TALLINN UNIVERSITY OF TECHNOLOGY
School of Information Technologies

Nikita Budovey 232072IACM

INVESTIGATION OF WATER SEGMENTATION APPROACHES FOR AUTONOMOUS VESSELS

Master's Thesis

Supervisor: Uljana Reinsalu
PhD

Co-supervisor: Karl Janson
PhD

Tallinn 2025

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Nikita Budovey 232072IACM

AUTONOOMSETE LAEVADE VEESEGMENTEERIMISE MEETODITE UURIMINE

Magistritöö

Juhendaja: Uljana Reinsalu
PhD

Kaasjuhendaja: Karl Janson
PhD

Tallinn 2025

Author's Declaration of Originality

I hereby certify that I am the sole author of this thesis. All the used materials, references to the literature and the work of others have been referred to. This thesis has not been presented for examination anywhere else.

Author: Nikita Budovey

20.05.2025

Abstract

Accurate water segmentation plays a vital role in ensuring the safe and autonomous navigation of unmanned surface vehicles (USVs), allowing them to effectively differentiate between water surfaces and potential obstacles. This functionality is essential for key operational tasks such as route planning and collision prevention in dynamic maritime environments. While RADAR and LIDAR technologies are widely used, vision-based systems present a more cost-efficient yet dependable alternative.

This thesis explores computer vision-based water segmentation techniques tailored to the marine conditions of the Gulf of Finland, focusing on their application in USVs.

The study begins with an evaluation of a range of segmentation strategies, from conventional image processing techniques based on color and shape, to modern deep learning-based approaches. Following an extensive literature review, the focus was narrowed to neural network-driven methods due to their superior accuracy and flexibility. Five distinct model architectures were selected for experimental comparison using a custom-built dataset.

This dataset combines images from the Tampere-WaterSeg and USVInland collections, supplemented by 226 frames captured along the Estonian coastline. Model performance was assessed using metrics such as Intersection over Union (IoU), precision, and F1-score, along with inference speed to evaluate real-time feasibility.

A total of twelve neural network configurations were trained under the same conditions. After thorough performance analysis, five models stood out: DeepLabV3+ with Xception backbone, U-Net with ResNet34, WaSR-Net (pretrained), and SegNet with both VGG16 and MobileNetV3 backbones. These configurations demonstrated a strong balance between segmentation accuracy and real-time processing capabilities.

However, testing on the Estonian dataset revealed certain limitations. Some models misclassified elements such as the sky or nearby structures (e.g., piers) as water. To mitigate this, the dataset was rebalanced to improve generalization. Retraining with the updated dataset resulted in improved performance—most notably, the DeepLabV3+ model with Xception achieved an IoU exceeding 97%, outperforming the larger and more

resource-intensive WaSR-Net.

The conducted experiments confirm that deep learning-based segmentation methods are highly effective for distinguishing between aquatic and terrestrial regions. Among the evaluated models, DeepLabV3+ with an Xception backbone delivered the best results, making it a promising candidate for deployment in autonomous USV systems.

The thesis is written in English and is 58 pages long, including 4 chapters, 21 figures and 9 tables.

Annotatsioon

Autonoomsete laevade veesegmenteerimise meetodite uurimine

Täpne vee segmenteerimine on ülioluline mehitamata veesõidukite (USV-de) ohutuks ja autonoomseks navigeerimiseks, võimaldades neil eristada veepinda ümbritsevatest takistustest. See võimekus toetab olulisi funktsioone, nagu liikumistee planeerimine ja kokkupõrgete vältimine keerulises meremiljöös. Kuigi RADAR- ja LIDAR-tehnoloogiad on laialdaselt kasutusel, pakuvad visioonipõhised lahendused kuluefektiivset ja usaldusväärset alternatiivi.

Käesolev magistritöö uurib visioonipõhiseid vee segmenteerimise meetodeid, mis on kohandatud Soome lahe meretingimustele ning suunatud USV-dele.

Uurimistöö algab erinevate segmenteerimistehnikate analüüsiga, hõlmates nii traditsioonilisi, värvi- ja kujupõhiseid meetodeid kui ka kaasaegseid süvaõppepõhiseid lähenemisi. Pärast põhjalikku kirjanduse ülevaadet keskenduti üksnes närvivõrkudel põhinevatele meetoditele, kuna need pakuvad suuremat täpsust ja kohanemisvõimet. Katsetamiseks valiti viis erinevat mudelit, mida hinnati spetsiaalselt koostatud andmestiku põhjal.

Andmestik koosneb Tampere-WaterSeg ja USVInland pildikogumitest, mida täiendavad 226 kaadrit, mis on salvestatud Eesti rannikuvetes. Mudelite sooritust hinnati selliste näitajate alusel nagu IoU (katvuse indeks), täpsus (precision), F1-skoor ning järeldustekiirus, et hinnata sobivust reaalajas kasutamiseks.

Kokku treeniti kaksteist närvivõrgupõhist mudelikonfiguratsiooni identsetes tingimustes. Analüüsi tulemusena osutusid kõige sobivamateks DeepLabV3+ Xception-taustvõrguga, U-Net ResNet34-ga, eeltreenitud WaSR-Net ning SegNet koos VGG16 ja MobileNetV3 taustvõrkudega. Need mudelid näitasid head tasakaalu täpsuse ja reaalajas töövõime vahel.

Testides mudeleid Eesti rannikul jäädvustatud piltidel ilmnesid siiski mõned probleemid. Näiteks tuvastati valesti taevas veealana või vastupidi. Samuti tõlgendasid mudelid mõnikord suuri lähedalasuvaid objekte – näiteks kai – ekslikult veepinnana, mis võib põhjustada tõsiseid vigu stseeni mõistmisel.

Nende probleemide lahendamiseks ja üldistamise võime parandamiseks tasakaalustati andmestik ümber. Pärast mudelite uuesti treenimist paranesid tulemused oluliselt. Näiteks DeepLabV3+ koos Xception-taustvõrguga saavutas üle 97% IoU tulemuse, edestades isegi ressursimahukamat WaSR-Net mudelit.

Töö eksperimentide ja analüüsi põhjal võib kindlalt järeldada, et närvivõrkudel põhinevad meetodid on väga tõhusad vee- ja maismaa-alade eristamisel. Testitud mudelitest osutus parimaks DeepLabV3+ Xception-taustvõrguga, muutes selle tugevaks kandidaadiks autonoomse veesõiduki pardasüsteemi integreerimiseks.

Lõputöö on kirjutatud inglise keeles ning sisaldab teksti 58 leheküljel, 4 peatükki, 21 joonist, 9 tabelit.

List of Abbreviations and Terms

ARM	Attention Refinement Modules
ASV	Autonomous Surface Vessels
ASPP	Atrous Spatial Pyramid Pooling
DNN	Deep Neural Networks
DSLR	Digital Single-Lens Reflex camera
CNN	Convolution Neural Networks
FCN	Fully Convolutional Networks
FMM	Feature Fusion Module
FN	False Negatives
FPS	Frames Per Second
FP	False Positives
GPU	Graphics Processing Unit
ICVDS	IntCatch Vision Data Set
IMU	Inertial Measurement Unit
IoU	Intersection over Union
mIoU	mean Intersection over Union
MLP	Multi-Layer Perceptron
MNDWI	Modified Normalized Difference Water Index
mPA	mean Pixel Accuracy
NDWI	Normalized Difference Water Index
LIDAR	Light Detection And Ranging
LoRA	Low-Rank Adaptation
RADAR	Radio Detection And Ranging
RGB	Red Green Blue
RIWA	River Water Segmentation Dataset
HSV	Hue, Saturation, and Value
SLIC	Simple Linear Iterative Clustering
TN	True Negatives
TP	True Positives
UAV	Unmanned Aerial Vehicle
USV	Unmanned Surface Vehicle
ViTs	Vision Transformers
WaSR	Water-Obstacle Separation and Refinement Network

Table of Contents

List of Figures

List of Tables

1. Introduction

Accurate and reliable water segmentation is a fundamental capability for the safe and effective operation of unmanned surface vehicles (USVs). This process, which involves distinguishing the water surface from other elements in the environment such as land, rocks, vessels, and other obstacles, provides the essential foundation for autonomous navigation and informed decision-making. A fully autonomous system heavily relies on a robust collision avoidance mechanism, and the ability to precisely identify navigable water is paramount to minimizing risks to both life and assets within the complex marine environment. Currently, the common approach implemented in obstacle avoidance systems for marine vehicles are RADAR and LIDAR. A cheaper yet accurate and effective solution possible is by applying vision-based detection.

However, achieving precise water segmentation in maritime settings presents a unique set of challenges. Unlike controlled environments, the open sea and inland waterways are subject to considerable variability. Existing methods that rely solely on monocular images often struggle with fluctuations in illumination and adverse weather conditions. The visual data acquired can be significantly affected by high variance in scene properties, including differing light levels and the presence of reflections. Furthermore, the color characteristics of water can vary greatly, sometimes exhibiting similarities to surrounding terrain like brownish trees, which complicates visual classification tasks that heavily depend on color information. The dynamic and unpredictable nature of the marine environment, influenced by factors such as sun glare and sea fog, further underscores the complexity of developing a dependable situational awareness system.

Chapter 2 presents the theoretical and contextual background of the study, Chapter 3 outlines the experimental results obtained under various conditions and model configurations, and Chapter 4 offers a detailed conclusion based on the research findings and outcomes.

2. Background

Segmenting water for autonomous vessels involves identifying water regions in images to ensure safe navigation. The two primary approaches to data segmentation can be categorized as traditional methods and those based on neural networks. Traditional methods, such as edge detection and thresholding, are straightforward but often fail in complex maritime settings. Neural networks, particularly deep learning models, provide superior accuracy and adaptability, making them the preferred choice for modern autonomous vessels.

Traditional methods rely on hand-crafted rules to segment water. For example, edge detection identifies boundaries, while thresholding separates water based on color or intensity. These methods are fast and don't need training data, but they struggle with reflections, varying lighting, or complex scenes, leading to unreliable results in real-world navigation.

Deep learning models like U-Net, SegNet, and WaSR-Net use convolutional neural networks to classify pixels as water or non-water. They learn from large datasets, achieving high accuracy and handling diverse conditions. However, they require significant computational power and annotated data, which can be a challenge for smaller vessels.

In this chapter, an evaluation of both approaches will be conducted based on the reviewed literature. The more suitable method will then be selected for further adaptation and application to the specific conditions of the Baltic Sea coastline.

2.1 Overview of Traditional Architectures for Water Segmentation

While deep learning has shown significant promise, traditional computer vision techniques also offer viable solutions for water segmentation in certain scenarios. These methods often rely on analyzing specific image features and can be computationally less demanding than neural networks.

2.1.1 Color-Based Segmentation

Color is an intuitive feature for distinguishing water from many non-water surfaces. Color-based segmentation techniques typically involve defining a range of color values in a specific color space (e.g., RGB, HSV) that correspond to water. Pixels falling within this

range are classified as water, while others are classified as non-water [1]. However, the effectiveness of this approach can be limited by variations in water color due to factors like sediment, algae, and lighting conditions. Reflections of the sky or surrounding objects on the water surface can also introduce colors that might lead to misclassifications [2]. Analyzing the robustness of different color spaces, such as RGB and HSV, can help in selecting a representation that is less sensitive to certain types of variations.

2.1.2 Texture-Based Segmentation

Texture analysis examines the spatial arrangement of pixel intensities to identify patterns that can differentiate between water and other surfaces. Water surfaces, especially when calm, often exhibit a relatively smooth texture compared to land, rocks, or vegetation. Techniques like wavelet texture analysis can be used to extract these textural features and segment the image accordingly [3]. However, the presence of waves, ripples, or floating debris can introduce complex textures on the water surface, potentially complicating the segmentation process.

2.1.3 Edge Detection Techniques

Identifying the boundary between water and non-water regions is a crucial aspect of segmentation. Edge detection algorithms aim to locate sharp changes in pixel intensity that correspond to these boundaries. While these techniques can be effective in clearly defined scenarios, they might be sensitive to noise and reflections, which can create spurious edges and lead to fragmented or inaccurate water boundaries [4]. An edge-aware approach, as seen in the ELNet method, suggests that incorporating edge information can enhance the segmentation process.

2.1.4 Geometric Feature-Based Methods

Geometric features such as shape and size can sometimes be used to identify water bodies, particularly in aerial views where the overall shape of lakes or rivers might be discernible. However, for on-surface marine vehicles with a limited field of view, relying solely on geometric features might be less practical as the entire extent of the water body might not be visible [5].

2.1.5 Water Index Methods (NDWI, MNDWI)

Water index methods, such as the Normalized Difference Water Index (NDWI) and the Modified Normalized Difference Water Index (MNDWI), are commonly used in remote sensing to detect water bodies from multispectral imagery. These indices utilize the different spectral reflectance properties of water in specific bands of the electromagnetic spectrum, typically involving green, near-infrared, and mid-infrared bands. NDWI, for example, uses the green and near-infrared bands to enhance water features and suppress vegetation [6]. MNDWI replaces the near-infrared band with the mid-infrared band to reduce the influence of buildings and soil on water extraction . While these methods offer fast computation and have been widely applied, they can still face challenges with shadows, differentiating water from certain land covers, and require careful selection of optimal thresholds, which can vary depending on the specific environment.

2.1.6 Advantages of Traditional Methods

Traditional computer vision techniques for image segmentation offer several advantages, particularly in terms of computational efficiency and ease of implementation [7]. Compared to the complex computations involved in deep learning, these methods often rely on simpler mathematical operations, making them less costly in terms of processing power and potentially more suitable for real-time applications on resource-constrained embedded systems [8]. Their relative simplicity can also make them easier to understand and implement.

2.1.7 Analysis and Discussion

Despite their advantages, traditional computer vision algorithms often have limitations in robustness and scalability when faced with the complexities of real-world maritime environments. They can struggle with varying environmental conditions, such as changes in illumination and the presence of reflections, which can significantly impact their accuracy. Many traditional methods require manual tuning of parameters and might rely on specific assumptions about the scene that might not always hold true, limiting their adaptability and generalization capabilities compared to the automatic feature learning of deep learning approaches. For instance, thresholding techniques can be highly sensitive to noise and might perform poorly when there is significant overlap in intensity values between the water and non-water regions.

2.2 Overview of Neural Network Architectures for Water Segmentation

Deep learning has emerged as a powerful paradigm for tackling complex computer vision tasks, and water segmentation for autonomous marine vehicles is no exception. Among the various neural network architectures, Convolutional Neural Networks (CNNs) designed for semantic segmentation have shown significant promise. These networks aim to classify each pixel in an image into predefined categories, such as water and non-water. Several architectures have been extensively studied and applied to this specific problem.

Fully Convolutional Networks (FCNs) were among the pioneering deep learning models adapted for semantic segmentation. These networks replace the fully connected layers of traditional CNNs with convolutional layers, enabling them to process images of arbitrary sizes and produce pixel-wise predictions. SegNet is another architecture that has gained traction, known for its encoder-decoder structure and the use of pooling indices to achieve efficient upsampling in the decoder, leading to precise segmentation maps. U-Net, with its distinctive U-shaped architecture and skip connections between the encoder and decoder, has also demonstrated remarkable effectiveness in various segmentation tasks, including water segmentation. These skip connections facilitate the fusion of high-resolution features from the encoder with the upsampled features from the decoder, preserving fine-grained details crucial for accurate boundary delineation.

Additionally, models like DeepLabv3+ have been investigated for their ability to perform semantic segmentation by employing techniques such as Atrous Spatial Pyramid Pooling (ASPP) to capture contextual information at multiple scales. The trend towards developing lighter network architectures suggests a growing emphasis on deploying these sophisticated models on the embedded systems commonly found in autonomous vehicles, where computational resources might be limited.

Below is a more detailed overview of the models that can be used to segment water from non-water.

2.2.1 Fully Convolutional Network (FCN)

FCNs, introduced in 2015, are the seminal pixel-wise segmentation models introduced by Long et al. [9] They replace a standard classification network's fully-connected layers with convolutional layers, allowing an input image of arbitrary size to produce a correspondingly-sized dense output. In practice, FCNs are built on classification backbones (e.g. VGG,

ResNet) and use learned upsampling (deconvolution) to recover full-resolution segmentations. They fuse coarse, deep semantic features with fine, shallow appearance features to refine object boundaries.

The architecture of an FCN typically consists of two main paths: a downsampling path, often referred to as the encoder, and an upsampling path, known as the decoder [10]. The downsampling path is responsible for extracting hierarchical features from the input image, progressively reducing its spatial resolution while learning increasingly complex representations. This part of the network often employs a series of convolutional layers followed by pooling layers. The upsampling path then takes these low-resolution feature maps and gradually increases their spatial resolution back to that of the original input image, ultimately producing a dense segmentation map where each pixel is assigned a class label. The Figure 1. shows the architecture visually:

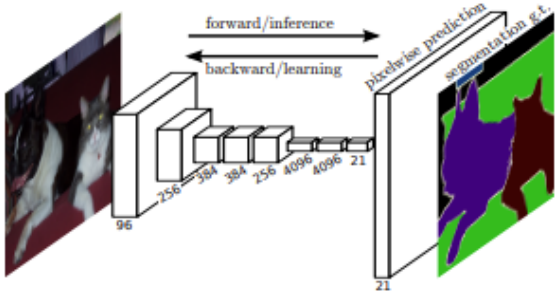

Figure 1. *FCN model architecture [9]*

A key characteristic of FCNs is their ability to process input images of arbitrary sizes, a direct consequence of the absence of fully connected layers which typically impose fixed input dimensions [11]. Existing pre-trained classification networks, such as AlexNet, VGG, and GoogLeNet, which have demonstrated strong performance on image recognition tasks, can be effectively adapted into FCNs by converting their fully connected layers into convolutional layers. The original paper on FCNs proposed three main variants: FCN-32s, FCN-16s, and FCN-8s. These variants differ primarily in the output stride of the network and the number of skip connections used, with FCN-8s being the most complex, incorporating skip connections from earlier layers to produce more refined segmentation maps [9].

Several studies have evaluated the performance of FCNs for water segmentation using various datasets and metrics. Lopez-Fuentes et al. [12] reported an Mean Intersection over Union (MIoU) of 70.05% for water segmentation using the FCN-8s variant. Mohd Adam et al. [2] when tested on the IntCatch Vision Data Set (ICVDS), FCN demonstrated an accuracy of 95.55%, a precision of 96.69%, a recall of 93.3%, and an F1-score of 94.09%, with an inference speed of 11 frames per second (FPS). However, when the same

FCN model was evaluated on the unseen Malaysia ASV Dataset (MASVD), its accuracy dropped significantly to 65.17% with an FPS of 4, indicating a lack of generalization to new environments with different visual characteristics. It was also noted that FCN tended to miss fine details near the boundaries between water and non-water regions, which contributed to lower F1-scores compared to models like U-Net and SegNet when tested on the ICVDS dataset.

In terms of advantages, FCNs are fully convolutional and relatively simple. They require no post-processing and can be trained end-to-end with relatively few modifications to standard classification nets. They also allow inputs of varying size and are efficient to run, particularly with modern GPUs. HDespite these benefits, vanilla FCNs have important limitations. The aggressive downsampling in the encoder causes loss of fine detail, making FCNs poor at capturing small water regions, thin shorelines or fine edges [3]. FCNs also tend to miss small obstacles or subtle transitions (e.g. wet patches or reflections) because spatial detail is lost. In USV scenarios, where precise edge detection (waterline) and small obstacle detection are critical, pure FCNs typically underperform.

2.2.2 U-Net

The U-Net is a symmetric encoder–decoder network originally developed for biomedical image segmentation [13]. Its encoder (contracting path) is typically a standard CNN (often with repeated conv+pool layers), while the decoder (expanding path) uses up-convolutions to reconstruct the output. Crucially, U-Net introduces skip connections: feature maps from each encoding layer are concatenated with the corresponding decoder layer. This "U" shape allows high-level semantic features to combine with low-level spatial detail, thereby preserving fine structures. In practice, for water segmentation a U-Net can be built with any CNN backbone; the decoder's skip connections help recover water boundary detail lost in pooling.

The U-Net architecture Figure 2. offers several key advantages that make it highly suitable for the task of water segmentation in images. It is known for producing accurate segmentation maps, particularly when working with high-resolution images or datasets with a large number of classes. The skip connections, a defining feature of U-Net, play a crucial role by allowing the model to incorporate both high-level semantic information from the deeper layers and low-level spatial features from the shallower layers of the encoder. This fusion of information enables precise localization of water boundaries, which is essential for accurate segmentation. U-Net is also effective in handling multi-class image segmentation tasks, making it capable of distinguishing water from various other elements in a scene. The use of skip connections further contributes to the efficient utilization of training data,

allowing the model to learn robust features even with smaller datasets. The architecture is designed to capture both the overall context of the image and the precise location of the objects of interest. In the context of water segmentation, U-Net has demonstrated its ability to achieve high pixel accuracy in identifying water regions within images.

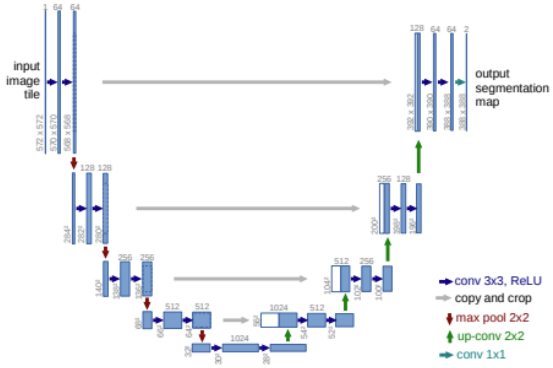

Figure 2. *U-net model architecture [13]*

Despite improvements over FCN, U-Net is still quite heavy and may not run in real-time on embedded USV hardware without acceleration. The large number of feature maps in decoder layers can incur significant computation and memory cost. Furthermore, while skip connections preserve detail, directly concatenating features at different resolutions can sometimes "confuse" the network if low- and high-level features are not balanced properly [14]. This may require careful tuning (e.g. attention weighting) to avoid degrading accuracy. Finally, like FCN, a plain U-Net lacks global context aggregation: it may still miss very large-scale context compared to ASPP-based networks (like DeepLab) or transformer models.

Vandaele et al. [15] have reported on the performance of U-Net for water segmentation across various datasets. One study focused on transfer learning for river water segmentation achieved a high pixel accuracy of 97.45% using a fully convolutional network based on U-Net. Steccanella et al. [16] made improvements to SegNet and reported an impressive pixel segmentation accuracy of 98.8% with an inference speed of 4.5 frames per second (FPS) on 160×160 images. Alhadi et al. [17] used Sentinel-2 data, the validation accuracy was 82.92% for water zone segmentation using U-Net architecture. A comparative analysis of simple U-Net, Residual Attention U-Net, and VGG16-U-Net on Sentinel-2 imagery revealed that VGG16-U-Net achieved the best mean-IoU score of 98.5%.

2.2.3 SegNet

SegNet is a deep learning architecture specifically designed for semantic segmentation tasks, aiming to classify each pixel in an image into a predefined category. The model was

realesed by Badrinarayanan et al. [18] in 2017. It follows an encoder-decoder neural network structure which Figure 3. shows, which is tailored for pixel-wise image segmentation, making it highly effective for tasks requiring detailed and precise segmentation of images.

The encoder network in SegNet is composed of 13 convolutional layers, mirroring the first 13 convolutional layers of the VGG16 network, which was originally designed for object classification. This design choice allows the encoder to benefit from the pre-trained weights of VGG16, facilitating efficient initialization and faster convergence during training. Similar to VGG16, the encoder performs convolution operations to extract features from the input image, followed by batch normalization and ReLU activation functions. Max-pooling operations are applied between blocks of convolutional layers to progressively reduce the spatial dimensions of the feature maps while increasing their depth.

A key innovation of SegNet lies in its decoder network, which is designed to upsample the low-resolution feature maps produced by the encoder back to the original input resolution. Unlike other architectures that might use learnable deconvolution layers for upsampling, SegNet utilizes the pooling indices computed during the max-pooling step in the corresponding encoder layers to perform non-linear upsampling in the decoder. This approach eliminates the need for the decoder to learn the upsampling process, making SegNet more efficient in terms of both storage and computation. The upsampled feature maps are initially sparse, containing the feature values at the locations indicated by the max-pooling indices, with other locations set to zero. These sparse maps are then convolved with trainable decoder filters to produce dense, high-resolution feature maps. Similar to FCNs, SegNet does not include any fully connected layers, making it a fully convolutional network suitable for end-to-end pixel-wise prediction. This architecture is designed to be efficient in terms of both memory usage and computational time during inference, and it typically has a smaller number of trainable parameters compared to other competing architectures.

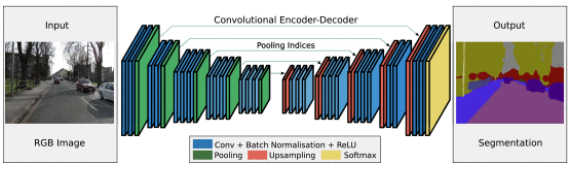

Figure 3. *SegNet model architecture [18]*

SegNet still inherits many of U-Net's downsides: it is a rather deep network and not optimized for real-time on small devices. Also, while pooling indices help upsample, SegNet does not explicitly fuse multi-scale context (no ASPP), so it may miss contextual cues compared to DeepLab or transformer methods. In practice, SegNet often lags behind

U-Net and more recent architectures in segmentation accuracy.

Ammar bin Mohd Adam et al. [2] have evaluated SegNet's performance in water segmentation tasks. On the IntCatch Vision Data Set, SegNet demonstrated high performance, achieving the highest scores for precision 95.77% and recall 97.57% with an inference speed of 32 frames per second (FPS) compared to FCN (11 frames) and U-Net (17 frames). Muhadi et al. [19] have evaluated for river water segmentation using RGB sensors, SegNet achieved a high accuracy of 98% and an IoU of 97% for both 256x256 and 512x512 input resolution.

2.2.4 DeepLabV3+

DeepLabv3+ is a state-of-the-art semantic segmentation architecture that builds upon the foundations of the DeepLab series, particularly DeepLabv3, by incorporating a simple yet highly effective decoder module. This addition is primarily aimed at refining the segmentation results, especially along the boundaries of objects within an image. The architecture follows an encoder-decoder structure, a common paradigm in semantic segmentation where the encoder extracts semantic information and the decoder aims to recover spatial details. [20]

A key component of DeepLabv3+ is the Atrous Spatial Pyramid Pooling (ASPP) module, which is typically located within the encoder part of the network. Figure 4. shows the architecture of DeepLabV3. ASPP is designed to capture multi-scale contextual information from the input feature maps by applying several parallel atrous convolutions with different dilation rates. Atrous convolution, also known as dilated convolution, allows for the expansion of the receptive field of the convolutional filters without increasing the number of parameters, which is crucial for understanding context at various scales.

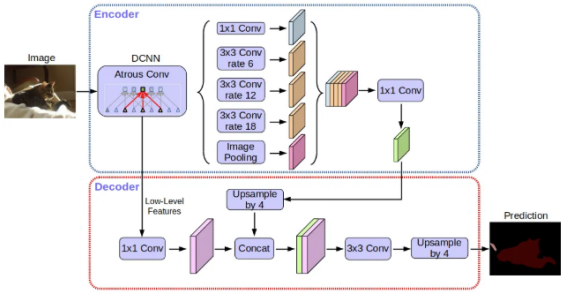

Figure 4. *DeepLabV3 model architecture [20]*

For water vs. non-water segmentation, DeepLabv3+ offers high accuracy. The multi-scale atrous filters help it capture complex shapes of water bodies, and the encoder–decoder refinement sharpens edges. Compared to earlier architectures, DeepLabv3+ achieves very

high IoU on many datasets. For instance in the work of Muhadi et al. [21], DeepLabv3+ achieved about 94.12% IoU. This was notably higher than SegNet's 91.05%.

However, DeepLabv3+ has disadvantages in this domain. It is a large model with many parameters and high computational cost. Using a deep backbone and dense ASPP means inference is slower and needs more memory. This makes it less suitable for real-time or embedded systems (unlike faster networks). In practice, DeepLabv3+ runs at only a few FPS on standard GPUs for high-resolution images. For water segmentation, this means limited applicability in resource-constrained environments. Another limitation is that DeepLabv3+ may still struggle with very small water features if they are below the scale of the atrous filters.

Numerous studies have evaluated the performance of DeepLabv3+ for water segmentation. On surveillance images, DeepLabv3+ achieved accuracy and IoU metrics 96.72% and 94.12%, and a boundary F1 score around 82.4%, outperforming SegNet in the same evaluation [21]. Xue et al. [22] presented a deep semantic segmentation model enhanced with the Simple Linear Iterative Clustering (SLIC) superpixel algorithm. Where the model was taken as DeepLabV3+ with ResNet101 backbone. The model achieved 90.1% of mIoU. In another work by Han et al. [3], the model DeepLabV3 achieved 99.1% and 98.09% of mean pixel accuracy (mPA) and mIoU, respectively.

2.2.5 BiSeNet

The Bilateral Segmentation Network (BiSeNet) (Yu et al. [23]) was designed for real-time semantic segmentation. BiSeNet employs a two-branch design: a Spatial Path and a Context Path. The Spatial Path consists of a few convolutional layers with small stride to preserve high-resolution spatial information. The Context Path is a lightweight backbone (originally a truncated Xception) with fast downsampling to gather a large receptive field. Features from both paths are fused via a Feature Fusion Module. The idea is to balance rich spatial detail with contextual understanding efficiently. In application, a BiSeNet model for USVs would feed the image into both branches simultaneously: the Spatial Path keeps full image resolution, while the Context Path quickly compresses it. Their outputs are then merged to produce the final segmentation mask. In the Figure 5. is shown the architecture of BiSeNet:

BiSeNet offers significant advantages for water segmentation, primarily due to its design that prioritizes real-time performance, making it suitable for applications where speed is critical. The dual-path structure of BiSeNet is particularly beneficial for water segmentation as it allows the network to simultaneously preserve the detailed spatial information, which

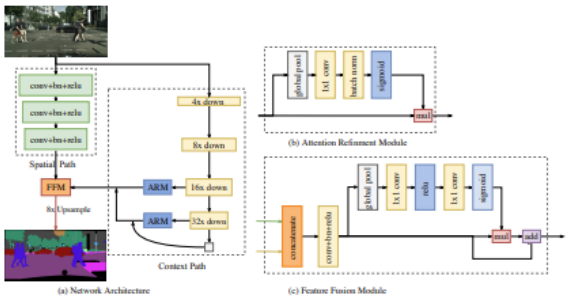

Figure 5. *BiSeNet model architecture [23]*

is essential for accurately delineating water boundaries, while also capturing the broader semantic context of the scene, which helps in distinguishing water from other spectrally similar elements like shadows or certain types of land cover. This architecture enables BiSeNet to achieve high frames per second (FPS) during inference while maintaining a reasonable level of segmentation accuracy, making it well-suited for real-time applications such as autonomous navigation on unmanned surface vehicles.

The flip side of BiSeNet's design is that its accuracy is limited by the constraints of its two-stream architecture. It may not reach the top accuracy of DeepLabv3+ or transformer models, especially on very challenging fine structures. The shallow Spatial Path has few layers, so its features may be too low-level. The original BiSeNet paper noted a trade-off: to maintain real-time speed, spatial path width is limited, which can reduce segmentation quality on very detailed images. In USV applications, BiSeNet might struggle with extremely small obstacles or complex reflections compared to specialized networks.

Han et al. [4] have evaluated the performance of BiSeNet and its variants for water segmentation. On the USVInland dataset with quantitative Evaluation of segmentation methods , BiSeNet achieved a precision of 97.89% and an F-score of 93.04%, while maintaining a frame rate of 36.12 FPS. SA-BiSeNet is presented by Zhang et al. [24], a modification incorporating a swap attention mechanism, achieved a high Mean IoU of 93.65% on the same USVInland dataset, with an inference speed comparable to the baseline BiSeNet (37.33 FPS). The study of Fan et al. [25] focusing on real-time performance, Rethinking BiSeNet (using the STDC network), reported a mIoU of 71.9% on the Cityscapes test set with a significantly faster speed of 250.4 FPS. Additionally, GFANet which was presented by Xie et al. [26], which employs a lightweight backbone and shares architectural similarities with BiSeNet in its aim for efficiency, achieved comparable segmentation accuracy to more complex models with a fast inference speed of 154.98 FPS on the Riwa dataset (mIoU 82.29%, mPA 89.49%), and also showed comparable performance on the USVInland and WaterSeg datasets.

2.2.6 Habaek

The Habaek network is a recently proposed architecture specifically designed for achieving high-performance water segmentation in images. The model was presented by Joo et al. [27] in 2024. Its foundation lies in the SegFormer model, a framework that utilizes Transformer networks for semantic segmentation. SegFormer itself is characterized by a hierarchical Transformer encoder, which is capable of outputting multiscale features, and a lightweight all-MLP (Multi-Layer Perceptron) decoder. A notable advantage of SegFormer is that it does not rely on positional encoding, which avoids potential performance degradation when the resolution of the test image differs from that used during training.

The development of the Habaek network involves upgrading the SegFormer model, which is shown in the Figure 6., through several key strategies. One of these is the use of data augmentation techniques, employing additional datasets such as ADE20K and RIWA, to enhance the model's ability to generalize to a wider range of water-related imagery. The research also examines the impact of inductive bias on attention-based models, noting that Vision Transformers (ViTs), like the one used in SegFormer, typically have less inductive bias compared to Convolutional Neural Networks (CNNs). This lower inductive bias allows ViTs to learn more complex relationships from data but often requires larger datasets to achieve optimal performance. To address the computational demands often associated with Transformer models, Habaek utilizes Low-Rank Adaptation (LoRA). LoRA is a technique that reduces the complexity of the model by decomposing the weight matrices into smaller, lower-rank matrices, which are then updated during training while keeping the original weights frozen. This approach allows for efficient adaptation of pre-trained models to specific tasks like water segmentation without a significant increase in computational resources.

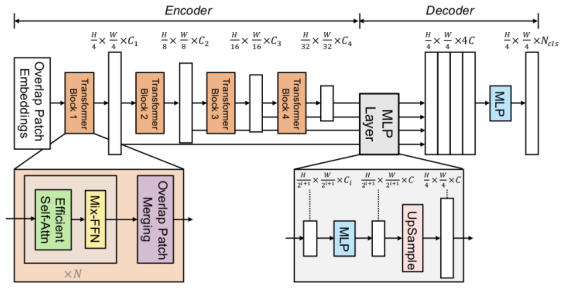

Figure 6. *SegFormer model architecture [27]*

The Habaek network offers several compelling advantages for the task of water segmentation. Notably, it has been shown to outperform existing models in terms of segmentation accuracy, achieving an impressive Intersection over Union (IoU) range of 91.986% to 94.397%. Furthermore, Habaek demonstrates superior performance compared to rival

models across a range of evaluation metrics, including F1-score (97.09%), recall (96.71%), overall accuracy (97.54), and precision (97.48%), indicating its robustness and effectiveness in accurately identifying water bodies.

The disadvantages of Habaek stem from its complexity. Vision transformers are computationally heavy and require more memory and data than CNNs. Even with LoRA, Habaek may be slower at inference than compact CNNs, and it may not be real-time on standard hardware. Furthermore, its reliance on large datasets could limit performance if only limited water imagery is available (which is why it uses transfer from ADE20K). Finally, because it is so new and designed for remote sensing, its behavior on onboard USV imagery is untested; it may not incorporate IMU or edge priors the way WaSR or ELNet do.

2.2.7 WaSR-Net (Water-Obstacle Separation and Refinement Network)

WaSR, which stands for Water Segmentation and Refinement network, is a deep encoder-decoder architecture specifically designed for the task of maritime obstacle detection and accurate water segmentation in marine environments by Bovcon et al. [28]. The development of WaSR is motivated by the challenges inherent in segmenting water in maritime settings, such as the presence of strong reflections, wakes, and varying sea conditions that can lead to inaccurate detection of water edges and false positives for obstacles.

The encoder component of the WaSR network is based on a deep ResNet101 architecture, incorporating atrous convolutions to enable the extraction of rich visual features from the input images. Atrous convolutions help in increasing the receptive field of the network's filters without reducing the spatial resolution of the feature maps, which is crucial for capturing both local and global context in the marine scene. The decoder part of the WaSR network is designed to gradually fuse these visual features with inertial information obtained from an Inertial Measurement Unit (IMU). This fusion of visual and inertial data plays a key role in improving the segmentation accuracy of the water component, particularly in situations where visual ambiguities, such as fog on the horizon or reflections from the water surface, might otherwise lead to errors. The architecture of WaSR network is shown in the Figure 7:

To effectively integrate the inertial information, WaSR constructs an IMU feature channel that encodes the location of the horizon at a pixel level. This IMU-derived information

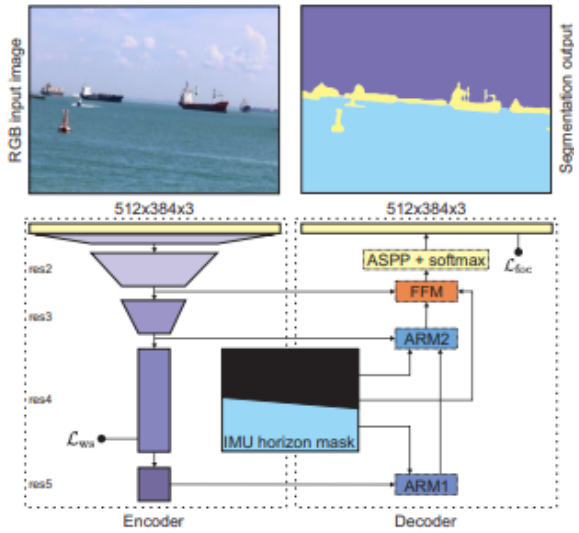

Figure 7. *WaSR model architecture [28]*

serves as a prior probability for the location of water in the scene, aiding in the accurate estimation of the water's edge in the final segmentation output. The IMU mask is treated as an externally generated feature channel that is fused with the encoder features at multiple levels within the decoder. To handle the potential differences in scale between the IMU channel and the encoder features, WaSR employs Attention Refinement Modules (ARM) and Feature Fusion Modules (FFM), which are learned during training to determine the optimal strategy for combining these two distinct types of information. The final block of the decoder incorporates an Atrous Spatial Pyramid Pooling (ASPP) module, followed by a softmax layer. The ASPP module is used to improve the segmentation of small structures, such as buoys or other minor obstacles, and to produce the final semantic segmentation mask.

This design gives WaSR strong advantages in water segmentation accuracy. By explicitly using IMU information and a dedicated separation loss, WaSR greatly improves water-edge estimation and obstacle detection under challenging conditions. Empirically, WaSR "outperforms the current state-of-the-art by a large margin" in maritime segmentation. For example, on the MODD2 benchmark, WaSR achieved 93.7% F1 with a water-edge pixel error of only 9.6 pixels, far ahead of the next best methods (e.g. BiSeNet at 81.9% F1). The authors report that WaSR increases true positives by 8% and greatly reduces false positives/negatives compared to BiSeNet. Qualitatively, WaSR remains robust to severe water reflections, wakes, and mirroring, where other models produce many false obstacles. In summary, WaSR's advantage is its very high segmentation accuracy and robust boundary detection in marine environments, thanks to its sensor fusion and specialized loss.

However, WaSR has significant disadvantages in complexity and deployment. Its backbone

and decoder make it a large, resource-intensive model. As noted in subsequent work, "state-of-the-art models such as WaSR cannot be deployed to lightweight devices due to memory limitations". WaSR typically requires a powerful GPU for real-time inference; on embedded hardware it fails altogether due to its size [29]. The original WaSR runs at only 10 FPS on high-end GPUs, making it unsuitable for fast real-time on standard devices. Furthermore, WaSR's reliance on IMU means it requires synchronization of sensor data and may be less applicable if IMU is unavailable or noisy. In summary, WaSR achieves the highest accuracy on water segmentation tasks, but its disadvantage is extreme computational cost, large memory footprint, and dependence on extra sensors.

2.2.8 Analysis and Discussion

The suitability of each network for water segmentation is contingent on the specific requirements of the application. FCN, U-Net, SegNet and DeepLabv3+ generally provide a simplicity and lightness of the model, as well as the ease of modification and deployment on embedded devices, but the downside is the relatively low accuracy in segmentation, especially at the boundaries. BiSeNet offer a valuable trade-off between accuracy and speed, making them particularly advantageous for real-time applications such as autonomous navigation and rapid environmental monitoring. WaSR is uniquely tailored for the challenges of maritime environments, where reflections and other visual ambiguities are common, but it is a very resource-hungry model. Habaek has demonstrated the potential to achieve state-of-the-art accuracy in water segmentation, especially when large datasets are available, making it ideal for applications where the highest possible precision is required, but the author does not provide any sources for the model itself or an exact description of how it works.

2.3 Interim conclusion about the appropriate method for implementation

These methods are suitable for water segmentation tasks in embedded systems designed for real-time processing. Their effectiveness has been demonstrated in a number of previous studies [2] [3] [4] [28] [27] [23]. While traditional techniques are relatively easy to implement, they often produce unreliable results when deployed in autonomous systems without human supervision. This is largely due to the constantly changing environmental conditions and a wide variety of influencing factors that must be considered—such as water glare, reflections, and the variability of the underwater background. For example, the presence of aquatic vegetation can shift the apparent water color toward green, while sandy or silty bottoms may give it a yellowish or brownish tint.

Given these challenges, traditional methods may be better suited as a preprocessing step for neural network-based models rather than as standalone solutions [22]. Consequently, such traditional approaches are excluded from the scope of this study.

As for neural networks, the choice of model for implementation is not so obvious and the results of each model vary from work to work. In such a situation, it was decided to select the most suitable models for testing and then implement them in our work.

For testing, it was decided to select all models except FCN and Habaek. FCN is a pioneer in image segmentation, but it is quite outdated compared to its successors U-net and SegNet, which show better results in various works [21], [2], [3]. Regarding Habaek, as mentioned earlier, the author did not provide any links or clear instructions for implementing the model. Therefore, it will not be possible to test this model on our data.

2.4 Datasets

Training a robust model for water segmentation, a crucial task for autonomous vessels to perceive their environment, requires a carefully selected dataset. A good dataset should contain a large number of diverse images or video frames, each meticulously labeled with ground truth masks that precisely delineate the water region. The resolution of these images is also an important factor, as higher resolutions can provide more detailed information for the segmentation process. Without accurate and comprehensive ground truth data, a model cannot learn to reliably distinguish water from other elements in a scene, which is essential for safe and effective autonomous navigation.

2.4.1 IntCatch Vision Data Set (ICVDS)

ICVDS is a publicly available dataset designed for water segmentation presented by Steccanella et al. [30] in 2019, particularly in the context of autonomous surface vessels (ASV). It contains water images and videos with ground truth masks, following a format similar to the DAVIS dataset. It has been used for benchmarking deep learning models for binary semantic segmentation of water and non-water classes. While the exact resolution is not consistently specified across sources, its use in ASV applications suggests a resolution suitable for real-world navigation scenarios. In the Figure 13 example images from the ICVDS Dataset.

This dataset consists of 318 annotated images, which is a relatively small size compared to some other datasets available for similar tasks. The dataset is divided into three subsets:

Figure 8. *Example images from ICVDS dataset*

191 images are allocated for training the models, 87 images are used for validation during the training process, and 40 images are reserved for the final evaluation of the trained models.

2.4.2 Lufi-Riversnap (River Water segmentation)

The LuFI-RiverSnap dataset focuses on close-range river scene images and was presented Moghimi et al. [31]. It includes images obtained from various devices like UAVs, surveillance cameras, smartphones, and handheld cameras, with sizes reported up to 4624 × 3468 pixels. The dataset incorporates social media images to increase diversity in river landscapes. It is designed for river water segmentation and includes pixel-wise segmentation for river water.

Figure 9. *Example images from Lufi-Riversnap dataset*

This dataset offers a more substantial collection of images compared to ICVDS, containing a total of 1092 images, all of which are accurately annotated with ground truth masks. The dataset is further divided into subsets for different stages of model development and evaluation: 657 images are designated for training the segmentation models, 202 images are used for validation to tune model hyperparameters and monitor performance during training, and 233 images are reserved for the final testing of the trained models.

2.4.3 RIWA (River Water Segmentation Dataset)

RIWA is a dataset specifically created for river water segmentation. The dataset was introduced Gorriz et al. [32] in 2023. It provides pixel-wise binary segmentation of river water and includes images sourced from a variety of devices, such as smartphones, drones, and DSLRs, as well as images from the AED20K dataset. The dataset is manually labeled to ensure quality. While images may be processed into smaller patches (e.g., 512x512) for model input in some studies, the original resolutions from diverse sources contribute to the dataset's variability.

Figure 10. *Example images from RIWA dataset*

This version of the RIWA dataset contains a total of 1632 images, divided into 1142 images for training, 167 images for validation, and 323 images for testing the performance of trained models.

2.4.4 Tampere-WaterSeg

The Tampere-WaterSeg dataset has been specifically created by Taipalmaa et al. [33] for the task of water segmentation in images that were captured by an autonomous surface vehicle. This dataset comprises 600 labeled images, all of which are in high definition with a resolution of 1920×1080 pixels. The images were recorded using a GoPro Hero 4 Session camera mounted on a USV operating on Lake Pyhäjärvi in Tampere, Finland, during the wintertime. The dataset is further divided into three subcategories, each containing 200 images: open lake scenarios, channel area navigation, and docking situations.

The Tampere-WaterSeg dataset provides hand-annotated segmentation masks that accurately delineate the water regions within each image. Given that the imagery in this dataset was captured directly from a USV, it is highly relevant for training water segmentation models that will be used in similar autonomous vessel applications. The high resolution of the images allows for the capture of fine details of the water surface and the surrounding environment, which can be beneficial for training accurate segmentation models. However,

Figure 11. *Example images from Tampere-WaterSeg dataset*

it is important to note that the dataset focuses on winter conditions and a single lake environment, which might limit the ability of a model trained solely on this data to generalize effectively to other seasons and different types of water bodies.

2.4.5 USVInland

The USVInland dataset is a multi-sensor dataset which was introduced by Cheng et al. [34] in 2021 that has been specifically created for research and development related to unmanned surface vehicles operating in inland waterways, with water segmentation being identified as one of the key tasks supported by the dataset. The data collection for this dataset spanned over 26 kilometers of diverse real-world scenes in inland waterways. For the specific task of water segmentation, the USVInland dataset includes approximately 1400 images, comprising 364 high-resolution images (1280 × 640 pixels) and 1036 low-resolution images (640 × 320 pixels). The data was recorded using a self-driving boat that was equipped with a comprehensive suite of sensors, including lidar, stereo cameras, millimeter-wave radar, GPS, and inertial measurement units (IMUs).

Figure 12. *Example images from USVInland dataset*

The USVInland dataset provides comprehensive annotations for the entire water area within the images, using polygons to accurately delineate the water boundaries. Given its specific focus on inland waterways and the fact that the imagery was captured from a self-driving

boat, this dataset is exceptionally relevant for training water segmentation models for autonomous vessels that are intended to operate in such environments. While the dataset's multi-sensor nature extends its utility beyond just image-based water segmentation, the availability of detailed water area annotations as polygons makes it directly applicable to the user's needs. The inclusion of images at both high and low resolutions offers flexibility for training models that might need to operate under different computational constraints or for evaluating the impact of image resolution on segmentation performance.

2.4.6 Waternet (WaterDataset)

Waternet, also referred to as WaterDataset, is a dataset that has been designed for water segmentation in both still images and videos, with a particular focus on addressing the challenges posed by the dynamically changing appearance of water. The dataset was introduced by Liang et al. [35] in 2020. The latest version, WaterDataset v2, comprises a substantial collection of 4413 images, of which 2392 contain labeled water objects. Additionally, the dataset includes 20 evaluation videos that can be used for testing the temporal consistency and robustness of water segmentation models. The image data is split into a training set of 2188 images and a validation set of 2225 images. The Waternet dataset incorporates a diverse range of water-related images sourced from the ADE20K dataset and the RiverDataset, encompassing various types of water bodies such as lakes, canals, rivers, oceans, and even floods. It also includes surveillance videos that capture open water scenes and the dynamics of sea waves.

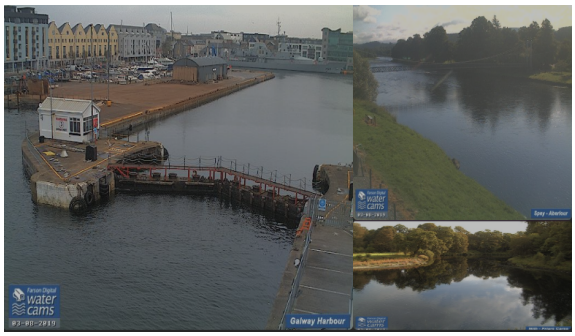

Figure 13. *Example images from WaterDataset dataset*

The annotations in the Waternet dataset provide pixel-level semantic segmentation for water, clearly identifying the water regions within the images. The dataset's large size and the wide variety of water environments represented make it a strong candidate for training water segmentation models that can be applied to autonomous vessels operating in diverse conditions. The inclusion of different water body types, from calm lakes to turbulent oceans, suggests that a model trained on this dataset could potentially achieve a high degree of generalization across various operational environments. The video component

of the dataset might also be valuable for developing models that can leverage temporal information to improve the accuracy and stability of water segmentation in dynamic scenes.

2.4.7 Analysis and Discussion

To facilitate a direct comparison of the ten datasets based on their key features relevant to water segmentation for autonomous vessels, Table 1. summarizes their approximate size in terms of labeled images, image resolution, the type of annotation provided for water regions, the diversity of water scenes captured, and their overall relevance to autonomous vessel applications.

In summary, datasets like USVInland [34] and Tampere-WaterSeg [33], designed with autonomous surface vessels in mind and providing detailed water segmentation masks from a USV perspective, appear to be the most directly suitable for training models for this specific application. Moreover, Tampere-WaterSeg Dataset is more suitable for our latitudes with darker water in the seas and rivers, as well as gray and overcast clouds in the sky. Other datasets focusing on various water environments with pixel-wise segmentation also offer valuable data for developing robust water segmentation capabilities, but more suitable for Unmanned Aerial Vehicle (UAV) or surveillance cameras than for Unmanned Surface Vehicle (USV). The ICVDS dataset could also be suitable for our purposes; however, the resources containing the dataset are no longer accessible, preventing its use in this study.

Table 1. *A table with datasets*

Datasets	**Number of Images**	**Resolutions**	**Water Scene Diversity**
ICVDS	318	Not explicitly stated	Single lake, various times of day
Lufi-Riversnap	1092	Up to 4624 × 3468	Rivers from various locations and sources
RIWA	1632	400 x 400	Rivers from various sources
Tampere-WaterSeg	600	1920 × 1080	Lake in winter, USV perspective
USVInland	700	518 -> 640x320 182 -> 1280x640	Inland waterways, multi-sensor data, USV perspective

Continues...

Table 1 – *Continues...*

Datasets	**Number of Image**	**Resolutions**	**Water Scene Diversity**
WaterDataset	4413	313x472, 788x1144, 480x640, 256x256, 1632x1224	Diverse: lakes, canals, rivers, oceans, floods

2.5 Metrics

For autonomous vessels to operate safely and effectively, the ability to accurately perceive their surroundings is paramount. Given the complexities of aquatic environments, which often involve variations in lighting and reflections, robust evaluation methods are necessary to assess the performance of neural networks designed for this segmentation task. This chapter aims to identify and justify suitable evaluation metrics for a neural network performing water segmentation for USVs at a real-time processing speed of 30 frames per second (FPS). The focus will be on metrics that quantify the quality of the generated segmentation masks and the accuracy of pixel-wise classification.

A fundamental tool for understanding the performance of a semantic segmentation model is the confusion matrix [36] [37] [38]. For the specific case of water segmentation, the confusion matrix categorizes the model's predictions into four types based on their agreement with the ground truth for the water class: True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN). A True Positive occurs when a pixel that is actually water in the ground truth is correctly identified as water by the model. Conversely, a True Negative signifies a non-water pixel that is correctly classified as non-water. Errors made by the model are represented by False Positives and False Negatives. A False Positive happens when a non-water pixel is incorrectly identified as water, directly relating to the frequency with which non-water pixels are misclassified. A False Negative occurs when a water pixel is incorrectly identified as non-water, which is inversely related to how often water pixels are correctly identified. Understanding these components of the confusion matrix is essential for interpreting the various evaluation metrics used in semantic segmentation.

2.5.1 Intersection over Union (IoU) / Jaccard Index

Intersection over Union (IoU), also known as the Jaccard Index, is a widely used metric that quantifies the degree of overlap between the predicted water mask and the ground truth water mask [39]. It is calculated as the ratio of the area of intersection between the two masks to the area of their union. The formula for IoU is given by:

$$IoU = \frac{TP}{(TP + FP + FN)} \tag{2.1}$$

The value of IoU ranges from 0 to 1, where 0 indicates no overlap between the predicted and ground truth masks, and 1 signifies a perfect match. A higher IoU score indicates better segmentation quality, meaning the predicted water region closely aligns with the actual water region in the ground truth. For USV navigation, IoU is a particularly suitable metric as it directly measures the quality of the generated mask, which is crucial for determining the precise navigable areas for the vessel.

2.5.2 Precision

Precision in semantic segmentation measures the proportion of correctly identified water pixels among all the pixels that the model predicted as water [37] [38]. It addresses the question of how often a non-water pixel is incorrectly identified as water. The formula for precision is:

$$Precision = \frac{TP}{(TP + FP)} \tag{2.2}$$

A high precision score, closer to 1, indicates that the model has a low false positive rate. In the context of USV navigation, high precision is essential for safety. If the model frequently misclassifies non-water areas, such as land or obstacles, as navigable water, it could lead the USV into potentially hazardous situations.

2.5.3 Recall (Sensitivity, True Positive Rate)

Recall, also known as sensitivity or the true positive rate, measures the proportion of actual water pixels in the ground truth that were correctly identified by the model as water [38] [36]. It addresses how often a water pixel is correctly identified as water. The formula for recall is:

$$Recall = \frac{TP}{(TP + FN)} \tag{2.3}$$

A high recall score, closer to 1, signifies that the model has a low false negative rate, meaning it correctly identifies most of the actual water pixels present in the scene. For

USV navigation, high recall is important for ensuring that the vessel identifies most of the navigable water area and does not unnecessarily avoid safe regions due to misclassification as non-water.

2.5.4 F1-score (Dice Coefficient)

The F1-score, also known as the Dice Coefficient or Sørensen–Dice index, is the harmonic mean of precision and recall. It provides a single metric that balances the trade-off between precision and recall, which is particularly useful when there is an imbalance in the number of water and non-water pixels in the dataset [37] [40] [41]. The formula for the F1-score is:

$$F1 - score = \frac{2 * (Precision * Recall)}{(Precision + Recall)} = \frac{2 * TP}{(2 * TP + FP + FN)} \tag{2.4}$$

The F1-score ranges from 0 to 1, with 1 indicating perfect precision and recall. For USV navigation, this metric offers a comprehensive assessment of the segmentation performance, reflecting the model's ability to accurately identify navigable water while also minimizing misclassifications of both water and non-water pixels, which is crucial for reliable autonomous operation.

3. Experimental Results

This chapter presents a comparative analysis of different water segmentation models, evaluating their effectiveness under varying configurations. Key characteristics such as precision, recall, processing speed, and size are examined. Through systematic assessment, the analysis identifies optimal configurations that enhance segmentation accuracy while maintaining computational feasibility. The findings provide insights into the trade-offs between model complexity and performance, offering guidance for selecting appropriate approaches in practical applications.

3.1 Parameters of Training

As mentioned previously, two datasets—USVInland and Tampere-WaterSeg—were selected for training the models. After combining the datasets, a total of 1,300 images were obtained and divided into training (60%, 781 images), validation (30%, 390 images), and testing (10%, 129 images) sets. Additionally, 101 images collected in the Gulf of Finland near Tallinn were included in the testing set. This resulted in a total of 230 images used for testing. Figure 14 presents an example image from this dataset.

Figure 14. *Example images from our dataset*

All models were trained with the same parameters and on the same hardware. Table 2 shows all the parameters that were used during the training of the models.

Table 2. *Training Parameters*

Parameters	**Values**
Epochs	50
Learning Rate	0.01
GPU	GeForce RTX 2070
Python	3.11.4
Optimizer	Adam
Input Image Size	512x512 pixels

3.2 The Crucial Role of Backbones in Semantic Segmentation for USVs

Semantic segmentation, the task of classifying each pixel in an image, is fundamental for Unmanned Surface Vehicles (USVs) to understand their environment, particularly for identifying navigable water. Deep learning models like U-Net, SegNet, DeepLabV3+, and BiSeNet are widely used for this purpose. At the core of these models lies the backbone, a pre-trained convolutional neural network (CNN) that serves as the primary feature extractor. Without a robust backbone, these segmentation networks would struggle to interpret the visual information necessary for accurate pixel-level classification.

The backbone's main function is to process the input image through a series of convolutional and pooling layers, progressively extracting hierarchical features. Early layers capture low-level details like edges and textures, while deeper layers learn more complex and abstract representations corresponding to objects and regions. These extracted feature maps are then passed to the rest of the segmentation network, which uses them to predict the class label for each pixel.

3.2.1 Suitable Backbones for Water Segmentation on USVs

Selecting the right backbone for water segmentation on a USV requires considering the specific challenges of this environment, such as varying lighting conditions, reflections, waves, and the need for real-time performance on potentially resource-constrained platforms.

U-Net: For U-Net, backbones that offer a good balance between feature extraction power and computational efficiency are desirable.

- ResNet (e.g., ResNet-18, ResNet-34): ResNet variants are widely used and provide strong feature representations. Smaller versions like ResNet-18 or ResNet-34 can offer a good trade-off between performance and computational cost for USV deployment. Their residual connections help with training deeper networks, potentially improving accuracy in challenging water scenes.
- MobileNetV3: These are lightweight backbones designed for mobile and embedded vision applications. They offer significantly reduced computation and parameters compared to larger models while maintaining reasonable accuracy. This makes them highly suitable for real-time performance on USVs with limited processing power.

SegNet: As SegNet often uses VGG, considering its variants or more efficient alternatives as backbones is relevant for USVs.

- VGG (e.g., VGG-16): While historically common, VGG is computationally more expensive than newer architectures. For USV applications where real-time performance is critical, a lighter backbone would generally be preferred.
- MobileNetV3: Similar to U-Net, MobileNet variants are excellent choices for SegNet on a USV due to their efficiency, enabling faster inference times.

DeepLabV3+: DeepLabV3+ is known for its high accuracy, often at the cost of higher computational requirements. Choosing an efficient backbone is key for USV deployment.

- ResNet (e.g., ResNet-50, ResNet-101): While larger ResNet models (ResNet-101) offer excellent accuracy, using smaller ResNet variants (ResNet-50) and incorporating atrous convolutions within the backbone itself can help capture multi-scale information more efficiently for the ASPP module.
- Xception: Xception's depthwise separable convolutions can be more efficient than standard convolutions. Adapting a smaller Xception model could be an option, though careful consideration of its computational cost on the target USV hardware is necessary.

BiSeNet: This architecture is explicitly designed for real-time performance, making lightweight backbones the natural choice for its Context Path on a USV.

- ResNet (e.g., ResNet-18, ResNet-101): These are highly suitable because of their efficiency and ability to provide the necessary semantic context quickly.

- ShuffleNetV2: Another lightweight architecture designed for mobile devices, offering a good balance between speed and accuracy, making it a strong candidate for BiSeNet's backbone on a USV..

3.3 Model Performance Evaluation

This section presents a comparative evaluation of various semantic segmentation models tailored for real-time water segmentation tasks. Each architecture — U-Net, SegNet, DeepLabV3+, BiSeNet, and WaSR-Net were assessed using a consistent set of metrics, including Intersection over Union (IoU), precision, recall, F1 score, model size, and inference speed (FPS). The goal is to identify configurations that optimally balance segmentation accuracy with computational efficiency for deployment in resource-constrained, real-time environments such as unmanned surface vehicles (USVs). The following subsections detail the performance outcomes for each model family, highlighting their strengths, limitations, and suitability for different application scenarios.

3.3.1 Evaluation Results of U-Net

The U-net variants demonstrate a trade-off between accuracy and computational efficiency. As shown in Table 3, the ResNet34 + U-net configuration achieves the highest IoU (91.98%) and F1 score (95.39%) but has a larger model size (175.82 MB) and slower inference speed (161.68 FPS). In contrast, MobileNet3 + U-net offers the fastest inference (226.36 FPS) and smallest model size (9.48 MB) but sacrifices accuracy (IoU: 81.01%). ResNet18 + U-net balances these metrics moderately. For real-time applications requiring high accuracy, ResNet34 + U-net is optimal.

Table 3. *Evaluation Results of U-Net*

Models/Results	**Model Size**	**IoU**	**Precision**	**Recall**	**F1 Score**	**Average Inference Time**
ResNet18 + U-Net	137.26 MB	85.47%	85.98%	**99.29%**	91.72%	5.14 ms -> 194.41 FPS
ResNet34 + U-Net	175.82 MB	**91.98%**	**92.88%**	98.92%	**95.39%**	6.18 ms -> 161.68 FPS
MobileNetV3 + U-Net	**9.48 MB**	81.01%	83.1%	97.02%	88.61%	4.42 ms -> **226.36 FPS**

3.3.2 Evaluation Results of SegNet

MobileNet3 + SegNet outperforms other SegNet configurations with the highest IoU (94.24%) and F1 score (96.74%), despite its moderate inference speed (121.80 FPS). While VGG + SegNet achieves faster inference (253.64 FPS) and SegNet without backbone has the smallest size (71.79 MB), both exhibit significantly lower accuracy. MobileNet3 + SegNet is the clear choice for SegNet models, balancing superior segmentation performance with acceptable real-time capability. All results are shown in the Table 4.

Table 4. *Evaluation Results of SegNet*

Models/Results	**Model Size**	**IoU**	**Precision**	**Recall**	**F1 Score**	**Average Inference Time**
SegNet without Backbone	**27.89 MB**	77.67%	78.43%	98.69%	86.79%	3.94 ms -> **253.64 FPS**
MobileNetV3 + SegNet	71.79 MB	**94.24%**	**94.98%**	**99.19%**	**96.74%**	8.21 ms -> 121.8 FPS
VGG + SegNet	92.05 MB	88.42%	89.49%	98.35%	93.45%	3.94 ms -> **253.64 FPS**

3.3.3 Evaluation Results of DeepLab3+

Xception + DeepLabV3+ provides the best balance in this category, achieving an IoU of 94.59% and F1 score of 97.06% with fast inference (139.72 FPS). While ResNet50 + DeepLabV3+ marginally outperforms Xception in IoU (95.23%), its lower FPS (126.98) makes Xception more suitable for real-time applications. ResNet101 + DeepLabV3+ offers negligible accuracy gains but suffers from high computational latency (79.88 FPS), rendering it impractical for real-time use.

Table 5. *Evaluation Results of DeepLab3+*

Models/Results	**Model Size**	**IoU**	**Precision**	**Recall**	**F1 Score**	**Average Inference Time**
ResNet50 + DeepLabV3+	160.21 MB	95.23%	96%	99.15%	97.42%	7.88 ms -> 126.98 FPS

Continues...

Table 5 – *Continues...*

Models/Results	**Model Size**	**IoU**	**Precision**	**Recall**	**F1 Score**	**Average Inference Time**
ResNet101 + DeepLabV3+	232.66 MB	**95.28%**	**96.82%**	98.26%	**97.43%**	12.52 ms -> 79.88 FPS
Xception + DeepLabV3+	**143.59 MB**	94.59%	95.01%	**99.53%**	97.06%	7.16 ms -> **139.72 FPS**

3.3.4 Evaluation Results of BiSeNet

BiseNet variants underperform compared to other architectures. ResNet18 + BiseNet achieves the highest IoU (77.53%) and F1 score (86.8%) in this group, with moderate inference speed (168.52 FPS). ShufflenetV2 + BiseNet is lightweight (15.29 MB) but exhibits poor accuracy (IoU: 65.04%). BiseNet configurations are generally less suited for high-precision water segmentation tasks.

Table 6. *Evaluation Results of BiSeNet*

Models/Results	**Model Size**	**IoU**	**Precision**	**Recall**	**F1 Score**	**Average Inference Time**
ResNet18 + BiSeNet	55.49 MB	**77.53%**	**81.43%**	**93.94%**	**86.8%**	5.93 ms -> **168.52 FPS**
ResNet101 + BiSeNet	193.4 MB	73.84%	77.95%	93.31%	84.38%	13.04 ms -> 76.67 FPS
ShuffleNetV2 + BiSeNet	**15.29 MB**	65.04%	70.14%	90.3%	78.12%	8.8 ms -> 113.63 FPS

3.3.5 Analysis of WaSR-Net for Real-Time Water Segmentation

The WaSR model, pre-trained by its original authors, demonstrates superior segmentation performance compared to all other evaluated architectures, achieving an IoU of 96.36%, precision of 97.11%, and an F1 score of 97.86%. These metrics indicate exceptional accuracy in water segmentation, likely due to specialized training on maritime environments.

However, the model's practical deployment for real-time USV applications is severely limited due to two critical drawbacks:

1. Extremely Slow Inference Speed: With an average inference time of 299.15 ms (0.33 FPS), WaSR is far too slow for dynamic environments like open water or rivers, where rapid scene changes demand real-time processing (ideally > 30 FPS for smooth operation)
2. Large Model Size: At 1.12 GB, WaSR is significantly bulkier than alternatives (e.g., MobileNet3 + U-net at 9.48 MB), making it unsuitable for edge devices with limited storage and memory

Implications for USV Applications

While WaSR's accuracy is unmatched, its computational inefficiency renders it impractical for real-time use in unmanned surface vehicles. In maritime environments, where obstacles, waves, and currents require sub-second reaction times, a delay of 300 ms per frame could jeopardize navigation safety. For offline analysis (e.g., post-mission mapping), WaSR could be viable, but lighter, faster models (e.g., Xception + DeepLabV3+ or MobileNet3 + SegNet) remain preferable for live deployment.

3.4 Analysis of Results

For real-time water segmentation in USVs, model selection must prioritize a balance between inference speed, accuracy, and deployability. Based on the analysis, the following configurations are recommended:

Best Configurations per Model Type:

- WaSR-Net: Pretrained model (Highest accuracy: IoU 97.11%, FPS 0.33)
- U-net: ResNet34 + U-net (High accuracy with optimal speed: IoU 91.98%, FPS 161.68)
- SegNet: MobileNet3 + SegNet (Optimal accuracy-speed balance: IoU 94.24%, FPS 121.80)
- DeepLabV3+: Xception + DeepLabV3+ (Best overall performance: IoU 94.59%, FPS 139.72)
- BiseNet: ResNet18 + BiseNet (Best in category: IoU 77.53%, FPS 168.52)

Overall Ranking (Performance vs. Speed):

1. Xception + DeepLabV3+ (High accuracy, robust speed)
2. MobileNet3 + SegNet (Exceptional accuracy with moderate speed)
3. ResNet34 + U-net (High accuracy, moderate speed)

4. VGG16 + SegNet (Fastest inference: 253.64 FPS, moderate accuracy)
5. WaSR-Net (Highest accuracy)

For edge deployment with strict size constraints, MobileNet3 + U-net (9.48 MB) and MobileNet3 + SegNet (27.89 MB) are ideal. For high-accuracy scenarios, Xception + DeepLabV3+ and MobileNet3 + SegNet are superior. BiseNet architectures are not recommended for precision-critical tasks.

3.5 Analysis of the Shortcomings of Trained Models

All four selected models demonstrate strong performance based on their average evaluation metrics. However, analysis of individual predicted masks reveals limitations in accurately detecting boundaries between water and non-water regions. In some cases, the models also fail to correctly classify highlights or reflections on water surfaces, occasionally misidentifying them as non-water areas.

The Figure 15 shows three examples of defective masks generated by the DeepLab3+ model. The top image shows that the model detects a concrete pier as part of the water, and quite a large part of this pier. The middle image illustrates a case where certain regions of the sky are incorrectly detected as water. In the last image, portions of the vessel are misclassified as water.

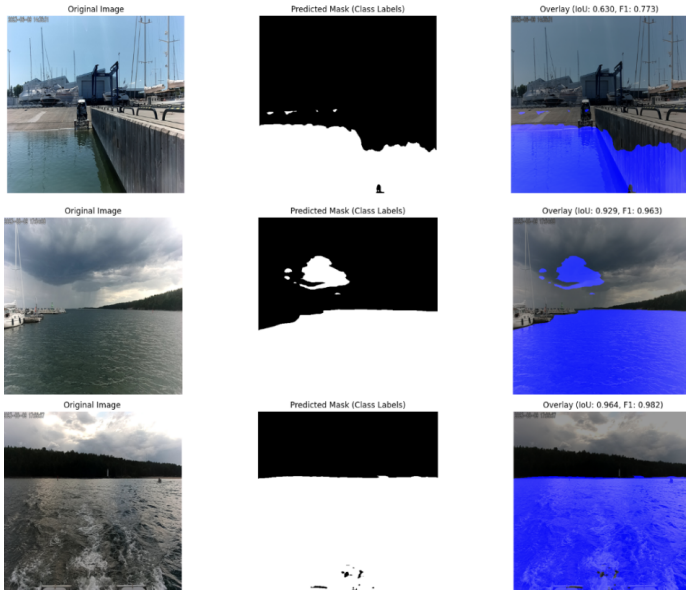

Figure 15. *Unsuccessful masks generation by Xception + DeepLab3+ model*

Sky defects are found in all models except WaSR-Net, but most often in SegNet + VGG16, where the model distinguishes water boundaries quite well, but its accuracy is spoiled by its frequent recognition of clouds or a clear sky as water. Examples of these defective

generations can be seen in the Figure 16.

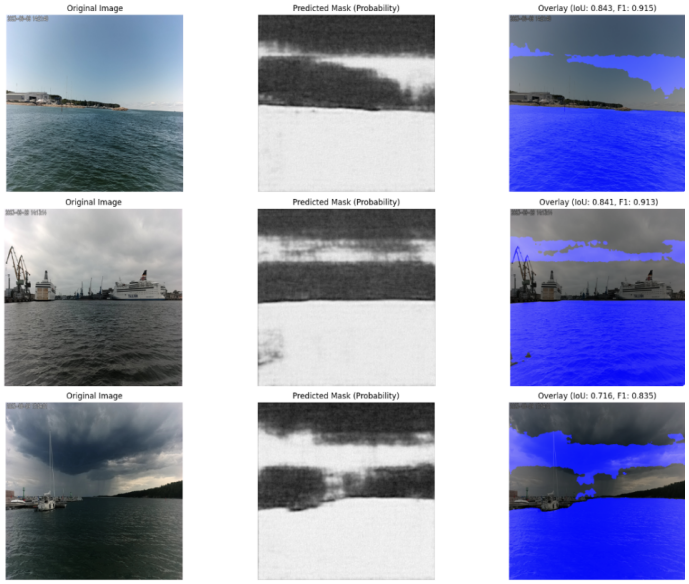

Figure 16. *Unsuccessful masks generation by VGG16 + SegNet model*

U-Net + ResNet34 can sometimes distinguish water as part of land or in Figure 17 in the topmost example you can see that it distinguishes most of the boat as water and behind the boat water as not a boat. The same problem as DeepLabV3+ can't distinguish part of the ship as not water in the top and bottommost example.

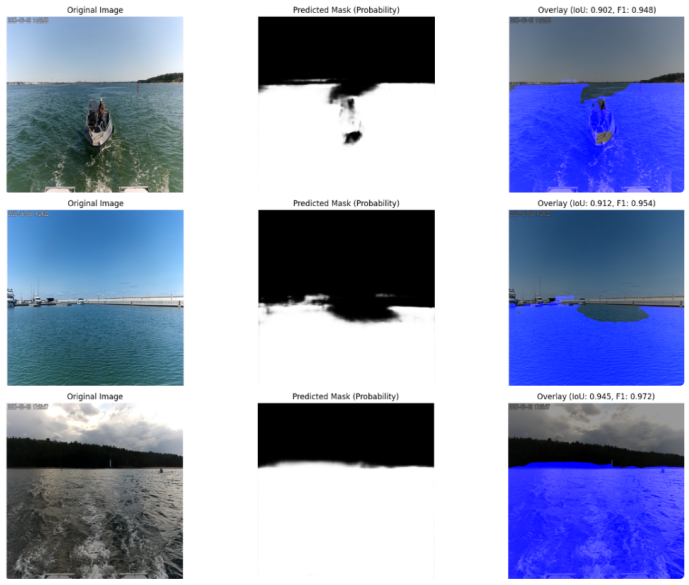

Figure 17. *Unsuccessful masks generation by ResNet34 + U-Net model*

SegNet in the MobileNetV3 configuration is less susceptible to generation defects compared to VGG16, but approximately at the same level as Xception + DeepLabV3+. You can see examples of unsuccessful generations in Figure 18.

It can be noted that the above-mentioned mask generation errors are not found in WaSR-

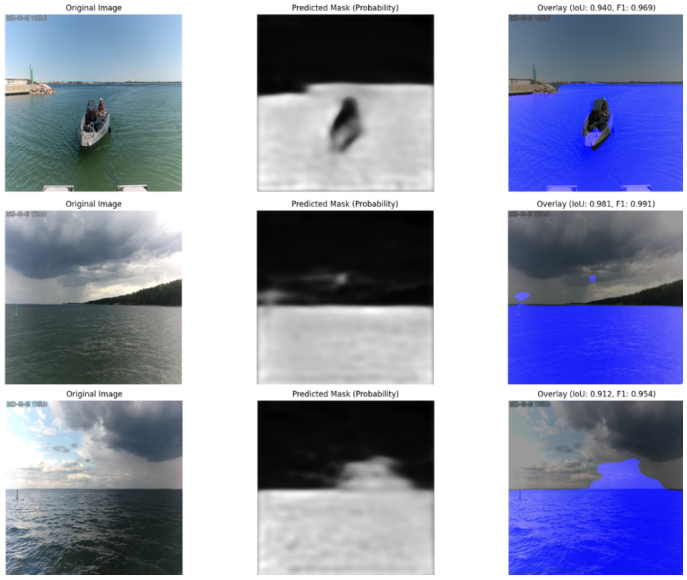

Figure 18. *Unsuccessful masks generation by MobileNetV3 + SegNet model*

Net on such a scale; for example, WaSR-Net has no particular problems with recognizing the sky, since the model is trained to find 3 classes: sky, land and water, and there are also no problems with finding such small objects as buoys. However, the model does not always recognize, for example, a concrete pier as part of the land, as shown in Figure 19.

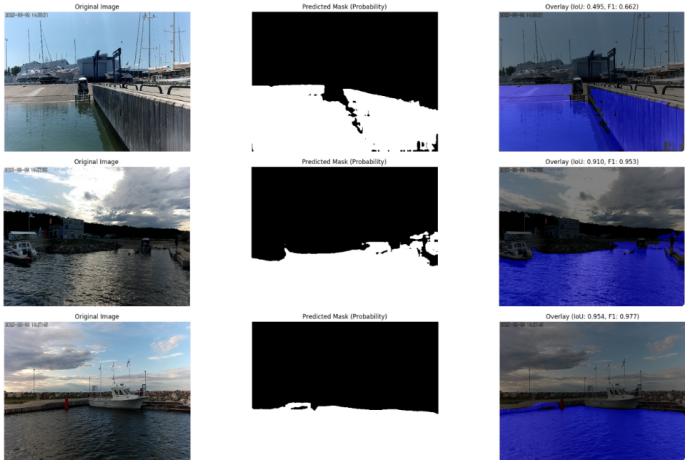

Figure 19. *Unsuccessful masks generation by WaSR-Net model*

3.6 Improving the Robustness of Models

The results from the previous section indicate that the trained models lack sufficient local context to accurately recognize obstacles. USVInland and Tampere-WaterSeg Datasets contain pictures of river views and the viewing angle is directed at the water rather than into the distance, which is why the sky is mostly not visible and the models have no idea what it is. Also, these Datasets do not contain objects in the middle of the water such as buoys or other vessels, which confuses the models when they see some foreign objects in

the water in our pictures and either do not recognize them at all, or define them as part of existing land and, for example, define the water behind them as not water either.

To improve the accuracy of the models, it was decided to add even more labels to the existing pictures from the Gulf of Finland and add them to Validation and retrain the models with the expanded Dataset. The training parameters remained consistent with the initial configuration; however, 129 images previously allocated to the test set were reassigned to the training set, leaving only 101 images from the Gulf of Finland for testing. Additionally, 125 newly annotated images from the Gulf of Finland were added to the validation set.

As a result, this is what our Dataset looks like:

- Train: 910 images
- Validation: 515 images
- Test: 101 images
- Total: 1526 images

Improvements in all models from training them on the new Dataset did not bring any significant changes. Table 7 shows the new results for the models and next to them in brackets the difference with the old values. The size of all models remained the same, so it was decided not to include this parameter in the new table. These figures indicate that the new images constitute a relatively small portion of the validation set—approximately 24% — and therefore have a minimal impact on the training process.

Table 7. *Evaluation Results with New Labeled Data*

Models/Results	**IoU**	**Precision**	**Recall**	**F1 Score**	**Average Inference Time**
ResNet34 + U-Net	92.84% (+0.86%)	93.11% (+0.23%)	**99.65%** (+0.73%)	95.92% (+0.53%)	6.52 ms (+0.34) -> (Equivalent FPS: 153.41)
Xception + DeepLabV3+	**95.16%** (+0.57%)	95.8% (+0.79%)	99.29% (-0.24%)	**97.39%** (+0.33%)	7.16 ms (+0.00) -> (Equivalent FPS: 139.72)
MobileNetV3 + SegNet	95.1% (+0.86%)	**95.95%** (+0.97%)	99.01% (-0.18%)	97.3% (+0.56%)	7.94 ms (-0.27) -> (Equivalent FPS: 125.89)

Continues...

Table 7 – *Continues...*

Models/Results	**IoU**	**Precision**	**Recall**	**F1 Score**	**Average Inference Time**
VGG16 + SegNet	89.4% (+0.98%)	90.18% (+0.69%)	98.86% (+0.51%)	94.1% (+0.65%)	**3.93 ms** (-0.01) -> (Equivalent FPS: 254.46)

The next step is to try to increase the percentage of influence of our data on validation by reducing the amount of data from validation. For the next training sessions, 295 images were transferred from the validation set to the training set, increasing the percentage of images in the validation set to 57%.

The new dataset now looks like this:

- Train: 1205 images
- Validation: 220 images
- Test: 101 images
- Total: 1526 images

Table 8 shows the performance of models with balanced Validation images towards the Gulf of Finland images. The difference with the original data is given in brackets next to the values. U-Net with ResNet34 shows a significant increase compared to the original parameters. For example, the IoU value increased by 3.27% from 91.98% to 95.25%, and the average increase was about 2% with the original results at the cost of 10 FPS. The image segmentation quality of DeepLabV3+ with Xception exceeded 97% for all indicators and at the same time slightly increased the processing time to 1 FPS. With SegNet, things are worse, since for example with MobileNetV3 the IoU increase is less than 1 percent - 0.78% but the mask generation speed decreased by 6 FPS, which rather harmed the model than improved it. SegNet with VGG16 showed the most dramatic results - IoU dropped by 9.85% from the initial values from 88.42% to 78.57%.

Table 8. *Evaluation Results with Balancing Dataset*

Models/Results	**IoU**	**Precision**	**Recall**	**F1 Score**	**Average Inference Time**
ResNet34 + U-Net	95.25% (+3.27%)	95.54% (+2.66%)	**99.67%** (+0.75%)	97.47% (+2.08%)	6.62 ms (+0.44) -> (Equivalent FPS: 151)

Continues...

Table 8 – *Continues...*

Models/Results	**IoU**	**Precision**	**Recall**	**F1 Score**	**Average Inference Time**
Xception + DeepLabV3+	**97.25%** (+2.66%)	**97.66%** (+2.65%)	99.56% (+0.03%)	**98.58%** (+1.52%)	7.21 ms (+0.05) -> (Equivalent FPS: 138.72)
MobileNetV3 + SegNet	95.02% (+0.78%)	95.75% (+0.77%)	99.18% (-0.01%)	97.32% (+0.58%)	8.63 ms (+0.42) -> (Equivalent FPS: 115.93)
VGG16 + SegNet	78.57% (-9.85%)	80.76% (-8.73%)	96.98% (-1.37%)	87.56% (-5.89%)	**4.19 ms** (+0.25) -> (Equivalent FPS: 238.59)

From Figure 20, it is clear that the generated mask of SegNet with VGG16 looks more like noise, where the probability of water is higher at the bottom and lower at the top.

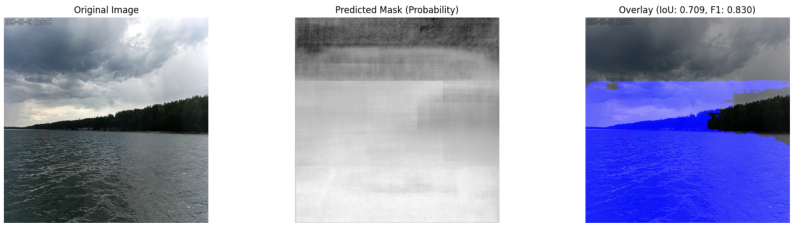

Figure 20. *Example of a generated mask using retrained SegNet + VGG16*

3.7 Analysis and Discussion

In conclusion, based on the metrics presented in Table 8, the DeepLabV3+ model with the Xception backbone demonstrates the best performance across most indicators. As a result, this model was selected for further use.

For the final decision, the results will also be compared with those of WaSR-Net. Table 9 shows that DeepLab3V+ is ahead of WaSR-Net in all parameters. Which makes it an ideal model for water segmentation in the conditions of the sea coast of the Gulf of Finland.

Table 9. *Comparisons of Xception + DeepLabV3+ with WaSR-Net*

Models/Results	**Model Size**	**IoU**	**Precision**	**Recall**	**F1 Score**	**Average Inference Time**
Xception + DeepLabV3+	143.59 MB	97.25%	97.66%	99.56%	98.58%	7.21 ms -> 138.72 FPS
WaSR-Net	1.12 GB	96.36%	97.11%	99.14%	97.86%	299.15 ms -> 0.33 FPS

However, upon examining specific examples of mask generation from different models, as shown in Figure 21, it becomes evident that WaSR-Net generally outperforms DeepLabV3+ by 1-2%. Nevertheless, in the cases shown at the bottom of the figure, WaSR-Net experiences a significant drop in prediction accuracy due to the concrete pier, while DeepLabV3+ also demonstrates some decline, though to a lesser extent. These images suggest that WaSR-Net's performance may be adversely affected by such features, contributing to its lower average accuracy.

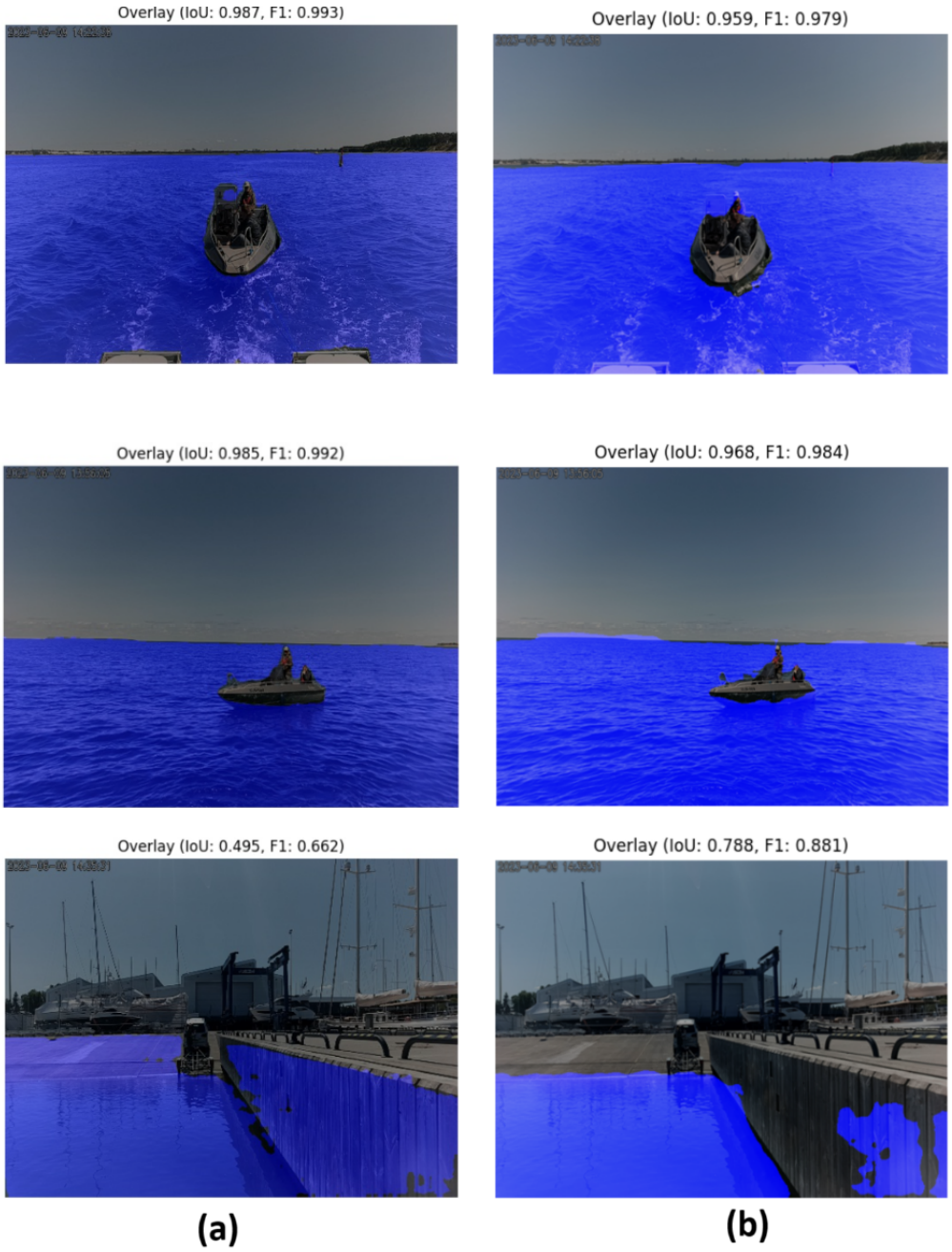

Figure 21. *Comparisons of mask generation on specific examples **a)** WaSR-Net **b)** Xception + DeepLabV3+*

4. Conclusion

In conclusion, this study has identified several key findings regarding object segmentation in images, specifically focusing on the separation of water from non-water elements.

Firstly, neural networks demonstrated significant flexibility and adaptability in this task compared to traditional image segmentation methods in computer vision. This makes them highly suitable for real-time image processing applications. When properly configured, neural network architectures can achieve high performance. In our case, the DeepLabV3+ model with an Xception backbone produced strong results, achieving 97.25% IoU and 98.58% F1 score, while maintaining a processing speed of 138.72 FPS. The combined dataset of USVInland and Tampere-WaterSeg, which included 1300 images covering various water and weather conditions, contributed to robust training outcomes.

Secondly, there exists a vast array of models and configurations within the realm of neural networks suitable for image segmentation tasks. Through a literature review, seven models were identified: FCN, U-Net, SegNet, DeepLabV3+, BiSeNet, Habaek, and WaSR-Net. Of these, five were selected for further testing based on their relevance and feasibility. FCN was excluded due to its outdated architecture and subpar performance in previous studies. The Habaek model was also omitted because of insufficient documentation and lack of publicly available implementation, making it unsuitable for this research.

Thirdly, for each selected model, except for WaSR-Net, numerous variants exist depending on the chosen backbone—the core feature extraction component responsible for processing input data and learning useful representations. In this study, three backbone configurations were selected for each of the four models (U-Net, SegNet, DeepLabV3+, and BiSeNet). For SegNet, one configuration without a backbone was also included for comparison. As mentioned previously, the DeepLabV3+ model with the Xception backbone emerged as the best-performing configuration.

Lastly, the WaSR-Net model, which was specifically developed for water segmentation, achieved an IoU of 96.36% and an F1 score of 97.86% on the dataset prepared for this project. While these results are slightly lower in terms of absolute metrics compared to the DeepLabV3+ configuration, a qualitative analysis of processed frames revealed that WaSR-Net often performs slightly better in practical scenarios. However, in specific cases, such as when vessels are docked, the model mistakenly classifies the pier as water, which

negatively impacts its overall accuracy.

Based on the text written above, the following conclusions can be drawn. DeepLabV3+, with its Xception backbone, struggles to resolve fine details, often classifying them as water or non-water based on spatial location rather than color cues. However, its high performance and compact model size make it well-suited for resource-constrained embedded systems and unmanned surface vehicles (USVs). Conversely, WaSR-Net offers superior accuracy in obstacle detection and environmental analysis but requires substantial computational resources, making it more appropriate for stationary surveillance systems or large vessels with robust processing capabilities. It is safe to say that DeepLabV3+ with Exception is well suited for the water detection conditions in the Baltic Sea.

The objective of segmenting water from non-water regions was successfully achieved using the DeepLabV3+ model with the Xception backbone, demonstrating high effectiveness under the specific environmental conditions of the Baltic Sea.

To enhance DeepLabV3+ for water segmentation, two strategies are particularly effective. First, extending the training dataset with diverse water-related images—encompassing varied lighting conditions, water textures, and small obstacles—can improve generalization and fine-detail recognition, addressing current limitations. Second, replacing the Xception backbone with advanced architectures like EfficientNet or ResNeXt can further optimize performance. EfficientNet provides a scalable balance of accuracy and computational efficiency, while ResNeXt enhances feature extraction through its cardinality-based design. These modifications, combined with DeepLabV3+'s extensive configuration options, make it a highly adaptable and effective solution for water segmentation in resource-limited applications.

Following an in-depth analysis and extensive experimentation, it can be confidently stated that neural network-based methods are highly effective in distinguishing between bodies of water and land areas. Among the models evaluated, DeepLabV3+ with the Xception backbone demonstrated the best performance, positioning it as a strong candidate for integration into the onboard navigation system of a vessel.

References

[1] Jussi Taipalmaa et al. *High-Resolution Water Segmentation for Autonomous Unmanned Surface Vehicles: a Novel Dataset and Evaluation*. 2019. URL: https://api.semanticscholar.org/CorpusID:208089717.

[2] Muhammad Ammar Mohd Adam et al. "Deep Learning-Based Water Segmentation for Autonomous Surface Vessel". In: *IOP Conference Series: Earth and Environmental Science* 540 (Aug. 2020), p. 012055. DOI: 10.1088/1755-1315/540/1/012055.

[3] Xin Han et al. "Water Segmentation for Unmanned Ship Navigation Based on Multi-Scale Feature Fusion". In: *Applied Sciences* 15.5 (2025). ISSN: 2076-3417. DOI: 10.3390/app15052362. URL: https://www.mdpi.com/2076-3417/15/5/2362.

[4] Wei Han, Binyu Zhao, and Jun Luo. "Towards Smaller and Stronger: An Edge-Aware Lightweight Segmentation Approach for Unmanned Surface Vehicles in Water Scenarios". In: *Sensors* 23.10 (2023). ISSN: 1424-8220. DOI: 10.3390/s23104789. URL: https://www.mdpi.com/1424-8220/23/10/4789.

[5] Swati Gautam and Singhai Jyoti. "Critical review on deep learning methodologies employed for water-body segmentation through remote sensing images". In: *Multimedia Tools and Applications* 83 (May 2023), pp. 1–21. DOI: 10.1007/s11042-023-15764-5.

[6] Kai Li, Juanle Wang, and Jinyi Yao. "Effectiveness of machine learning methods for water segmentation with ROI as the label: A case study of the Tuul River in Mongolia". In: *International Journal of Applied Earth Observation and Geoinformation* 103 (Dec. 2021), p. 102497. DOI: 10.1016/j.jag.2021.102497.

[7] IBM. *What is image segmentation?* [Accessed: 02-07-2025]. URL: https://www.ibm.com/think/topics/image-segmentation.

[8] Encord. *Guide to Image Segmentation in Computer Vision: Best Practices*. [Accessed: 02-07-2025]. URL: https://encord.com/blog/image-segmentation-for-computer-vision-best-practice-guide/.

[9] Jonathan Long, Evan Shelhamer, and Trevor Darrell. *Fully Convolutional Networks for Semantic Segmentation*. 2015. arXiv: 1411.4038 [cs.CV]. URL: https://arxiv.org/abs/1411.4038.

[10] Akanksha Chokshi. *Supporting Fully Convolutional Networks (and U-Net) for Image Segmentation*. [Accessed: 26-04-2025]. URL: https://www.datature.io/blog/supporting-fully-convolutional-networks-and-u-net-for-image-segmentation.

[11] Sahitya Arya. *Role of Fully Convolutional Networks in Semantic Segmentation*. [Accessed: 26-04-2025]. URL: https://www.analyticsvidhya.com/blog/2024/07/fully-convolutional-networks-for-semantic-segmentation/.

[12] Laura Lopez-Fuentes, Claudio Rossi, and Harald Skinnemoen. "River segmentation for flood monitoring". In: *2017 IEEE International Conference on Big Data (Big Data)*. 2017, pp. 3746–3749. DOI: 10.1109/BigData.2017.8258373.

[13] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. *U-Net: Convolutional Networks for Biomedical Image Segmentation*. 2015. arXiv: 1505.04597 [cs.CV]. URL: https://arxiv.org/abs/1505.04597.

[14] Lunhao Duan and Xiangyun Hu. "Multiscale Refinement Network for Water-Body Segmentation in High-Resolution Satellite Imagery". In: *IEEE Geoscience and Remote Sensing Letters* 17.4 (2020), pp. 686–690. DOI: 10.1109/LGRS.2019.2926412.

[15] Rémy Vandaele, Sarah Dance, and Varun Ojha. *Automated Water Segmentation and River Level Detection on Camera Images Using Transfer Learning*. Mar. 2021. DOI: 10.1007/978-3-030-71278-5_17.

[16] Lorenzo Steccanella et al. *Deep Learning Waterline Detection for Low-cost Autonomous Boats*. June 2018.

[17] M. Athallah Alhady et al. *Comparative Performance of Water Index for Water Segmentation Model Using U-Net Architecture*. Oct. 2024. DOI: 10.1109/IC3INA64086.2024.10732848.

[18] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. "SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39.12 (2017), pp. 2481–2495. DOI: 10.1109/TPAMI.2016.2644615.

[19] T. S. Akiyama et al. "DEEP LEARNING APPLIED TO WATER SEGMENTATION". In: *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* XLIII-B2-2020 (2020), pp. 1189–1193. DOI: 10.5194/isprs-archives-XLIII-B2-2020-1189-2020. URL: https://isprs-archives.copernicus.org/articles/XLIII-B2-2020/1189/2020/.

[20] Liang-Chieh Chen et al. *Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation*. 2018. arXiv: 1802.02611 [cs.CV]. URL: https://arxiv.org/abs/1802.02611.

[21] Nur Atirah Muhadi et al. "Deep Learning Semantic Segmentation for Water Level Estimation Using Surveillance Camera". In: *Applied Sciences* 11.20 (2021). ISSN: 2076-3417. DOI: 10.3390/app11209691. URL: https://www.mdpi.com/2076-3417/11/20/9691.

[22] Haolin Xue et al. "Deep Learning-Based Maritime Environment Segmentation for Unmanned Surface Vehicles Using Superpixel Algorithms". In: *Journal of Marine Science and Engineering* 9.12 (2021). ISSN: 2077-1312. DOI: 10.3390/jmse9121329. URL: https://www.mdpi.com/2077-1312/9/12/1329.

[23] Changqian Yu et al. *BiSeNet: Bilateral Segmentation Network for Real-time Semantic Segmentation*. 2018. arXiv: 1808.00897 [cs.CV]. URL: https://arxiv.org/abs/1808.00897.

[24] W.B. Zhang, C.Y. Wu, and Z.S. Bao. "SA-BiSeNet: Swap attention bilateral segmentation network for real-time inland waterways segmentation". In: *IET Image Processing* 17 (Sept. 2022), n/a–n/a. DOI: 10.1049/ipr2.12625.

[25] Mingyuan Fan et al. *Rethinking BiSeNet For Real-time Semantic Segmentation*. 2021. arXiv: 2104.13188 [cs.CV]. URL: https://arxiv.org/abs/2104.13188.

[26] Shiyu Xie and Lishan Jia. *GFANet: An Efficient and Accurate Water Segmentation Network*. 2025. DOI: 10.3390/electronics14091890. URL: https://www.mdpi.com/2079-9292/14/9/1890.

[27] Hanseon Joo, Eunji Lee, and Minjong Cheon. *Habaek: High-performance water segmentation through dataset expansion and inductive bias optimization*. 2024. arXiv: 2410.15794 [cs.CV]. URL: https://arxiv.org/abs/2410.15794.

[28] Borja Bovcon and Matej Kristan. *A water-obstacle separation and refinement network for unmanned surface vehicles*. 2020. arXiv: 2001.01921 [cs.CV]. URL: https://arxiv.org/abs/2001.01921.

[29] Matija Teršek, Lojze Žust, and Matej Kristan. *eWaSR – an embedded-compute-ready maritime obstacle detection network*. 2023. arXiv: 2304.11249 [cs.CV]. URL: https://arxiv.org/abs/2304.11249.

[30] Lorenzo Steccanella et al. *Deep Learning Waterline Detection for Low-Cost Autonomous Boats: Proceedings of the 15th International Conference IAS-15*. Jan. 2019. DOI: 10.1007/978-3-030-01370-7_48.

[31] Armin Moghimi et al. “A Comparative Performance Analysis of Popular Deep Learning Models and Segment Anything Model (SAM) for River Water Segmentation in Close-Range Remote Sensing Imagery”. In: *IEEE Access* 12 (2024), pp. 52067–52085. DOI: 10.1109/ACCESS.2024.3385425.

[32] Xabier Blanch Gorriz, Franz Wagner, and Anette Eltner. *River Water Segmentation Dataset (RIWA)*. Jan. 2023. DOI: 10.34740/kaggle/dsv/4901781.

[33] Jussi Taipalmaa et al. *High-Resolution Water Segmentation for Autonomous Unmanned Surface Vehicles: a Novel Dataset and Evaluation*. Oct. 2019. DOI: 10.1109/MLSP.2019.8918694.

[34] Yuwei Cheng et al. *Are We Ready for Unmanned Surface Vehicles in Inland Waterways? The USVInland Multisensor Dataset and Benchmark*. 2021. arXiv: 2103.05383 [cs.RO]. URL: https://arxiv.org/abs/2103.05383.

[35] Yongqing Liang et al. “WaterNet: An adaptive matching pipeline for segmenting water with volatile appearance”. In: *Computational Visual Media* 6 (Mar. 2020). DOI: 10.1007/s41095-020-0156-x.

[36] Jeremy Jordan. *Evaluating image segmentation models.* [Accessed: 02-05-2025]. URL: https://www.jeremyjordan.me/evaluating-image-segmentation-models/.

[37] Nghi Huynh. *Understanding Evaluation Metrics in Medical Image Segmentation*. [Accessed: 02-05-2025]. URL: https://www.kaggle.com/code/nghihuynh/understanding-evaluation-metrics-in-segmentation/notebook.

[38] Yassine Alouini. *All the segmentation metrics!* [Accessed: 02-05-2025]. URL: https://www.kaggle.com/code/yassinealouini/all-the-segmentation-metrics.

[39] Elhassan Mohamed, Konstantinos Sirlantzis, and Gareth Howells. *Incorporation Of Rejection Criterion - A Novel Technique For Evaluating Semantic Segmentation Systems*. 2021. DOI: 10.1109/HSI52170.2021.9538787.

[40] Ivan Popov. *A Deep Dive Into Semantic Segmentation Evaluation Metrics*. [Accessed: 02-05-2025]. URL: https://hackernoon.com/a-deep-dive-into-semantic-segmentation-evaluation-metrics.

[41] Dominik Müller, Iñaki Soto-Rey, and Frank Kramer. *Towards a Guideline for Evaluation Metrics in Medical Image Segmentation*. 2022. arXiv: 2202.05273 [eess.IV]. URL: https://arxiv.org/abs/2202.05273.

Appendix 1 – Non-Exclusive License for Reproduction and Publication of a Graduation Thesis[1]

I Nikita Budovey

1. Grant Tallinn University of Technology free licence (non-exclusive licence) for my thesis "Investigation of Water Segmentation Approaches for Autonomous Vessels", supervised by Uljana Reinsalu and Karl Janson
   1.1. to be reproduced for the purposes of preservation and electronic publication of the graduation thesis, incl. to be entered in the digital collection of the library of Tallinn University of Technology until expiry of the term of copyright;
   1.2. to be published via the web of Tallinn University of Technology, incl. to be entered in the digital collection of the library of Tallinn University of Technology until expiry of the term of copyright.
2. I am aware that the author also retains the rights specified in clause 1 of the non-exclusive licence.
3. I confirm that granting the non-exclusive licence does not infringe other persons' intellectual property rights, the rights arising from the Personal Data Protection Act or rights arising from other legislation.

20.05.2025

[1]The non-exclusive licence is not valid during the validity of access restriction indicated in the student's application for restriction on access to the graduation thesis that has been signed by the school's dean, except in case of the university's right to reproduce the thesis for preservation purposes only. If a graduation thesis is based on the joint creative activity of two or more persons and the co-author(s) has/have not granted, by the set deadline, the student defending his/her graduation thesis consent to reproduce and publish the graduation thesis in compliance with clauses 1.1 and 1.2 of the non-exclusive licence, the non-exclusive license shall not be valid for the period.

Appendix 2 - Link to GitHub with the Project

https://github.com/Parsifal22/Segmentation-models