillation from optical imagery is an approach presented in 2023 in the paper “Deepaqua: Semantic segmentation of wetland water surfaces with sar imagery using deep neural networks without manually annotated data” [1]. It aims to reduce the required manual annotation work for training the model. The approach, described in Section 2.3, utilizes a student UNET architecture. This project replicates the methodology described in [1] and extends results to the SegNet and UNET++ student architectures.

2.5 Summary

The detection of wetlands from satellite imagery is a complex task. The automatic extraction of water presence from multispectral optical satellite imagery is possible with methods like NDWI, but satellite optical imagery is sensitive to clouds, weather, and day/night conditions; moreover, water hidden under vegetation, very common in wetlands, cannot be seen.

Adopting deep learning algorithms to perform water segmentation from SAR imagery is also a common method researchers use. It works independently of weather and light conditions and detects water hidden under vegetation. Various deep architectures are available, but all the current methods require extensive manual annotation work to train the models.

Self-supervised learning through knowledge distillation provides a method to extract, without manual annotations, water presence in SAR imagery utilizing a deep network as the student model and the NDWI algorithm working on optical imagery as the teacher model.

This project utilizes self-supervised learning through knowledge distillation with three CNN-based student architectures, UNET, SegNet, and UNET++, and tests their performances on the Svartådalen data set.

Chapter 3

Methods

This chapter describes the methods used in this thesis project. Section 3.1 provides an overview of the adopted research process. Section 3.2 discusses the research paradigm. Section 3.3 describes data collection and preparation. Section 3.4 describes the metrics used to assess the segmentation quality, the baseline chosen for performance evaluations, and the methods adopted to evaluate and compare the performance of the selected algorithms.

3.1 Research Process

I addressed the research problem with the following steps:

Step 1 Creation of the Swedish wetlands dataset used for the project: training and validation data are reused from [1] while the test set is a newly created manually annotated set created with the methods described in 3.3;

Step 2 Implementation from scratch of UNET (Section 3.4);

Step 3 Assessment of the performances of the self-supervised learning through knowledge distillation method of a UNET student model working on the wetlands dataset (assessment performed with the methods described in Section 3.4);

Step 4 Repetition of Step 2 and Step 3 for SegNet;

Step 5 Repetition of Step 2 and Step 3 for UNET++;

Step 6 Performance comparison of UNET, SegNet, and UNET++ and conclusions (methods described in Section 3.4).

3.2 Research Paradigm

I developed the project, adopting a selection of quantitative methods. I replicated and extended the experiments developed in [1] to address the research question.

The replica of [1] consists in the analysis of the performances of the UNET student architecture (Section 2.2.2), while the additions consist in the analysis of the performances of the two student architectures SegNet (Section 2.2.1) and UNET++ (Section 2.2.3).

I then applied statistical analysis (Section 3.4.4) to compare the performances of the three student architectures to a baseline (Section 3.4.3) and between each other to validate the answers for the research questions exposed in Section 1.4.

3.3 Data Collection

This section describes the methods adopted for creating the Swedish wetlands dataset utilized for testing purposes in this project. Section 3.3.1 describes how the SAR imagery has been obtained. Section 3.3.2 describes how the SAR imagery was annotated, and Section 3.3.3 describes how the SAR images and their annotations have been tiled to create the wetlands dataset. The methodology replicates the process described in [1].

3.3.1 SAR Imagery

The list of Swedish wetlands sites and their GPS coordinates are available at the site ramsar.org. Based on the amount and quality (substantially affected by the quantity of snow in the images during winter time) of the available imagery, the site of Svartadålen has been chosen to create the test data set.

I fetched Sentinel-1 imagery from the Google Earth Engine Platform using JavaScript. Years in scope included the range from 2020 to 2022. Sentinel-1 data before 2020 is not considered in this project due to speckle and noise, possibly due to an adjustment on the Sentinel-1 sensors, as noted in [1].

Each month, one image has been selected (the first available date). January, February, March, and December are excluded to avoid images containing high snow. Following the same approach described in [1], each image's data in the VH band has been considered.

All Sentinel-1 imagery in the Google Earth Engine Platform is pre-processed using the following steps: thermal noise removal, radiometric calibration, and terrain correction [26]. Therefore, no filter or pre-processing techniques have been applied to clean the original SAR images, except for removing outlier pixel values by discarding values lower than percentile one and higher than percentile 99. Min-max scaling was also applied to bring the pixel values to the range [0, 1].

3.3.2 Manual Annotation

Each Sentinel-1 image, visualized in grey-scale, has been manually annotated using the geometry import function in Google Earth Engine and then converted with JavaScript into a black and white mask, as shown in Figure 3.1.

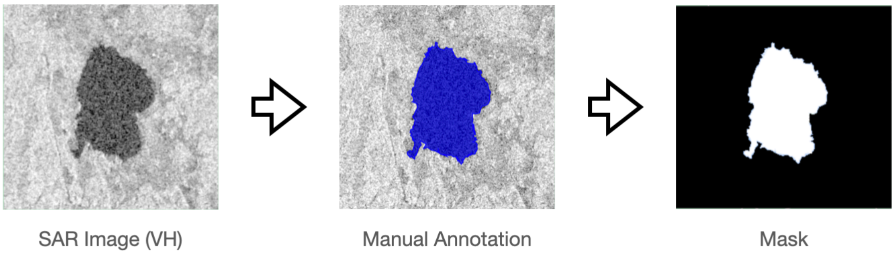

Figure 3.1: Steps of the manual annotation process

Darker areas in the SAR image correspond to water, and lighter areas correspond to the soil. The area in blue in the annotated image shows the water body whose contour was manually created by selecting the boundary profile point by point. Review sessions with three other researchers were held to mitigate the risk of subjective judgment in the manual annotation.

3.3.3 Image Tiling

The test dataset is created, splitting the collected SAR Sentinel-1 imagery and the corresponding manual annotations into 64 × 64 pixels tiles. Each pixel had a resolution of 10 meters. Python scripts originally developed for [1] were re-used for the tiling process.

Figure 3.2 illustrates how the test dataset was created by combining the tiles obtained from the SAR imagery and the corresponding manual annotations.

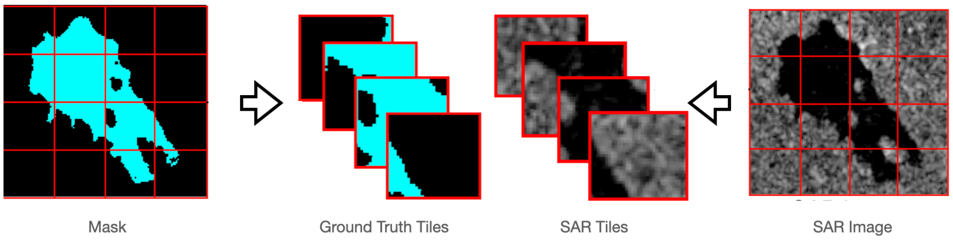

Figure 3.2: The test dataset combines the tiles from the SAR imagery and the corresponding manually annotated ground truth mask [1]

3.4 Evaluation Methods

This section describes the metrics that I have chosen to quantify the quality of the image segmentation tasks (Section 3.4.1), the method adopted to train and test the student models (Section 3.4.2), the selected baseline for performance assessment, and the methods chosen to perform comparisons versus the baseline (Section 3.4.3).

3.4.1 Metrics

Results are reported using the Intersection Over Union (IOU) metric [27], a standard for semantic segmentation tasks. IOU measures how well a predicted object aligns with the object annotation. IOU is determined by calculating the overlap among two bounding boxes, a predicted box, and a ground truth box. Mathematically, IOU is defined as:

$$IOU = \frac{TP}{TP + FP + FN} \tag{3.1}$$

Where True Positive (TP) are the pixels that are correctly labeled as water, False Positive (FP) are the pixels that are incorrectly labeled as water, False Negative (FN) are the pixels that are incorrectly labeled ground, and True Negative (TN) are the pixels that are correctly labeled as ground.

The IOU metric ranges from 0 to 1. A higher IOU value indicates a better alignment between the predicted and actual regions, reflecting a more accurate model. Additionally, IOU is robust to class imbalance, which is typical in our wetland segmentation tasks. For this last reason, I decided not to use other common metrics for semantic segmentation tasks like the Pixel Accuracy (PA), which can be misleading when there is a class imbalance [1].

3.4.2 Student Models Training and Testing

When applying the knowledge distillation method described in Section 2.3, the student model is trained using the Sentinel-1 (SAR) and the Sentinel-2 (multi-spectral optical) images of the county of Örebro (8550 km^2) in Sweden (this is the same imagery used in [1]). The steps below are adopted:

- The NDWI method is used on the Sentinel-2 images to generate water masks as described in Step 1 in Section 2.3;
- The Sentinel-1 imagery and the corresponding NDWI mask for the entire Örebro region are split into tiles of 64 × 64 pixels (as recommended in [1] and described in 3.3.3).

The final result consists of 45500 NDWI-SAR pairs obtained for the Örebro region, which are utilized for training the student model.

For the correct application of the cross-modal distillation process, it is important that the SAR and optical imagery are taken simultaneously and that the weather and visibility conditions are good. This limits the amount of suitable dates. I reused the Orebrö training imagery utilized in [1] taken on 2020-06-23.

The manually annotated Svartadålen set (Section 3.3.2) is then used to test the student models' image segmentation performance.

3.4.3 Baseline

Otsu's method [28] is used as the baseline for performance comparisons of the student models. This method is chosen since it is unsupervised and does not require manually annotated data. Additionally, this is the same baseline adopted in [1].

In summary, Otsu's method considers every possible threshold value for the pixels representing the soil and the water in the radar image, calculates the variance within each of the two clusters, and selects the value for which the weighted sum of these variances is the least [1]. This project reuses OpenCV's Python implementation of Otsu's method utilized in [1].

3.4.4 Performance Comparisons

The comparison of the student model performance to the baseline (the Otsu model) is done considering the confidence interval according to the following steps:

- Given the test set, the predictions for both the student model and the baseline model are observed, and then for each test instance, the random variable difference between the IOU performance of the model and the baseline is calculated;

- The confidence interval for the above difference random variable at 95% is inferred. The calculation assumes that the sample mean is normally distributed and utilizes the *t* distribution;

- If the above confidence interval includes the value 0, we can not conclude that one of the models performs better at the given confidence level.

When comparing the performances of the three student models between each other, since the more models we compare, the higher the risk of making a *type I* (False Positive) error, the Bonferroni correction [29] is applied i.e., instead of using a confidence level of $1 - \alpha$ for n confidence intervals, a confidence level of $1 - \alpha/n$ is used. In our case, α = 5% and n = number of pairwise comparisons = 3 (UNET vs. SegNet, UNET++ vs. SegNet, and UNET vs. UNET++).

Chapter 4

Results and Discussion

This chapter presents and discusses the experiments and their results. Section 4.1 describes the hardware and software environments used for the project. Section 4.2 provides an overview of the code utilized in the project. Section 4.3 describes the adopted approach to tune the hyperparameters of the student models. Section 4.4 presents and discusses the obtained results.

4.1 Hardware and Software Environment

The machine learning experiments of this project were developed using the Graphics Processing Unit (GPU) environment provided by the Alvis cluster. Alvis is a national resource provided by the National Academic Infrastructure for Supercomputing in Sweden (NAISS) dedicated to Artificial Intelligence and Machine Learning research. Alvis provides several types of compute nodes with multiple NVIDIA GPUs. My project used Nvidia GPUs of type T4 and A40.

The machine learning code is developed in a Python v. 3.8.10 environment and uses the following libraries:

- albumentations 1.3.0
- eeconvert 0.1.22
- geemap 0.20.5
- geopandas 0.12.2
- geojson 3.0.1

- matplotlib 3.7.1
- python-dotenv 1.0.0
- pillow 9.5.0
- rasterio 1.3.6
- rtree 1.0.1
- scikit-image 0.20.0
- scikit-learn 1.2.2
- scipy 1.9.1
- tqdm 4.65.0
- torch 2.0.0
- torchmetrics 0.11.4
- torchvision 0.15.1
- wandb 0.14.2

4.2 Project Code

To develop this project, I reused and adapted part of the code developed by F. Peña for the paper [1]. The three student model architectures are instead developed from scratch. The code is available in GitHub at `https://github.com/melqkiades/deep-wetlands-2023/tree/Ezio`.

Below is a concise description of the main functionalities implemented by the script files:

- *container.def*: definition of the Apptainer container file (used to define the software environment described in Section 4.1);
- *.env*: list of environment-specific variables that I used;
- *generate_sar.py*: script to generate tiles (Section 3.3.3) from the SAR imagery (both for training/validation and for test);

- *generate_ndwi.py*: script to generate tiles (Section 3.3.3) from the NDWI masks (for training/validation data set). It is also used to create tiles from the manually annotated masks (test set);
- *unet.py*: script to implement the UNET student model (Section 2.2.2);
- *SegNet.py*: script to implement the SegNet student model (Section 2.2.1);
- *archs.py*: script to implement the UNET++ student model (Section 2.2.3);
- *train_models.py*: script to train the student models using the cross-modal knowledge distillation method (Section 2.3);
- *baseline.py*: script to implement the Otsu method (Section 3.4.3);
- *test_IOU_vs_OTSU.py*: script to calculate the confidence intervals for each model and the Otsu's method on the test set (Section 3.4.4);
- *bonferroni.py*: script to execute the Bonferroni method on the pairwise performance comparisons of the best performers of each student model in the test set (Section 3.4.4);
- *map_wetlands.py*: script to visualize the segmentation results of each of the student models;
- *map_otsu.py*: script to visualize the segmentation results of the Otsu's method;
- *utils.py, geo_ utils, viz_utils*: utility scripts for file format conversions, completely re-used from [1];
- *jaccard_similarity.py*: script for IOU Calculation, completely re-used from [1].

4.3 Models Optimization

I trained the student model architectures with the tiled SAR imagery and the NDWI masks from the Örebro region collected on 2020-06-23 (same dataset utilized in [1]), and I have split the resulting 66,625 tile couples into training and validation with a ratio of 80-20.

As described in Section 2.3, the training used the Dice Loss and the AdamW [30] optimizer.

I optimized model hyper-parameters using grid search rounds based on the performance of the validation set. Otsu's method, being a non-parametric model, did not require any grid search.

The parameters considered for the grid search rounds and their ranges of values are listed below:

- Learning Rate (LR): [0.000005, 0,000001, 0.00005, 0.00001, 0.0005, 0.0001, 0.005, 0.001];
- Batch Size (BS): [4, 8, 16, 32, 64, 128, 256];
- Weight Decay (WD): [0.0001, 0.001, 0.01, 0.1, 0.3, 0.5].

I also doubled the size of the training data set by trying the following data augmentation (DA) techniques:

- Additive Gaussian Noise (GN) with zero mean and standard deviation of 1;
- Horizontal Flips (HF);
- Vertical Flips (VF);
- Random crop and resizing to the 64x64 size of the tiles (RC);
- Random rotations (RR) of the tiles in the ranges of:
  - Option a: [0°- 90°] (RRa),
  - Option b: [0.5°- 1.5°] (RRb),
  - Option c: [1.5°- 4.5°] (RRc),
  - Option d: [2°- 4°] (RRd),
  - Option e: [2.5°- 5.5°] (RRe),
  - Option f: [3°- 5°] (RRf),
  - Option g: [4°- 6°] (RRg),
  - Option h: [10°- 40°] (RRh).

Table 4.1 summarises the hyper-parameter selections that gave the best IOU performances on the Svartadålen set for the three architectures in scope.

Table 4.1: Best Performing Hyper-parameter Selection (additive Gaussian Noise is included in all augmentations)

Architecture	LR	WD	BS	Augmentation
SegNet	0.000 05	0.000 1	256	RC
UNET	0.000 05	0.000 1	256	RRd
UNET++	0.000 1	0.001	256	RRf

4.4 Summary Results

To assess the test performances of the student models, I used the manually annotated Svartadålen set, consisting of 24 SAR images, shown in Figure 4.1, which resulted in 2,352 annotated tiles of size 64 x 64 pixels.

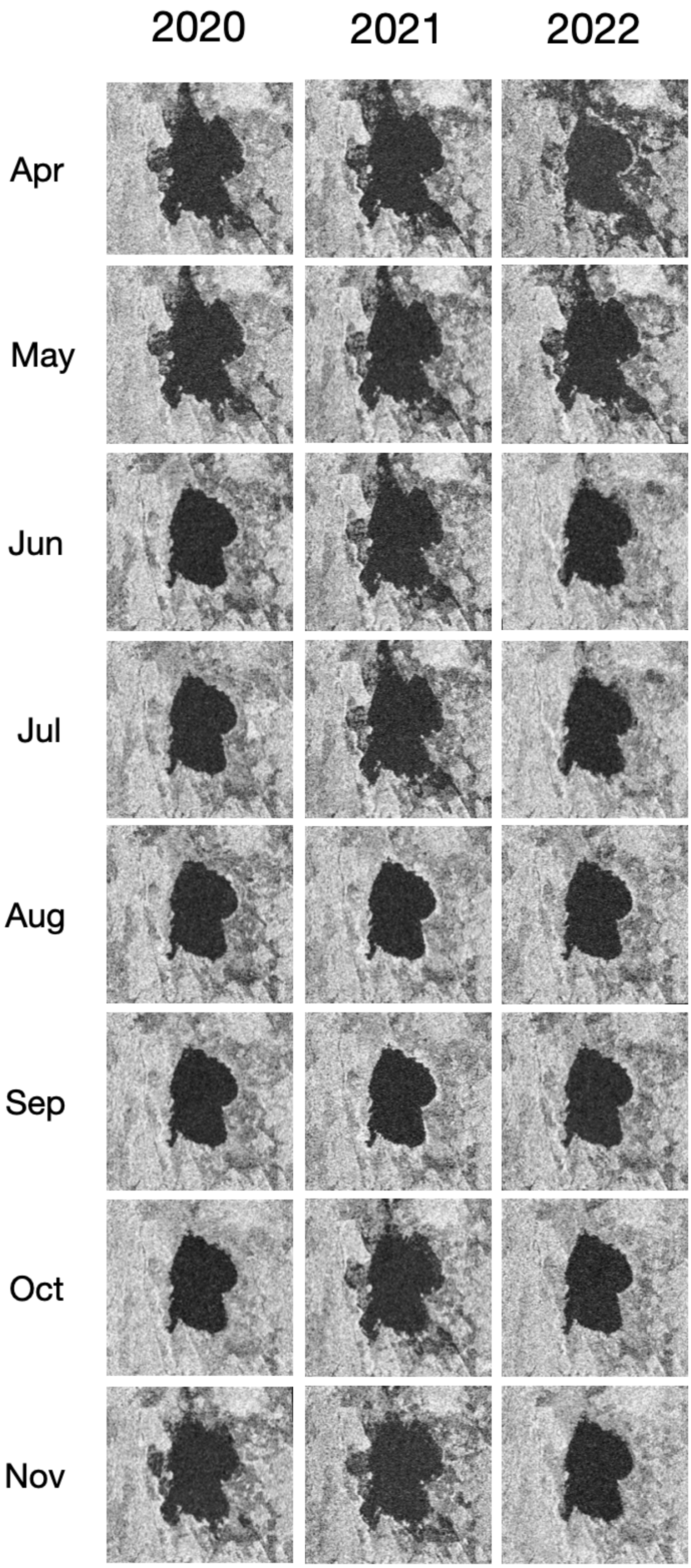

Figure 4.1: SAR Imagery constituting the Svartådalen test set

Table 4.2: Summary IOU test results

Rounds	Otsu	SegNet	UNET	UNET++
Grid search	0.613 ± 0.020	0.856 ± 0.012	$\mathbf{0.878 \pm 0.011}$	0.877 ± 0.011
Augmentation	0.613 ± 0.020	0.863 ± 0.012	0.883 ± 0.011	$\mathbf{0.884 \pm 0.011}$

Table 4.2 summarises the IOU test performance obtained for the Otsu method, the SegNet, the UNET, and the UNET++ student models after the grid search rounds and the (final) addition of data augmentation. IOU performances are described with a 95% confidence interval. Bold characters highlight the highest observed values in each optimization step. To be noted that:

- No change in IOU performances is observed for the Otsu model in the different optimization steps since the method is not parametric;
- Data Augmentation improved performances of all the parametric student models;
- The UNET IOU performances obtained in this project are in line with the ones reported in [1];
- The best final performance obtained in this project is with the UNET++ student model.

Figure 4.2 shows a comparative qualitative view of the model predictions obtained by the three best architectures on two representative dates, 2021-09-10 (A) and 2021-11-09 (B), respectively, illustrating conditions with no snow and with snow.

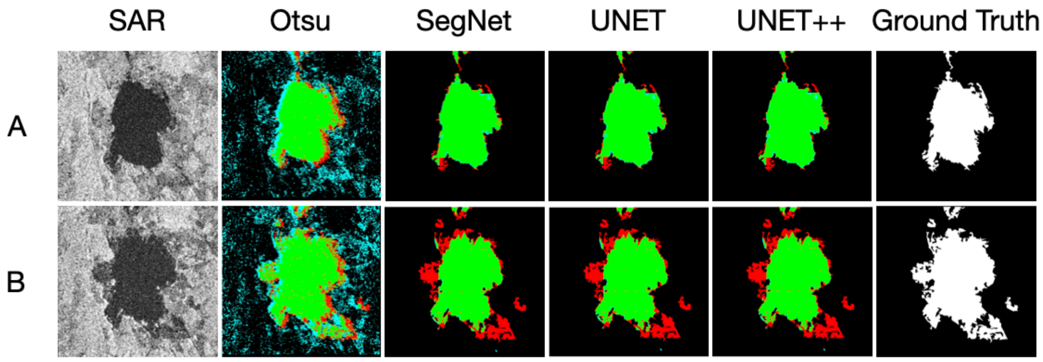

Figure 4.2: Illustration of the performance of the best student models for two different times of the year, without (A) and with (B) snow. The original SAR images are on the left, followed by the Otsu, SegNet, UNET, and UNET++ predictions, and on the far right, the manually annotated mask

In the above Figure, the following color coding is adopted (Positive = ’water’, Negative = ’no water’):

- Green for True Positive;
- Cyan for False Positive;
- Red for False Negative;
- Black for True Negative.

Sections 4.4.1, 4.4.2, and 4.4.3 present more detailed results for the three student models.

4.4.1 SegNet results

Figure 4.3 shows the plots of training loss, validation loss, training IOU, and validation IOU obtained by the best SegNet model. The hyperparameters of the best model are summarised in Table 4.1. The training and validation set consisted of the Örebro data set of 2020-06-23. The applied data augmentation doubled the original data size and included additive Gaussian Noise and Random Crop and Resizing to 64 x 64.

Each graph shows the mean value of the best model over a sample of ten experiments performed with ten different randomly selected seeds in the range of 1 to 100. The lighter blue area in each plot shows the standard error. The graphs clearly show that the model converges already after five epochs.

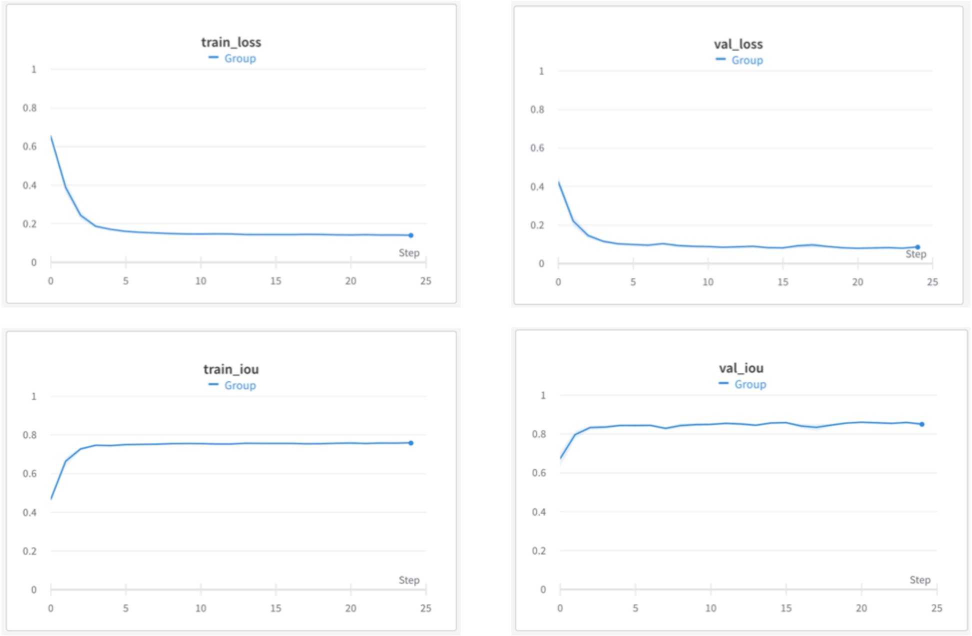

Figure 4.3: Average training and validation loss and IOU obtained with the SegNet model. The light blue area visualizes the standard error

Table 4.3 shows the 95% confidence intervals of the following variables obtained on the Svartådalen test set:

- IOU obtained with the Otsu method;
- IOU obtained with the SegNet student model;
- Difference between the IOU obtained by SegNet and the IOU of Otsu (positive values show that SegNet performed better).

The Table shows each variable's lower bound of confidence interval (LB), the mean value, and the upper bound (UB).

Table 4.3: IOU results SegNet vs. Otsu

IOU Otsu			IOU SegNet			IOU SegNet - IOU Otsu		
LB	Mean	UB	LB	Mean	UB	LB	Mean	UB
0.593	0.613	0.632	0.851	0.863	0.874	**0.232**	0.250	0.268

Positive values of the lower bound of the “difference variable” are highlighted in bold characters and confirm that with 95% confidence, the SegNet model performs better than Otsu.

4.4.2 UNET results

Figure 4.4 shows the plots of training loss, validation loss, training IOU, and validation IOU obtained by the best UNET model. The hyperparameters of the best model are summarised in Table 4.1. The training and validation set consisted of the Örebro data set of 2020-06-23. The applied data augmentation doubled the original data size and included random tile rotation in the range [2°- 4°].

Each graph shows the mean value of the best model over a sample of ten experiments performed with ten different randomly selected seeds in the range of 1 to 100. The lighter blue area in each plot shows the standard error. The graphs show that the best UNET model converges already after five epochs.

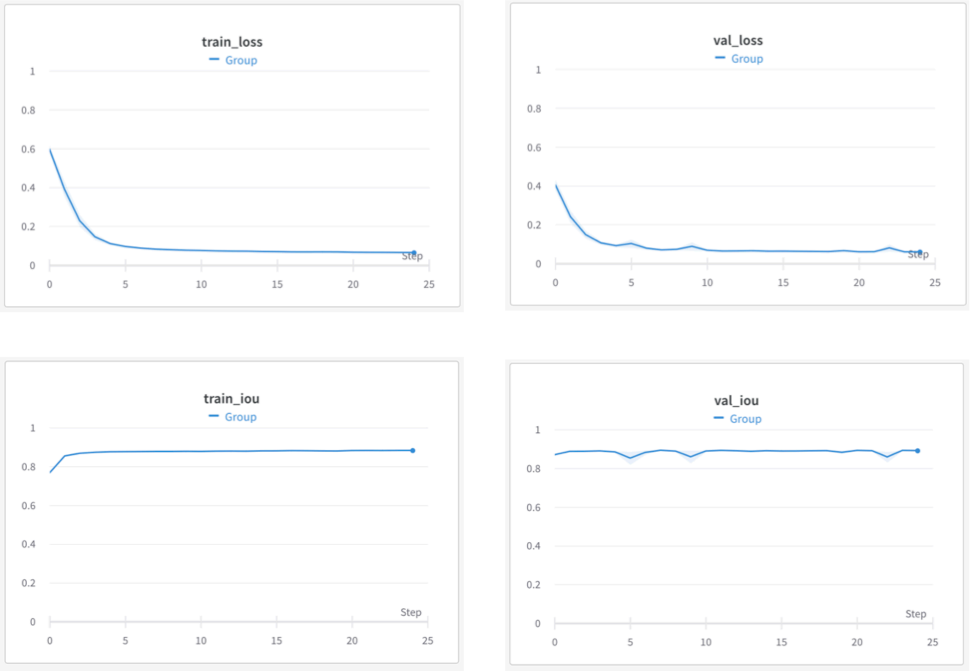

Figure 4.4: Average training and validation loss and IOU obtained with the UNET model. The light blue area visualizes the standard error

Table 4.4 shows the 95% confidence intervals of the following variables obtained on the Svartådalen test set:

- IOU obtained with the Otsu method;
- IOU obtained with the UNET student model;
- Difference between the IOU obtained by UNET and the IOU of Otsu (positive values show that UNET performed better).

The Table shows each variable's lower bound of confidence interval (LB), the mean value, and the upper bound (UB).

Table 4.4: IOU results UNET vs. Otsu

IOU Otsu			IOU UNET			IOU UNET - IOU Otsu		
LB	Mean	UB	LB	Mean	UB	LB	Mean	UB
0.593	0.613	0.632	0.872	0.883	0.894	**0.252**	0.270	0.282

Positive values of the LB of the "difference variable" are highlighted in bold characters and confirm that with 95% confidence, the UNET model performs better than Otsu.

4.4.3 UNET++ results

Figure 4.5 shows the plots of training loss, validation loss, training IOU, and validation IOU obtained by the best UNET++ model in accurate mode and with the Dice Loss function (see Section 2.2.3). The hyperparameters of the best model are summarised in Table 4.1.

The training and validation set consisted of the Örebro data set of 2020-06-23. The applied data augmentation doubled the original data size and included random tile rotation in the range [3°- 5°].

Each graph shows the mean value of the best model over a sample of ten experiments performed with ten different randomly selected seeds in the range of 1 to 100. The lighter blue area in each plot shows the standard error. The graphs show that the best UNET++ model converges within 25 epochs.

Table 4.5 shows the 95% confidence intervals of the following variables obtained on the Svartådalen test set:

- IOU obtained with the Otsu method;
- IOU obtained with the UNET++ student model;

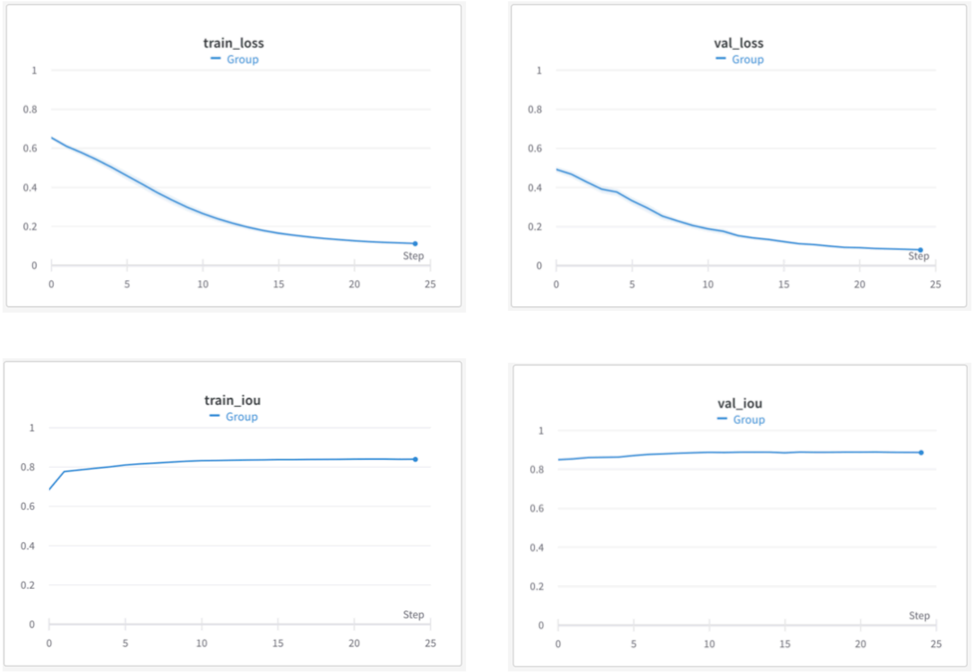

Figure 4.5: Average training and validation loss and IOU obtained with the UNET++ model. The light blue area visualizes the standard error

- Difference between the IOU obtained by UNET++ and the IOU of Otsu (positive values show that UNET++ performed better).

The table shows each variable's lower bound of confidence interval (LB), the mean value, and the upper bound (UB).

Table 4.5: IOU results UNET++ vs. Otsu

IOU Otsu			IOU UNET++			IOU UNET++ - IOU Otsu		
LB	Mean	UB	LB	Mean	UB	LB	Mean	UB
0.593	0.613	0.632	0.873	0.884	0.895	**0.253**	0.271	0.289

Positive values of the LB of the "difference variable" are highlighted in bold characters and confirm that with 95% confidence, the UNET++ model performs better than Otsu.

4.4.4 Discussion

The experimental results of this project show that the three student models, SegNet, UNET, and UNET++, trained using NDWI as the teacher model on the training dataset collected in Örebro performed better in terms of IOU than the baseline Otsu with the Svartådalen test set (Tables 4.3, 4.4 and 4.5) showing both higher mean IOU values and smaller standard deviations (Table 4.2).

Data augmentation improved all the models' accuracy performances (table 4.2). The combination of additive Gaussian Noise and random rotations in small angle ranges from 2°to 5°gave the best performance among all the tested augmentation options.

Simpler models like SegNet and UNET converged after five epochs, while the relatively more complex UNET++ converged within 25 epochs.

The visual comparison of the model's predictions (figure 4.2) confirms that all the student models performed better than the baseline (which appears to suffer from higher false positive predictions). The different student models graphically show substantially similar performances with 'snow' and 'no snow' (in the 'snow' condition, all the models show similar higher false negative predictions).

Table 4.6 shows the 95% confidence intervals of the following variables obtained on the Svartådalen test set:

- Difference between the IOU obtained by UNET and the IOU of Segnet;
- Difference between the IOU obtained by UNET++ and the IOU of Segnet;
- Difference between the IOU obtained by UNET++ and the IOU of UNET.

The table shows each variable's lower bound of confidence interval (LB), the mean value, and the upper bound (UB). In this case, I have calculated the Upper and Lower bound values using the Bonferroni correction (par. 3.4.4), with n (the number of comparisons performed) equal to 3. Positive values of the LB of the "difference variable" are highlighted in bold black characters and confirm that the minuend model performs better than the subtrahend with a 95% confidence. Negative values of the LB or the mean of the "difference variable" are highlighted in bold red characters and show that there is no statistical evidence that the "minuend" model performs better than the "subtrahend" with a 95% confidence.

Table 4.6: IOU results SegNet vs. UNET vs. UNET++

UNET - SegNet			UNET++ - SegNet			UNET++ - UNET		
LB	Mean	UB	LB	Mean	UB	LB	Mean	UB
0.012	0.020	0.028	**0.013**	0.021	0.030	**-0.004**	**0.001**	0.007

In conclusion, the analysis of the confidence intervals in table 4.6 shows that with the Svrtådalen dataset, the UNET and UNET++ models performed better in terms of IOU than SegNet with a confidence level of 95%. **At the same time, even if UNET++ showed the highest performance, there is no statistical evidence that UNET++ performs better than UNET with the same confidence in the considered experiment conditions**.

The UNET architecture, thanks to the skip connections, manages to deliver higher segmentation accuracy compared to the SegNet architecture without requiring a substantially higher number of parameters in the model (the two architectures utilize the same number of convolutions and max-pooling blocks). The UNET++ model, due to the addition of the dense convolutional blocks, is more complex and has a higher number of parameters, which results in a slightly longer time to converge and implies that a higher amount of data might be beneficial for training compared to the other architectures. Additionally, while this project followed the method described in Section 2.3 and trained all the architectures with the Dice Loss (Section 2.3.1), in their original paper [10], the UNET++ authors used a combination of Dice Loss and Binary Cross Entropy. Therefore, further experiments adding the Binary Cross Entropy Loss for UNET++ and utilizing an enlarged or more augmented training dataset are recommended for future comparative studies.

Chapter 5

Conclusions and Future work

This Chapter summarises conclusions (Section 5.1), highlights limitations (Section 5.2), and proposes a few possible directions for future work (Section 5.3).

5.1 Conclusions

This project aimed at investigating the performances of self-supervised learning through cross-modal knowledge distillation in the context of water detection in SAR imagery [1].

The methodology's primary objective is to replace the manual annotation work of the SAR imagery with automatically generated masks through the non-parametric teacher model NDWI [7], which works in the satellite optical domain.

CNN-based student architectures in scope have been SegNet, UNET, and UNET++, while the Otsu method has been used as the baseline.

All the three goals for the projects have been achieved (see Section 1.4):

- This project showed that the self-supervised learning through cross-modal knowledge distillation is viable and performed better in terms of IOU than the baseline for all the CNN architectures in scope trained on the Örebro data set and tested on the Svartådalen data set.

- Segmentation accuracy performances in terms of IOU reached the value of 0.863 ± 0.012 for SegNet, 0.883 ± 0.011 for UNET, and $\mathbf{0.884} \pm \mathbf{0.011}$ for UNET++ with a 95% confidence.

- Statistical analysis with a 95% confidence level showed that the IOU performances of the UNET and UNET++ models are higher than SegNet. Still, there is insufficient evidence regarding the better performances of UNET++ versus UNET.

It is to be noted that the performances obtained in this study with the UNET model are in line with the results presented in [1]. Finally, the hypothesis that the UNET++ model's performances can improve by increasing the size of the training data set is to be further investigated in future comparative work.

Finally, the qualitative visual comparison of the predictions confirms a better performance of the student models versus Otsu and substantially comparable performance for the three student models.

5.2 Limitations

This project focused on verifying the feasibility and assessing the performances of an automated annotation approach based on NDWI. Other alternative methods to replace manual annotation with automatic processing, e.g., self-supervised learning or auto-encoders, have been outside this project's scope.

5.3 Future work

Other suitable future work could include:

- The comparison of UNET++ and UNET student model performances with bigger training data sets and the addition of the Binary Cross Entropy for UNET++.

- The addition of other CNN-based architectures, e.g., UNET3+.

- The verification of performances on other wetlands regions beyond Sweden.

References

[1] 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*, to be published in 2024. [Online]. Available: https://arxiv.org/abs/2305.01698 [Pages ix, 1, 3, 4, 8, 9, 15, 16, 18, 21, 22, 23, 24, 25, 28, 29, 33, 41, and 42.]

[2] U. S. E. P. Agency, "Definition of a wetland," -, 03 2023. doi: -. [Online]. Available: https://www.epa.gov/wetlands/what-wetland [Page 1.]

[3] J. Long and E. Shelhamer, "Fully convolutional networks for semantic segmentation," *CVF*, vol. 11, Nov. 2014. [Online]. Available: https://arxiv.org/abs/1411.4038 [Pages 2 and 17.]

[4] X. Yunchao, Wei Huaxin, "Revisiting dilated convolution: A simple approach for weakly- and semi- supervised semantic segmentation," *CVF*, vol. 5, May 2018. [Online]. Available: https://arxiv.org/pdf/1805.04574.pdf [Page 2.]

[5] L.-C. Chen and G. Papandreou, "Rethinking atrous convolution for semantic image segmentation," *Computer Sciences*, vol. 12, Dec. 2017. [Online]. Available: https://arxiv.org/abs/1706.05587 [Pages 2 and 17.]

[6] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," *CoRR*, vol. abs/1505.04597, 2015. [Online]. Available: http://arxiv.org/abs/1505.04597 [Pages ix, 2, 9, 12, and 17.]

[7] 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, 7 1996. [Online]. Available: https://doi.org/10.1080/01431169608948714 [Pages 2, 7, and 41.]

[8] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, - 2015. doi: 10.1038/nature14539. [Online]. Available: https://doi.org/10.1038/nature14539 [Pages 9 and 17.]

[9] V. Badrinarayanan, A. Kendall, and R. Cipolla, “Segnet: A deep convolutional encoder-decoder architecture for image segmentation,” 2016. [Pages ix, 9, 10, 11, and 17.]

[10] Z. Zhou, M. M. R. Siddiquee, N. Tajbakhsh, and J. Liang, “Unet++: A nested u-net architecture for medical image segmentation,” *CoRR*, vol. abs/1807.10165, Aug. 2018. [Online]. Available: http://arxiv.org/abs/1807.10165 [Pages ix, 9, 12, 13, 14, 17, and 40.]

[11] S. Shurrab and R. Duwairi, “Self-supervised learning methods and applications in medical imaging analysis: a survey,” *PeerJ Computer Science*, vol. 8, 2021. [Online]. Available: https://api.semanticscholar.org/CorpusID:237571561 [Page 14.]

[12] G. Hinton, O. Vinyals, and J. Dean, “Distilling the knowledge in a neural network,” 2015. [Page 14.]

[13] H. Hu, L. Xie, R. Hong, and Q. Tian, “Creating something from nothing: Unsupervised knowledge distillation for cross-modal hashing,” 2020. [Page 15.]

[14] T. A. Soomro, A. J. Afifi, J. Gao, O. Hellwich, M. Paul, and L. Zheng, “Strided u-net model: Retinal vessels segmentation using dice loss,” in *2018 Digital Image Computing: Techniques and Applications (DICTA)*, 2018. doi: 10.1109/DICTA.2018.8615770 pp. 1–8. [Page 16.]

[15] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Advances in Neural Information Processing Systems*, vol. 25. Curran Associates, Inc., 2012. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf [Pages 17 and 18.]

[16] H. Huang, L. Lin, R. Tong, H. Hu, Q. Zhang, Y. Iwamoto, X. Han, Y.-W. Chen, and J. Wu, “Unet 3+: A full-scale connected unet for medical image segmentation,” *arXiv e-prints*, 04 2020. doi: 10.48550/arXiv.2004.08790. [Online]. Available: https://ui.adsabs.harvard.edu/abs/2020arXiv200408790H [Page 17.]

[17] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, “Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs,” 2017. [Page 18.]

[18] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. doi: 10.1109/CVPR.2016.90 pp. 770–778. [Page 18.]

[19] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, “Densely connected convolutional networks,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. doi: 10.1109/CVPR.2017.243 pp. 2261–2269. [Page 18.]

[20] D. Geudtner, R. Torres, P. Snoeij, M. Davidson, and B. Rommen, “Sentinel-1 system capabilities and applications,” in *2014 IEEE Geoscience and Remote Sensing Symposium*, 2014. doi: 10.1109/IGARSS.2014.6946711 pp. 1457–1460. [Page 18.]

[21] B. Slagter, N.-E. Tsendbazar, A. Vollrath, and J. Reiche, “Mapping wetland characteristics using temporally dense sentinel-1 and sentinel-2 data: A case study in the st. lucia wetlands, south africa,” *International Journal of applied Earth Observation and Geoinformation*, vol. 86, Apr. 2020. doi: 10.1016/j.jag.2019.102009 [Page 18.]

[22] 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. PP, pp. 1–14, 10 2021. doi: 10.1109/TGRS.2021.3113856 [Page 18.]

[23] R. M. Schmidt, “Recurrent neural networks (rnns): A gentle introduction and overview,” 2019. [Page 18.]

[24] 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, 11 2022. doi: 10.1016/j.ecoinf.2022.101904 [Page 18.]

[25] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial networks,” 2014. [Page 18.]

[26] G. E. Enngine, “Sentinel-1 pre-processing.” [Online]. Available: https://developers.google.com/earth-engine/guides/sentinel1 [Page 23.]

[27] M. Everingham, S. Eslami, L. Van Gool, C. Williams, J. Winn, and A. Zisserman, “The pascal visual object classes challenge: A retrospective,” *International Journal of Computer Vision*, vol. 111, no. 1, pp. 98–136, Jan. 2015. doi: 10.1007/s11263-014-0733-5 [Page 24.]

[28] N. Otsu, “A threshold selection method from gray-level histograms,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 1, pp. 62–66, 1979. doi: 10.1109/TSMC.1979.4310076 [Page 25.]

[29] C. Bonferroni, *Teoria statistica delle classi e calcolo delle probabilità*, ser. Pubblicazioni del R. Istituto superiore di scienze economiche e commerciali di Firenze. Seeber, 1936. [Online]. Available: https://books.google.se/books?id=3CY-HQAACAAJ [Page 26.]

[30] I. Loshchilov and F. Hutter, “Decoupled weight decay regularization,” 2019. [Page 30.]