WaSR – A Water Segmentation and Refinement Maritime Obstacle Detection Network

Borja Bovcon[1] and Matej Kristan[1]

***Abstract*—Obstacle detection using semantic segmentation has become an established approach in autonomous vehicles. However, existing segmentation methods, primarily developed for ground vehicles, are inadequate in an aquatic environment as they produce many false positive (FP) detections in the presence of water reflections and wakes. We propose a novel deep encoder-decoder architecture, a water segmentation and refinement (WaSR) network, specifically designed for the marine environment to address these issues. A deep encoder based on ResNet101 with atrous convolutions enables the extraction of rich visual features, while a novel decoder gradually fuses them with inertial information from the inertial measurement unit (IMU). The inertial information greatly improves the segmentation accuracy of the water component in the presence of visual ambiguities, such as fog on the horizon. Furthermore, a novel loss function for semantic separation is proposed to enforce the separation of different semantic components to increase the robustness of the segmentation. We investigate different loss variants and observe a significant reduction in false positives and an increase in true positives (TP). Experimental results show that WaSR outperforms the current state-of-the-art by approximately 4% in F1-score on a challenging USV dataset. WaSR shows remarkable generalization capabilities and outperforms the state of the art by over 24% in F1 score on a strict domain generalization experiment.**

***Index Terms*—obstacle detection, unmanned surface vehicles, deep learning, semantic segmentation, separation loss**

I. Introduction

ADVANCES in field robotics and mobile sensors have led to the development of unmanned surface vehicles (USVs). These autonomous vehicles are typically small, portable (up to two meters), and equipped with low-power sensors for missions with long endurance. These characteristics make them ideal for use in confined harbours and inspection of hard-to-reach hazardous areas (e.g., near dams). Safe and uninterrupted navigation during missions requires a high degree of autonomy. Thus, the USV must be able to sense the immediate environment and avoid collisions with obstacles in time. For this task, larger boats typically use heavyweight and expensive range sensors (e.g. RADAR [1], LIDAR [2], SONAR [3]), while smaller boats typically rely on more affordable alternatives - cameras combined with computer vision algorithms [4], [5], [6], [7], [8].

A typical USV environment is dynamic and constantly changing, which makes classical background subtraction methods [7] inappropriate due to a high false positive detection rate. Stereo-based reconstruction methods [9], [8] better address the dynamic environment, but require a sufficiently textured scene and a sufficiently large stereo baseline. Calm, poorly textured water leads to inaccurate ground plane estimation and detection failure. Hand-crafted semantic segmentation methods based on fitting structured models to the image [4], [10], [6] have achieved excellent obstacle detection. However, these approaches rely on simple features that cannot fully capture the diversity of the scene appearance. Segmentation quality deteriorates especially in the presence of visual ambiguities and reflections [10].

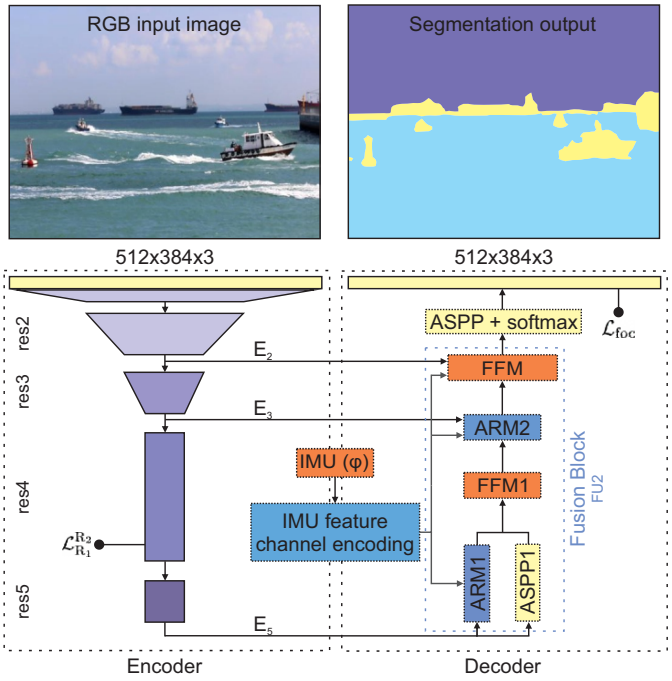


Fig. 1: Architecture of the proposed WaSR network. The encoder generates rich deep features that are gradually fused in the decoder with a IMU feature channel to improve water edge detection and estimation. A semantic separation loss $\mathcal{L}_{\mathrm{R}_1}^{\mathrm{R}_2}$ computed at the end of the encoder drives discriminative feature learning, further reducing false positives and increasing true positives.

Rich features can be learned by deep convolutional neural networks, and indeed developments in autonomous ground vehicles (AGVs) [11], [12], [13], [14] have shown remarkable semantic segmentation results. Recent studies [15], [16] show that these networks cannot be readily applied to USVs due to many differences between the AGV and USV domains. The most obvious difference is that the traversable surface in a maritime domain (i.e., water) is not flat, is dynamic, varies sig-

[1]Borja Bovcon and Matej Kristan are with University of Ljubljana, Faculty of Computer and Information Science, Slovenia
{borja.bovcon, matej.kristan}@fri.uni-lj.si

nificantly in appearance, and is strongly influenced by weather conditions. Studies show that the networks tend to produce many false positives and often miss small obstacles, which is both ineffective and dangerous for real-world applications.

To address the discussed perception challenges specific to the USV environment, we propose a novel encoder-decoder architecture, WaSR (Figure 1). The network fuses encoded inertial data from IMU with visual information at multiple levels of the decoder to improve the accuracy of water segmentation and obstacle detection in the presence of visual ambiguities such as haze on the horizon. Moreover, a new loss computed on the deep encoder features enforces an efficient and compact feature coding of the different water phenomena and separates them from the features corresponding to obstacles. A preliminary version of the network was published in [17].

We claim the following three contributions. The first contribution is a new encoder-decoder architecture with multiple fusion modules in the decoder, which account for a wide range of water appearances and reduce the amount of FP detections. The fusion modules are responsible for combining the inertial information encoded as Euler angles with the RGB image data. Therefore, in our second contribution we propose three techniques to encode the inertial measurements as feature channels adapted for the fusion modules. Our third contribution is a new loss function that clusters features of a given semantic component into a compact cluster and separates them from the features of the remaining semantic components. Clustering multiple semantic components improves sensitivity (clustering water features) or specificity (clustering obstacle features). Extensive experimental results on the most challenging datasets currently available [10], [18] show that WaSR significantly outperforms all state-of-the-art methods in obstacle detection and water edge estimation, and generalizes better to new environments and hardware not used in training.

This paper goes beyond our preliminary work [17] by proposing a new fusion block that significantly reduces the false positive rate. We propose and analyze several IMU encoding techniques and propose a more advanced semantic separation loss that further reduces the false positive rate and improves the generalization ability of the network. We provide a more detailed analysis and comparison with the state of the art and perform a new network generalization analysis.

The remainder of this work is organised as follows. Section II reviews and discusses the most closely related work. An overview of the entire obstacle detection pipeline is given in Section III. Datasets and the evaluation protocol are presented in Section IV, along with a detailed experimental evaluation and ablation study. Section V gives an overview of the main results and indicates possible future research directions.

II. Related work

Over the past decade, cameras have emerged as powerful and affordable obstacle detection devices [19], [4], [5], [6], [7], [8], [20], [21]. Although autonomous marine robotics is still a relatively young research field, numerous hand-crafted image processing methods have already been proposed for obstacle detection. The evaluation of background subtraction methods on a marine domain [7], has shown that the misleading dynamics of water cause a large amount of false positive detections. Moreover, classical background subtraction methods are not suitable for USVs because the undulating sea constantly rocks the USV, changing its viewing angle and violating the assumption of a static camera.

Stereo reconstruction techniques such as [22] and [23] can detect obstacles that protrude clearly through the water surface. A problem arises in calm seas where the water has no texture and a 3D reconstruction of the water surface is severely compromised, leading to false detections. Muhovič *et al.* [8] have addressed this drawback by using an on-board IMU to correctly approximate the water surface even in calm seas. However, stereo reconstruction cannot detect partially submerged obstacles, and detection of distant obstacles requires an increase in stereo baseline, which negatively affects the stability of the USV.

Bai *et al.* [24] proposed a ship detection method based on infrared images. The method successfully and accurately detects various ships, but cannot detect arbitrary objects that do not emit enough heat and are thus invisible to the infrared camera. Kristan *et al.* [4] proposed a semantic segmentation method based on fitting structured models to the RGB image, which is able to detect both partially submerged obstacles and those protruding through the water surface. Bovcon *et al.* [10], [6] extended this method with a stereo camera system and an IMU sensor to improve the accuracy of water edge estimation and reduce the number of false-positive detections caused by sun glint and sea foam. Yang *et al.* [25] have extend [4] by a pre-segmentation module for online adjustment of priors and to further improve the overall segmentation. Despite achieving state-of-the-art results, approaches [4], [10], [6], [25] rely solely on simple visual features that are not expressive enough to reliably address the diversity of a marine scene, resulting in poor segmentation in the presence of visual artifacts on the water such as wakes, sea foam, glitter, and reflections.

Deep convolutional neural networks (CNNs) enable the learning of rich features specialized to the target task, and several works have considered the application of general-purpose detection CNNs to the aquatic domain. Lee *et al.* [26] and Yang *et al.* [25] applied a state-of-the-art deep classifier [27] to detect and classify different types of vessels. Ma *et al.* [28] extended a Faster R-CNN [27] classifier with ResNet [29] backbone and improved the existing DenseNet [30] block by adding trainable weight parameters to each skip connection. The method achieved state-of-the-art classification on their private dataset. A drawback of category-specific object detection methods (i.e., [26], [25], [28]) is that they cannot detect arbitrary obstacles that were not included in the training phase.

Segmentation-based CNNs are better than object detection CNNs for detecting general obstacles, but these methods are notoriously data hungry and require large and meticulously annotated datasets for training. Several marine datasets have been proposed [31], [10], [18], [32], [16], [21], but few are annotated per pixel, i.e., [32], [16], [21]. Of these, [21] contain annotations only for the water region and [32] is a semi-automatically segmented collection of sequences from

SMD [18] – however, the annotations were made for training classifiers and are not accurate enough for training segmentation networks. MaSTr1325 [16] was specifically designed for training segmentation networks and is currently the only publicly available diverse maritime dataset with accurate annotations of the water, obstacle, and sky pixels for training large segmentation-based networks.

Two studies [15], [16] have evaluated general deep segmentation networks from the AGV domain on the task of obstacle detection in maritime surveillance. Cane *et al.* [15] used a filtered ADE20k [11] dataset for training and several maritime datasets (MODD [4], IPATCH [31], SEAGULL [33], and SMD [18]) for evaluation. Their study benchmarked three popular deep segmentation networks of varying complexity: SegNet [34], ENet [35], and ESPNet [36]. SegNet achieved the highest overall accuracy and precision, followed by ENet, while ESPNet was better at correctly classifying objects. Bovcon *et al.* [16] trained a disjoint set of widely used segmentation networks (PSPNet [37], DeepLab2 [38] and UNet [39]) on MaSTr1325 and evaluated them on the challenging MODD2 [10] dataset. The DeepLab2 architecture, which has a deep encoder with atrous convolutions, produced the most promising results. Nevertheless, both studies [16], [15] showed consistent drawbacks in water segmentation accuracy and misclassification of small obstacles, leading to the conclusion that water environment-specific segmentation architectures are needed. These have only recently emerged. Kim *et al.* [40] applied skip-connections and whitening layers to E-Net [35] to improve segmentation accuracy of small obstacles and reported improvements over DeepLab3+ [14] on their private dataset. Steccanella *et al.* [21] enhanced the U-Net [39] architecture with depth-wise convolutional layers and reported state-of-the-art performance on the IntCatch [21] dataset. This network, however, attempts to separate the water from the rest of the environment, which in turn leads to problems in the presence of reflections where regions of the water have very similar visual information to the environment above the water surface. Zhan *et al.* [41] proposed an online adaptation of their network from external measurements and super-pixel clustering. While their method requires accurate initialization and time-consuming non-autonomous calibration, they report state-of-the-art performance on their private dataset.

The network proposed in this work builds on the best practices of published methods and goes beyond the state of the art in several ways. We use a DeepLab2 encoder as a starting point for our work, as it has shown great promise [16]. Inspired by [10], [6], we introduce inertial sensor information into the network to improve the accuracy of water edge estimation in the presence of visual ambiguities. For the fusion of inertial and visual data, we have developed a novel decoder that, in combination with a carefully designed loss function, refines the segmentation. This in turn enables the detection of small and distant obstacles, which is currently one of the main drawbacks of existing networks [15], [16].

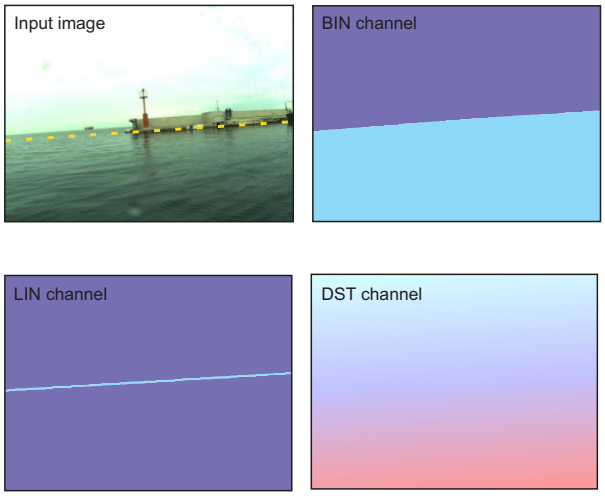


Fig. 2: Three IMU feature channel encoding methods calculated from the estimated horizon line (yellow dotted line). Zero, positive, and negative values in the channels are shown as blue, cyan, and red, respectively.

III. The WaSR network architecture

The proposed water segmentation and refinement obstacle detection network (WaSR, shown in Figure 1) consists of an encoder path and a decoder path. The encoder computes rich features at different levels of detail, while the decoder fuses inertial and visual information and increases the spatial resolution of the segmentation output. In the following, we describe the encoder for inertial measurements in Section III-A, the image encoder in Section III-B, and the decoder in Section III-C. A novel semantic separation loss function is presented in Section III-D, and Section III-E describes the segmentation postprocessing procedure.

A. The inertial measurements encoder

The IMU information has been used in previous works [10], [6] with some success to improve obstacle detection in handcrafted algorithms. In particular, the IMU-camera geometry [10] allows horizon projection into the image to constrain water edge estimation. In a convolutional neural network, a natural IMU encoding is a feature channel with pixel values set according to the horizon location, which can serve as a prior water probability. However, it is unclear what is the best method for channel encoding of this probability.

We therefore investigate three IMU feature channel encoding methods, all of which are based on computing the horizon location by the projection model from [10]. The first method (LIN) draws a thick line of pixels with value one on the horizon, while the remaining pixels are set to zero. The second method (DST) encodes a signed distance to the horizon, i.e., each pixel above the horizon is assigned a positive distance, while pixels below the horizon are assigned a negative distance. The third method (BIN) creates a binary mask where all pixels below the horizon are set to one and the others are set to zero. Examples of the three IMU feature channel encoding methods are shown in Figure 2.

B. The image encoder

Encoding a diverse water appearance requires significant capacity of the encoder network. Following our recent analysis of different architectures [16], we chose a general-purpose segmentation backbone from [38], i.e., a ResNet101 [29] with atrous convolutions. Specifically, the model consists of four residual convolutional blocks (*res2, res3, res4* and *res5*) in combination with max-pooling layers (see Figure 1). Hybrid atrous convolutions [42] are added to the last two blocks to increase the receptive field and encode a local context information into deep features.

C. The decoder

The task of the decoder is to fuse features from the image encoder and the IMU encoder and refine them into the final segmentation output (fusion block in Figure 1). The fusion block considers features from three encoding blocks (*res2*, *res3*, and *res5*) to leverage the generalization strength of the coarse resolution *res5* features and the details of the high resolution *res2*/*res3* features.

We build on our preliminary decoder [17] and extend the fusion block with additional modules to improve segmentation quality. Its architecture is based on a gradual fusion approach using Attention Refinement Modules (ARMs), similar to [13], to learn an optimal fusion strategy for different value scales between the IMU and the different visual feature channels. First, the encoded E_5 features are simultaneously input to the pruned Atrous Spatial Pyramid Pooling [38] (ASPP) with three pyramid layers and the ARM1 module (Figure 3). The pruned ASPP module applied to coarse resolution E_5 features is used to refine segmentation of small maritime structures, while the ARM1 block handles fusion and reweighting of visual and inertial features using global average pooling and depth reduction with normalization. The resulting features of the pruned ASPP module and the ARM1 block are fused using an adaptive fusion method (Feature Fusion Module, similar to [13]) called FFM1 (Figure 3), which generates $1,024$ feature channels. These features are further fused with the encoded E_3 features and the IMU channel by another ARM block, which we call ARM2 (Figure 3). ARM2 first applies an ARM1 block to fuse the IMU-channel and the E_3 features from the encoder. This is followed by a set of 1×1 convolutions to double the number of feature channels, which are summed per channel with the features from the FFM1 at the bottom of the decoder. Finally, the refined and reweighted features from the ARM2 block, along with the encoded E_2 features and the feature channel IMU, are fed to the second Feature Fusion Module, which is called the FFM (Figure 3). The FFM implements more complex fusion pathways than ARM and is used to combine the ARM2 low-level and E_2 high-level features. In particular, the ARM2 output features are up-sampled and concatenated with the E_2 features and the IMU feature channel. The depth of the resulting feature channels is halved by a 3×3 convolution block and normalized by a batch normalization block. A weight vector is computed similarly to ARM1 and used to reweight the features, leading to feature selection and fusion.

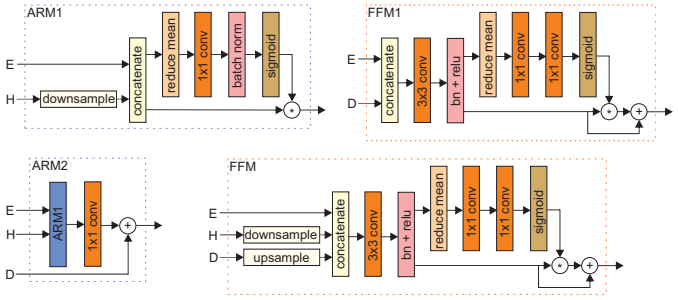


Fig. 3: The attention refinement modules ARM1, ARM2 and the feature fusion modules FFM, FFM1 adjust the scaling of the heterogeneous input feature channels and gradually fuse inertial and visual information in the fusion block. The encoder features, horizon feature channel, and decoder features from the previous layer are denoted as E, H and D, respectively.

The output of the decoder fusion block (Figure 1) is finally processed to extract per-pixel semantic labels. Our recent study [16] has shown that Atrous Spatial Pyramid Pooling [38] (ASPP) leads to significant improvements in the segmentation of small maritime structures while incurring only a small computational overhead. Therefore, the ASPP block followed by a softmax is added as the last block in our decoder. The resolution of the output is one-fourth of the input resolution, so the output is up-sampled by a factor of four to match the resolution of the input image.

D. Enforcing separation of semantic features

Care must be taken when designing a loss function for the maritime environment. While some obstacles may be large, the majority of pixels in a typical marine scene belong to either water or sky. This leads to a class imbalance that overwhelms the classical cross entropy loss [43]. Moreover, the difficulty of segmentation varies greatly between different water regions. For example, it may be easy to classify regions of slightly rippled blue water, but it is much more difficult to classify glitter and reflections of objects as the water component. Therefore, to adapt the focus of the network to challenging regions during training, we use a weighted focal loss [44], $\mathcal{L}_{\text{foc}}$, adapted for segmentation, and add the classical L_2 loss ($\mathcal{L}_{L_2}$) for weight regularisation [45].

Our recent study [16] has shown that water phenomena such as glitter and object reflections are a major challenge for water segmentation networks. While mistaking water for sky is not a threat, mistaking obstacles for water and vice versa leads to potential USV collision or frequent false alarms, rendering the network useless for practical navigation. To avoid this, the network should ideally learn the encoding in early layers such that it produces very similar features for a variety of water appearances and very different features for obstacles. This facilitates subsequent learning of the classifier in the higher layers of the network.

We propose to enforce early feature separation by designing a novel loss. Let $\{x_j^c\}_{j\in R_1}$ and $\{x_j^c\}_{j\in R_2}$ be features in channel c belonging to pixels of semantic components R_1 and R_2, respectively. To enforce a clustering of the R_1 features,

we can fit their distribution by a Gaussian with per-channel means$\{\mu^c\}_{c\in N_c}$ and the variances $\{\sigma^{c2}\}_{c\in N_c}$, where N_c is the number of channels, and we assume channel independence for computational tractability. The similarity of all other pixels corresponding to the semantic component R_2 can be measured as a joint probability under this Gaussian, i.e.,

$$p(\{x_j\}_{j\in R_1}) \propto \prod_{\substack{j\in R_1 \\ c=1:N_c}} exp(-0.5(x_j^c-\mu^c)^2/\sigma^{c2}). \quad (1)$$

We would like to enforce learning of features that minimize this probability. By expanding the equation for the semantic component R_1 per channel standard deviations, taking the logarithm of (1), flipping the sign, and inverting, we get the following equivalent separation loss

$$\mathcal{L}_{\mathrm{R_1}}^{\mathrm{R_2}} = \frac{N_{R_2}}{N_C N_{R_1}} \sum_{c}^{N_c} \frac{\sum_{i\in R_1}(x_i^c-\mu^c)^2}{\sum_{j\in R_2}(x_j^c-\mu^c)^2}, \quad (2)$$

which clusters the semantic component R_1 and separates it from the semantic component R_2. The N_{R_1} and N_{R_2} are added as normalisation constants, making the scale independent of the number of channels and the number of pixels of the semantic component in each frame.

Ideally, we would like to separate the semantic components of water and obstacles as much as possible, so as to reduce the number of false positive and false negative detections and improve the safety of autonomous navigation. To this end, we propose four different variants of loss (2):

1) *A water-obstacle separation (WSL)* focuses on learning the appearance of the water component, grouping different appearances of water into a tight cluster of features that is well separated from the obstacle features. The separation loss is defined as $\mathcal{L}_{\mathrm{R_1}}^{\mathrm{R_2}} = \mathcal{L}_{\mathrm{wat}}^{\mathrm{obs}}$, where *obs* and *wat* denote obstacles and water semantic components, respectively.
2) *An obstacle-water separation (OSL)* focuses on learning the appearance of obstacles by grouping various appearances of obstacles into a tight cluster of features and enforces separation from the water features. The separation loss is defined as $\mathcal{L}_{\mathrm{R_1}}^{\mathrm{R_2}} = \mathcal{L}_{\mathrm{obs}}^{\mathrm{wat}}$.
3) *A water-obstacle separation combined with a sky-obstacle separation (WSSL)* tightly clusters features of the water and sky component and separates them from the obstacles features. The combined separation loss is defined as $\mathcal{L}_{\mathrm{R_1}}^{\mathrm{R_2}} = \mathcal{L}_{\mathrm{wat}}^{\mathrm{obs}} + \mathcal{L}_{\mathrm{sky}}^{\mathrm{obs}}$, where *sky* denotes the sky semantic component.
4) *A water-obstacle separation combined with an obstacle-water separation (WOSL)* groups various appearances of water and obstacles into two tight clusters that are well separated from each other. The combined separation loss is defined as $\mathcal{L}_{\mathrm{R_1}}^{\mathrm{R_2}} = \mathcal{L}_{\mathrm{wat}}^{\mathrm{obs}} + \mathcal{L}_{\mathrm{obs}}^{\mathrm{wat}}$.

The final loss is a weighted summation of the individual losses, i.e.,

$$\mathcal{L} = \mathcal{L}_{\mathrm{foc}} + \lambda_1 \mathcal{L}_{\mathrm{R_1}}^{\mathrm{R_2}} + \lambda_2 \mathcal{L}_{\mathrm{L_2}}, \quad (3)$$

where λ_1 and λ_2 are the weights.

E. Segmentation post-processing

The WaSR network described in Section III-C outputs a segmentation mask in which each pixel belongs to exactly one semantic component (water, sky, or environment). Pixels labelled water are used to construct the water region mask as in [10]. The largest contiguous component in the water region mask represents the navigable surface of the USV, and its top edge corresponds to the edge of the water. The list of potential obstacles is obtained by extracting pixel blobs with environment labels within the water region.

IV. Experimental evaluation

This section reports on the analysis of WaSR on public benchmarks. The datasets and evaluation protocol are described in Section IV-A, while the implementation details are given in Section IV-B. The different decoder fusion blocks are compared in Section IV-C, the IMU encoding techniques are compared in Section IV-D, and the impact of water-obstacle separation loss is analyzed in Section IV-E. Section IV-F compares the best-performing WaSR architecture to the state of the art, and domain generalization capabilities are analyzed in Section IV-G.

A. Performance evaluation protocol

All tested networks were trained on the MaSTr1325 [16] dataset, which to our knowledge is the only sufficiently large and diverse maritime dataset available for training deep segmentation methods with detailed per-pixel segmentation masks. The dataset was acquired from an USV equipped with a compass, LIDAR, GPS, IMU, two side cameras, and a main stereo camera system *Vrmagic VRmMFC* which consists of two *Vrmagic VRmS-14/C- COB* CCD sensors with a baseline $0.3\,\mathrm{m}$, *Thorlabs MVL4WA* lens with $3.5\,\mathrm{mm}$ focal length, maximum aperture of $\mathrm{f}/1.4$, and a $132.1°$ FOV. The main stereo camera system is connected to the on-board computer via a USB 2.0 bus, which limits the data flow to 10 frames per second at a resolution of $1,278 \times 958$ pixels. The dataset contains $1,325$ unique images captured over a 24 month period. The images are time-synchronised with available on-board sensors and manually annotated per pixel for three semantic components: Sea, Sky, and Obstacles. The edges of the semantically distinct components are labelled with an additional label "unknown" to address annotation uncertainty and enable automatic exclusion of these pixels from the learning procedure. Images from the dataset and their corresponding ground truth annotations are shown in the top row of Figure 4.

Dataset augmentation is used to increase the generalisation of the trained networks. From the common approaches to data augmentation presented in [46], we selected two augmentation types that fit the maritime domain - vertical mirroring and central rotations. These augmentations increase the variety of local textures in the training set. The rotation parameter is constrained to $\pm\{5, 15\}$ degrees to prevent the generation of unrealistic images. We additionally applied an elastic deformation to the water component of the training images to artificially simulate waves and curls. Finally, following [16],

we also applied colour transfer augmentation, resulting in a total of $54,325$ training images.

Performance was evaluated on two very different marine datasets (MODD2 [10] and SMD [18]) to analyse the transferability and general applicability of the tested segmentation methods. MODD2 [10] was recorded in the coastal region of the Adriatic Sea, near the Gulf of Koper, Slovenia. It consists of 28 visually distinct stereo sequences time-synchronised with IMU measurements. For its acquisition a USV was used for the recording, with similar technical characteristics as described above. A wide variety of scenarios within the sequences, combined with extreme conditions (object mirroring, glint, and various weather conditions) make this dataset the most challenging USV dataset publicly available to date. Images from this dataset are shown in Figure 4 (middle row). Obstacles and water edges in MODD2 [10] are annotated with bounding boxes and a polygon, respectively. We noticed that some of the far obstacles in MODD2 were not annotated, which can bias the number of false positives. We fixed this by adding the missing annotations. The bounding box annotation of the large ship that appears in sequence 25 (Figure 4 middle row, first column) and is entirely located above the horizon, was removed because it degrades the ground-truth water-edge accuracy. Following [16] guidelines, only the left camera images were used for analysis.

The SMD [18] dataset was recorded at different locations in Singapore harbour. It consists of 66 mono sequences containing the following: On-board footage, On-shore footage and Near-infrared footage. The on-board and on-shore sequences were shot with a handheld Canon 70D camera with Canon EF 70-300mm f/4-5.6 lens, while the near-infrared footage was shot with a modified Canon 70D with Mid-Opt BP800 Near-IR bandpass filter. Unlike MODD2, SMD includes sequences taken at dusk, in the middle of the night, and even in light rain. We selected a subset of 13 challenging sequences (images in the bottom row of Figure 4) that fit our problem domain and were taken from a fixed position. Obstacles and water edges in SMD [18] are annotated with bounding boxes and a polygon, respectively. Due to a lack of IMU data, we manually annotated the horizon line in the first frame of each sequence and use its location to artificially generate inertial data.

Fast and accurate detection is critical for autonomous systems. To gain speed, all input images were prescaled to 512×384 pixel resolution using bi-linear interpolation. At the chosen resolution, all dangerous obstacles remain visible. Detected as well as ground-truth obstacles with an area of less than 5×5 pixels (at the original resolution of $1,278 \times 958$) were ignored because they are either too small or too far away to pose a threat to the USV.

Following [16], the standard performance evaluation measures from [4] are used. The accuracy of water edge estimation is given by the mean square error computed over all sequences and measured in pixels. Obstacle detection accuracy, primarily measured by the number of true positives (TP) and false positives (FP), is expressed by the precision $(\mathrm{Pr} = \frac{\mathrm{TP}}{\mathrm{TP+FP}})$, recall $(\mathrm{Re} = \frac{\mathrm{TP}}{\mathrm{TP+FN}})$ and its harmonic mean, i.e. $\mathrm{F1} = \frac{2 \cdot \mathrm{Pr} \cdot \mathrm{Re}}{\mathrm{Pr+Re}}$. We additionally report the average number of true-positive (TPr) and false-positive (FPr) detections per one hundred images to give a better insight into the amount of detections triggered.

Fig. 4: MaSTr1325 [16] (top), MODD2 [10] (middle) and SMD [18] (bottom) datasets exhibit large scene and appearance variability.

B. Implementation details

All networks were trained for 10 epochs with early stopping using the RMSProp optimizer [47] with a momentum 0.9, an initial learning rate 10^{-4} and standard polynomial reduction decay of 0.9. The chosen values of the hyperparameters follow good practice and are commonly used in deep learning approaches. Based on a preliminary study, the loss weights from Equation (3) were set to $\lambda_1 = 1.0$ and $\lambda_2 = 10^{-2}$. The ResNet101 backbone was pre-trained on ImageNet [48], while the remaining additional trainable parameters were randomly initialized using the Xavier [49] initialization method.

The WaSR network was implemented in Tensorflow 1.2.0[1] and all experiments were performed on a desktop computer with Intel Core i7-7700 3.6GHz CPU and nVidia GTX1080 Ti GPU with 11GB GRAM.

C. Analysis of the decoder fusion block variants

We analyze and compare the performance of two distinct fusion block variants: the preliminary fusion block variant [17] (denoted FU1) and our proposed fusion block of Section III-C (denoted FU2). Both variants were evaluated on the MODD2 dataset using the BIN IMU encoding technique (Section III-A) and the water obstacle separation loss (Section III-D). The results for the WaSR variants, denoted as $\mathrm{WaSR}_{\mathrm{FU1}}$ and $\mathrm{WaSR}_{\mathrm{FU2}}$, are shown in Table I.

Both variants achieve comparable results for water edge estimation. The FU1 variant estimates the water edge better in the presence of prominent wakes (Figure 5 second column), while the FU2 variant estimates the water edge better in the presence of reflections and sun glitter (Figure 5 third column).

[1]A reference WaSR implementation is made publicly available on our project page https://box.vicos.si/borja/viamaro/index.html.

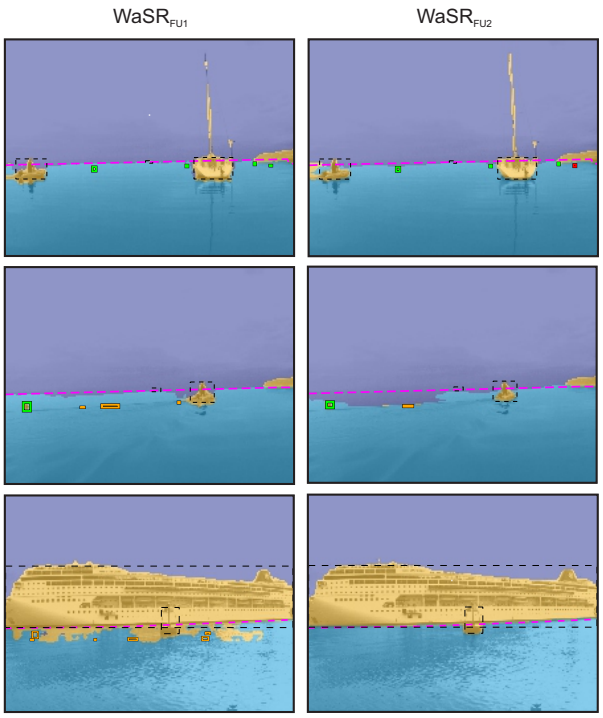


Fig. 5: Qualitative comparison of using the FU1 (WaSR$_{FU1}$) and FU2 (WaSR$_{FU2}$) decoder fusion block variants. Correctly detected obstacles are marked with a green bounding box, false positive detections are marked with an orange bounding box, and undetected obstacles are marked with a red bounding box. The ground truth water edge is indicated by a dotted pink polygon.

In terms of detection accuracy, the proposed FU2 outperforms the FU1 by far. The FU1 variant achieves a slightly better precision resulting from a higher number of TP detections. Most of these detections are small obstacles with an area size of less than 900 pixels located near the horizon (Figure 5 first column). Nevertheless, the difference in the average number of TP detections (TPr) is marginal. On the other hand, the average number of FP detections (FPr) is twice that of FU2, which significantly degrades its precision by six percentage points compared to FU2. A large number of FPs are located on glitter/reflection below the ship (Figure 5 third column) and on a wake (Figure 5 second column). In contrast, the proposed FU2 variant segments these regions more precisely, which contributes to a lower number of FPs and consequently to the best overall F1 score.

We conclude that the proposed FU2 fusion block architecture contributes to a more robust segmentation of the water components and consequently to a significantly lower number of FP detections. Although there is a slight decrease in TP detections, this does not significantly affect the overall accuracy.

D. Analysis of the IMU feature channel encoding methods

The IMU feature channel encoding methods from Section III-A were analyzed by evaluating WaSR with the FU2 fusion block on the MODD2 dataset. In the following, we denote WaSR variants by IMU *line encoding* (WaSR$_{LIN}$), *distance encoding* (WaSR$_{DST}$), *binary encoding* (WaSR$_{BIN}$) and a variant without IMU fusion (WaSR$_{NOIMU}$).

TABLE I: The performance of the different decoder fusion block variants, given by the water-edge estimation error in pixels (μ_{edg}), the average number of true-positive (TPr) and false-positive (FPr) detections per hundred images. We additionally report precision, recall, and F1 scores in percentages.

Architecture	μ_{edg}	Pr	Re	TPr	FPr	F1
WaSR$_{FU1}$ [17]	**10.0**	88.6	**97.8**	**52.8**	6.8	93.0
WaSR$_{FU2}$	10.5	**94.6**	96.5	52.1	**3.0**	**95.5**

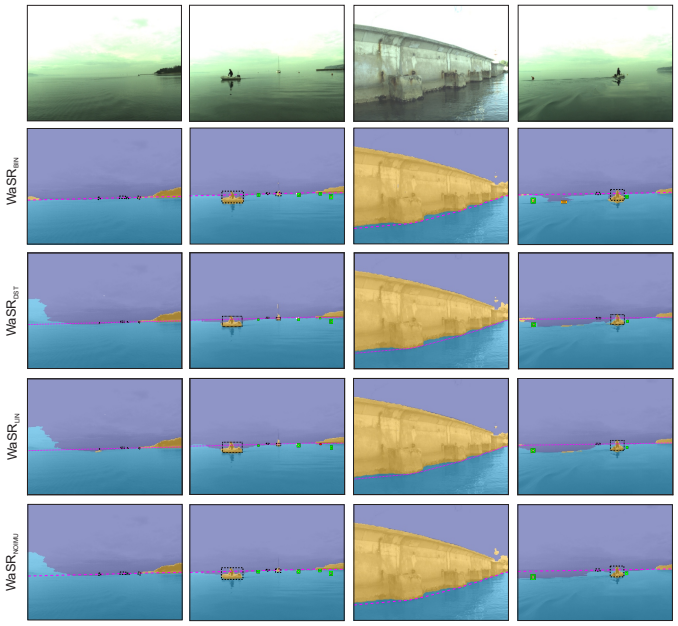


Fig. 6: Qualitative analysis of the effects of using different IMU feature channel encoding methods. The sky, obstacles, and water are indicated by deep blue, yellow, and cyan, respectively. Detected obstacles are indicated by green (true positive), orange (false positive), and red (false negative).

The results in Table II show that the most accurate water-edge estimation is achieved by WaSR$_{BIN}$. WaSR$_{BIN}$ also achieves the best precision and recall scores, indicating that it is able to accurately detect the most TPs while keeping the number of FPs low.

The improvements from using IMU are most pronounced when the USV faces open water (Figure 6 first, second, and fourth columns). We observe that the water edge is consistently better estimated with IMU fusion and that the IMU channel generation method plays an important role. Despite an overall good performance, WaSR$_{DST}$ and WaSR$_{LIN}$ tend to overestimate the water edge (Figure 6, first column) and perform worse compared to WaSR$_{BIN}$. Thus, the binary channel encoding method seems to be a preferred method for encoding the inertial representation.

Figure 6 (last column) shows a failure case where all WaSR variants underestimate the water edge, and it shows room for further improvement.

TABLE II: Comparison of different IMU encodings on the water edge estimation error μ_{edg}, measured in pixels, the average number of true-positive (TPr) and false-positive (FPr) detections per hundred images. We also report the precision (Pr), recall (Re), and F1 scores in percentage.

Architecture	μ_{edg}	Pr	Re	TPr	FPr	F1
$\mathrm{WaSR}_{\mathrm{BIN}}$	**10.5**	**96.5**	**96.5**	**52.1**	**3.0**	**95.5**
$\mathrm{WaSR}_{\mathrm{DST}}$	11.1	92.7	89.8	48.5	3.8	91.3
$\mathrm{WaSR}_{\mathrm{LIN}}$	11.7	92.5	87.0	47.0	3.8	89.7
$\mathrm{WaSR}_{\mathrm{NOIMU}}$	11.1	94.1	90.6	48.9	3.1	92.3

TABLE III: Comparison of WaSR variants with separation loss and WaSR trained without separation loss ($\mathrm{WaSR}_{\mathrm{NOSL}}$). We report the water edge estimate μ_{edg}, measured in pixels, the average number of true-positive (TPr) and false-positive (FPr) detections per hundred images. We also report the precision (Pr), recall (Re), and F1 scores in percentage.

Architecture	μ_{edg}	Pr	Re	TPr	FPr	F1
$\mathrm{WaSR}_{\mathrm{NOSL}}$	11.0	91.7	92.1	49.7	4.5	91.9
$\mathrm{WaSR}_{\mathrm{WSL}}$	**10.5**	94.6	**96.5**	**52.1**	3.0	95.5
$\mathrm{WaSR}_{\mathrm{OSL}}$	11.3	**96.9**	94.5	51.0	**1.6**	**95.7**
$\mathrm{WaSR}_{\mathrm{WSSL}}$	10.6	93.9	95.2	51.4	3.4	94.5
$\mathrm{WaSR}_{\mathrm{WOSL}}$	11.1	93.2	94.4	50.9	3.7	93.8

E. Analysis of the semantic separation loss

The purpose of semantic separation loss (Section III-D) is to learn a transformation that groups various appearances of the selected semantic component into a tight cluster of features that is well separated from the features of the other semantic components. Five baseline WaSR architectures using the FU2 fusion method and the BIN IMU encoding method were re-trained with different combinations of semantic separation losses: a water-obstacle separation loss ($\mathrm{WaSR}_{\mathrm{WSL}}$), an obstacle-water separation loss ($\mathrm{WaSR}_{\mathrm{OSL}}$), a combination of water-obstacle and sky-obstacle separation loss ($\mathrm{WaSR}_{\mathrm{WSSL}}$) and a combination of water-obstacle and obstacle-water separation loss ($\mathrm{WaSR}_{\mathrm{WOSL}}$) as well as a variant without semantic separation loss ($\mathrm{WaSR}_{\mathrm{NOSL}}$).

A qualitative analysis was performed first. A set of 50 images with distinctive and challenging scenes from MaSTr1325 was selected. Features from the WaSR *res4b20* block (the layer right after the semantic separation loss) were extracted and projected onto 2D using a t-distributed Neighbour Embedding method (t-SNE) [50].

The projected features for WaSR variants are shown in Figure 7. Note that the features of different semantic categories (especially water and obstacles) are indeed much better separated when semantic separation loss is applied. A favourable separation of water and obstacle pixels leads to a more accurate estimate of the water edge and a better detection rate (more TP, less FP and FN detections). Although the proposed semantic separation loss separates water and obstacles relatively well in general (Figure 7 second to fifth images), some water and obstacle/sky pixels remain intertwined for $\mathrm{WaSR}_{\mathrm{WSL}}$, $\mathrm{WaSR}_{\mathrm{WSSL}}$, and $\mathrm{WaSR}_{\mathrm{WOSL}}$. These pixels lead to FP detections. In addition, the separation of sky and obstacle elements is generally imperfect, due to the poorer annotation quality of thin obstacle structures (poles, cranes, and treetops) in sky regions.

The results on the MODD2 [10] dataset shown in Table III quantitatively support the observations from Figure 7. $\mathrm{WaSR}_{\mathrm{WSL}}$ estimates water-edge the best, which is due to additional focused learning of the water appearance. The increase in the overall accuracy of water edge estimation is mainly seen in the improved segmentation of nearby large objects and the mainland. All the proposed semantic separation losses significantly improve the detection accuracy, which is most evident in the significant increase in both precision and recall scores. The highest recall score is achieved by the $\mathrm{WaSR}_{\mathrm{WSL}}$ variant, which detects the most TPs due to an overall better segmentation of water components resulting from focused learning of water appearances. Improved segmentation of water appearance is shown in Figure 8 above. Additional learning of obstacle appearances increases accuracy by significantly reducing the number of FPs (seen in $\mathrm{WASR}_{\mathrm{OSL}}$, Figure 8 bottom). However, since the training set contains only a limited number of obstacles, the variants that use obstacle-water separation loss ($\mathrm{WaSR}_{\mathrm{OSL}}$ and $\mathrm{WaSR}_{\mathrm{WOSL}}$) are not able to generalize well at TP detections. The best F1 is achieved by $\mathrm{WaSR}_{\mathrm{OSL}}$ and $\mathrm{WaSR}_{\mathrm{WSL}}$ with negligible difference, however $\mathrm{WaSR}_{\mathrm{WSL}}$ detects more obstacles.

F. Comparison with the current state-of-the-art

Based on the analysis from previous sections, we have selected the WaSR architecture with BIN IMU encoding and FU2 fusion method with water-obstacle separation loss as the best performing architecture - hereafter we will simply refer to it as WaSR. For comparison, we selected ten current state-of-the-art segmentation networks with different architecture of the encoder and decoder. The first six architectures (PSPNet [11], SegNet [34], $\mathrm{DeepLab2}_{\mathrm{NOCRF}}$ [16], DeepLab3+ [51], BiSeNet [13], and RefineNet [12]) use a deep ResNet101 backbone for feature extraction and different decoder architectures. These networks were selected because they achieve state-of-the-art performance in both autonomous vehicle segmentation tasks and general segmentation tasks. The remaining four state-of-the-art networks are variants of a fully-convolutional DenseNet [52] (FC-DenseNet56 and FC-DenseNet103) and UNet [39] (MobileUNet [53] and $\mathrm{IntCatch}_{\mathrm{full}}$ [21]). These networks achieve astonishing performance on different domains [54], [55], [21] and have lightweight architectures with much less trainable parameters (Table IV) than the aforementioned networks. Moreover, the $\mathrm{IntCatch}_{\mathrm{full}}$ method has reportedly achieved the best performance in a maritime domain despite its lightweight architecture. All the selected networks, including the proposed WaSR, run at more than 12 frames per second (Table IV), which is considered real-time since the given camera system transmits data over the USB 2.0 bus and does not support frame rates higher than ten frames per second.

The results on MODD2 [10], are summarized in Table V. WaSR outperforms all competing networks in water edge estimation by a large margin. Second best is BiSeNet [13], with

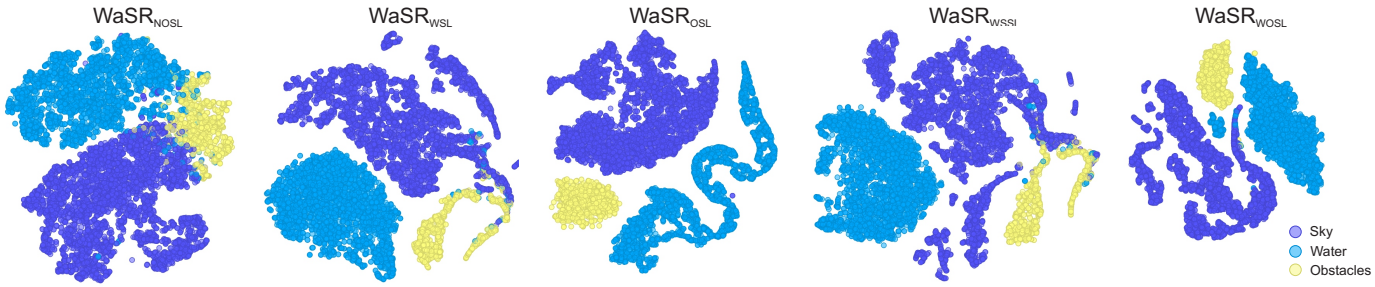


Fig. 7: T-SNE visualization of water, obstacle and sky features extracted from WaSR$_{WSL}$, WaSR$_{OSL}$, WaSR$_{WSSL}$, WaSR$_{WOSL}$ and WaSR without the separation loss (WaSR$_{NOSL}$). The semantic clusters from WaSR$_{NOSL}$ are not well separated, while well-separated and compact clusters are observed for WaSR variants with semantic separation loss.

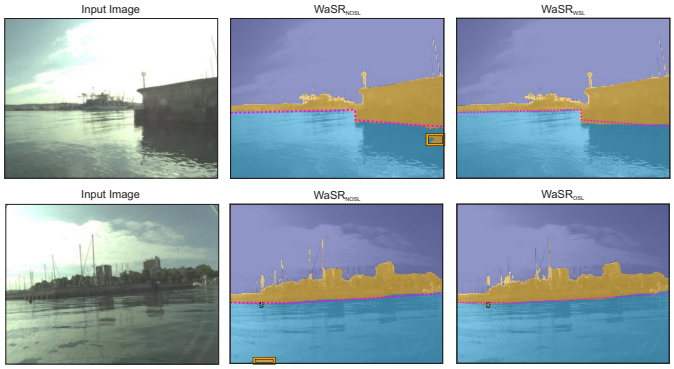


Fig. 8: Qualitative comparison of WaSR variants with different separation losses and their effect on water appearance segmentation and obstacle detection.

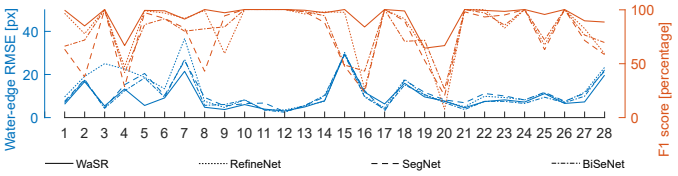


Fig. 9: Per-sequence performance for the four best methods (WaSR, RefineNet [12], BiSeNet [13] and SegNet [34]) on MODD2 [10].

accuracy approximately two pixels lower, and closely followed by DeepLab2$_{NOCRF}$ [38]. Visual inspection (see Figure 10 for examples) shows that other networks have difficulty accurately estimating the water edge in the presence of haze on the horizon, while WaSR neither overestimates nor underestimates its position. This observation is also confirmed by the graph shown in Figure 9, where a significant improvement is seen in estimating the water edge in sequences where the horizon is not well visible (sequences 5, 6 and 7). WaSR also shows impressive robustness to strong ambient reflections in the water, accurately estimating the water edge even under these ambiguous conditions (Figure 10 third row).

WaSR achieves the best recall due to the highest TP rate, followed by PSPNet [11], RefineNet [12], and SegNet [34]. WaSR also achieves the best precision score, closely followed by DeepLab2$_{NOCRF}$ [38], which triggers the lowest number of false alarms, but at the cost of reduced TPs.

For safe and uninterrupted autonomous navigation, a trade-off between the number of FP and TP detections is required, which is summarized by the F1 score. The best performing methods based on F1 are WaSR, RefineNet [12], BiSeNet [13] and SegNet [34]. Further insight into the quality of the segmentation masks is provided by the precision, recall, and F1 plots as a function of the obstacle detection overlap threshold. For the sake of presentation clarity, only the plots of the four best performing networks are shown in Figure 11. The proposed WaSR network has the highest precision and recall scores at moderate overlap thresholds, followed by RefineNet. However, the order changes at very high thresholds (above 70%), where RefineNet performs best. This means that WaSR detects more obstacles with fewer false positives, which confirms the results of the detailed analysis in Figure 9. Further inspection showed that a large fraction of the extra TPs detected by WaSR and missed by other networks are of small size (area smaller than 900 pixels). However, a curve dip at high thresholds (Figure 11) suggests that these detections are not well localized. Nonetheless, accurately segmenting smaller obstacles is very challenging for all networks, as shown by the graphs in Figure 11, where performance decreases with an increase in the overlap threshold.

The qualitative comparison (Figure 10 second, third line) shows that WaSR detects smaller obstacles more accurately than the other networks. Most other networks produce false positives on glitters, reflections, and wakes (Figure 10). Exceptions are DeepLab2$_{NOCRF}$ [38] and RefineNet [12], which are robust to such scenarios. However, both networks are prone to poor detection of isolated obstacles, resulting in a relatively low TP rate. Poor segmentation also occurs in the presence of distinct wakes surrounding the dinghy (Figure 10 first and second lines), resulting in either degraded water edge estimation or FP detections at wave edges.

The last row of Figure 10 visualizes a failure case common to all evaluated networks, where they inefficiently segment the pier in close proximity. In addition to incomplete segmentation of the pier, some networks, such as SegNet, tend to trigger FP detections on reflections in the water. This suggests a potential for improvement, despite the overall robustness of the networks.

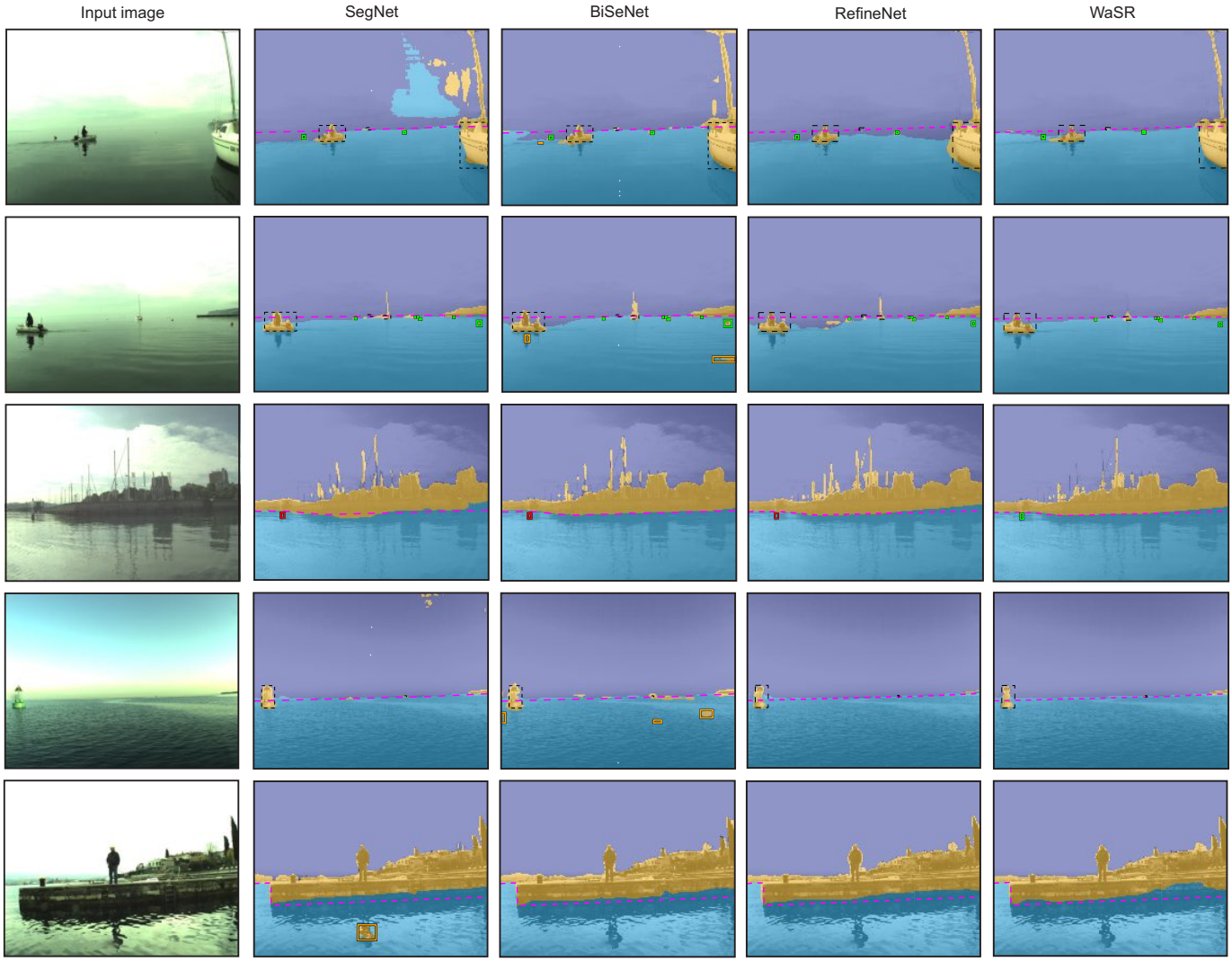


Fig. 10: Segmentation quality for the four best performing methods from Table V. The sky, obstacles, and water components are marked with deep blue, yellow, and cyan colours, respectively. Correctly detected obstacles are marked with a green bounding box, false positive detections with an orange bounding box, and undetected obstacles with a red bounding box. The ground-truth water edge is marked with a dashed pink polygon. WaSR is the most robust method for a wide range of scenes.

TABLE IV: The number of trainable parameters (N_{param}) for each tested architecture and their average time required to segment a single image, measured in milliseconds.

Architecture	N_{param}	δ_t [ms]
PSPNet [11]	56.0M	22.2
SegNet [34]	35.0M	33.5
DeepLab2$_{\text{NOCRF}}$ [38]	44.0M	32.6
DeepLab3+ [14]	48.0M	12.6
BiSeNet [13]	47.5M	17.7
RefineNet [12]	85.7M	38.5
WaSR$_{\text{FU1}}$ [17]	71.4M	71.3
WaSR$_{\text{FU2}}$	84.6M	63.9
FC-DenseNet56 [52]	1.4M	51.5
100-Layer Tiramisu [52]	9.3M	80.4
MobileUNet [53]	8.9M	31.2
IntCatch$_{\text{full}}$ [21]	1.9M	7.1

TABLE V: MODD2 [10] performance in terms of water edge estimation error μ_{edg} in pixels, the average number of true-positive (TPr) and false-positive (FPr) detections per hundred images. We also report the precision (Pr), recall (Re), and F1 scores in percentages.

Architecture	μ_{edg}	Pr	Re	TPr	FPr	F1
PSPNet [11]	13.5	56.8	93.2	50.3	38.2	70.6
SegNet [34]	13.2	74.3	92.5	49.9	17.2	82.4
DeepLab2$_{\text{NOCRF}}$ [38]	12.5	94.5	62.6	33.8	**2.0**	75.3
DeepLab3+ [14]	13.8	64.9	84.2	45.4	24.6	73.3
BiSeNet [13]	12.0	77.1	90.4	48.4	14.5	83.2
RefineNet [12]	14.4	90.2	92.7	50.0	5.4	91.4
WaSR	**10.5**	**94.6**	**96.5**	**52.1**	3.0	**95.5**
FC-DenseNet56 [52]	14.5	74.5	91.4	49.3	16.9	82.1
100-Layer Tiramisu [52]	13.1	73.0	89.2	48.2	17.9	80.3
MobileUNet [53]	13.8	54.5	89.2	48.1	40.1	67.7
IntCatch$_{\text{full}}$ [56]	20.4	52.8	82.7	44.6	39.8	64.5

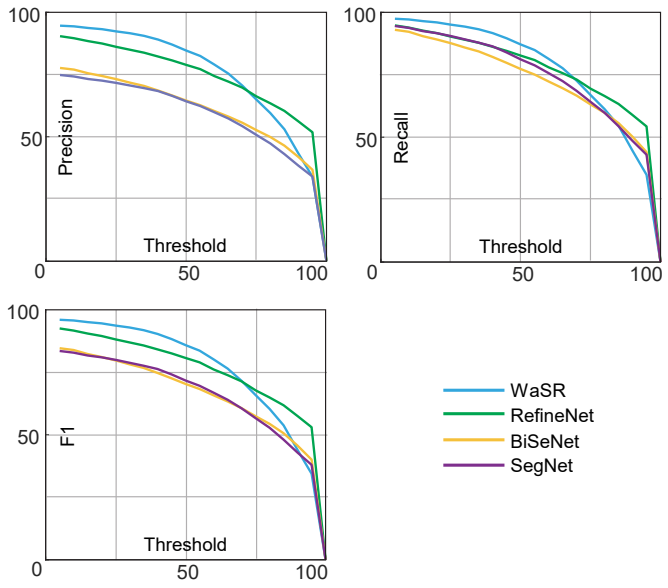


Fig. 11: Precision, Recall and F1 plots for the four top-performing networks on MODD2 [10] as a function of obstacle detection overlap threshold.

G. Domain transfer analysis

To evaluate the generalization capabilities, we trained WaSR with different semantic separation losses on MaSTr1325 and evaluated them on the SMD [18] sequences without fine-tuning on this dataset. The following top three performers from Table V were selected for comparison: RefineNet [12], BiSeNet [13], and SegNet [34]. The results are shown in Table VI.

The WaSR variants significantly outperform all other networks in estimating water edges, beating the second best method, RefineNet [12], by approximately nine pixels. The WaSR_{WSL} variant estimates the water edge best. Nevertheless, the performance of all WaSR variants evaluated for this task is comparable, as the difference in water edge estimation is less than one pixel. A qualitative comparison in Figure 12 confirms the accurate water edge estimation over a range of scene appearances in SMD. Other methods result in either a drastic overestimation (e.g., SegNet [34], Figure 12 first line) or a severe underestimation of the water edge caused by contiguous false-positive detections (e.g. SegNet [52] and RefineNet [12], Figure 12 third row). The per sequence performance in Figure 13 further confirms these observations.

WaSR_{WSL} achieves the best recall score due to the highest true-positive (TP) detection rate, followed by $\text{WaSR}_{\text{WSSL}}$ and BiSeNet [13]. This order is also evident from the qualitative comparison (fourth row of Figure 12), where WaSR_{WSL} successfully detects all obstacles, while $\text{WaSR}_{\text{WSSL}}$ misses a rower on the right. All WaSR variants trigger significantly fewer false positive detections (FPr) on average compared to other networks, as indicated in Figure 12. The main source of FP detections are sea foam, prominent wakes and different water textures, as shown in the second and third rows of Figure 12. $\text{WaSR}_{\text{WSSL}}$ achieves the lowest average false positive detection rate (FPr), which contributes to its high precision score. A very high precision score, combined with a second best recall score, contributes to the best overall F1 score for $\text{WaSR}_{\text{WSSL}}$. However, the WaSR_{WSL} variant lags behind the F1 score by approximately one percentage point. The per-sequence performance in Figure 13 shows that the overall improvement in F1 score in $\text{WaSR}_{\text{WSSL}}$ is mainly due to better performance on sequence eight. Further investigation revealed that WaSR_{WSL} triggers approximately 600 more FP detections compared to $\text{WaSR}_{\text{WSSL}}$ on this sequence alone. The majority of these false detections are due to different water texture, which was not observed during training.

TABLE VI: Generalization performance on SMD [18] in water edge estimation error μ_{edg}, the average number of true-positive (TPr) and false-positive (FPs) detections per hundred frames. We also report the precision (Pr), recall (Re), and F1 scores in percentages.

Architecture	μ_{edg}	Pr	Re	TPr	FPr	F1
SegNet [34]	31.0	31.2	76.3	74.0	163.2	44.3
BiSeNet [13]	28.9	15.9	88.6	85.9	452.7	27.0
RefineNet [12]	24.8	49.2	79.7	77.3	79.9	60.8
WaSR_{WSL}	**16.0**	77.0	**94.1**	**91.2**	27.2	84.7
WaSR_{OSL}	16.6	72.6	87.2	84.6	31.9	79.3
$\text{WaSR}_{\text{WSSL}}$	16.3	**81.1**	91.5	88.7	**20.6**	**86.0**
$\text{WaSR}_{\text{WOSL}}$	16.3	52.9	87.2	84.6	75.4	65.8

Surprisingly, the segmentation of the *night sequence* (Figure 12, second row) is overall accurate and consistent for all methods, despite the fact that no similar images or conditions were present in the training dataset. This indicates that all tested methods generalize well under significant changes in ambient lighting.

Regardless of the overall accurate segmentation, the compared methods still tend to occasionally trigger false alarms for water ripples. Such failure cases are shown in the last row of Figure 12. As can be seen from the results in Table VI, SegNet triggers the most FP detections, which are also the largest in terms of surface area. Similarly, WaSR_{WSL} triggers more false alarms than the $\text{WaSR}_{\text{WSSL}}$ variant, which is consistent with the evaluation results. RefineNet does not detect any FPs in this case, but misses many obstacles, which is much more dangerous from the point of view of autonomous navigation safety.

V. Conclusion

A novel obstacle detection deep neural network, WaSR, for autonomous USV navigation was presented. WaSR addresses water edge segmentation and obstacle detection by fusing visual information with inertial data from an on-board IMU. A deep encoder extracts rich visual features from the input image, while a non-symmetric shallow decoder fuses the visual features with the inertial data. Additional robustness is achieved by introducing a novel semantic separation loss at the end of the encoder, which supports learning a feature space with increased separation between semantic components appearances.

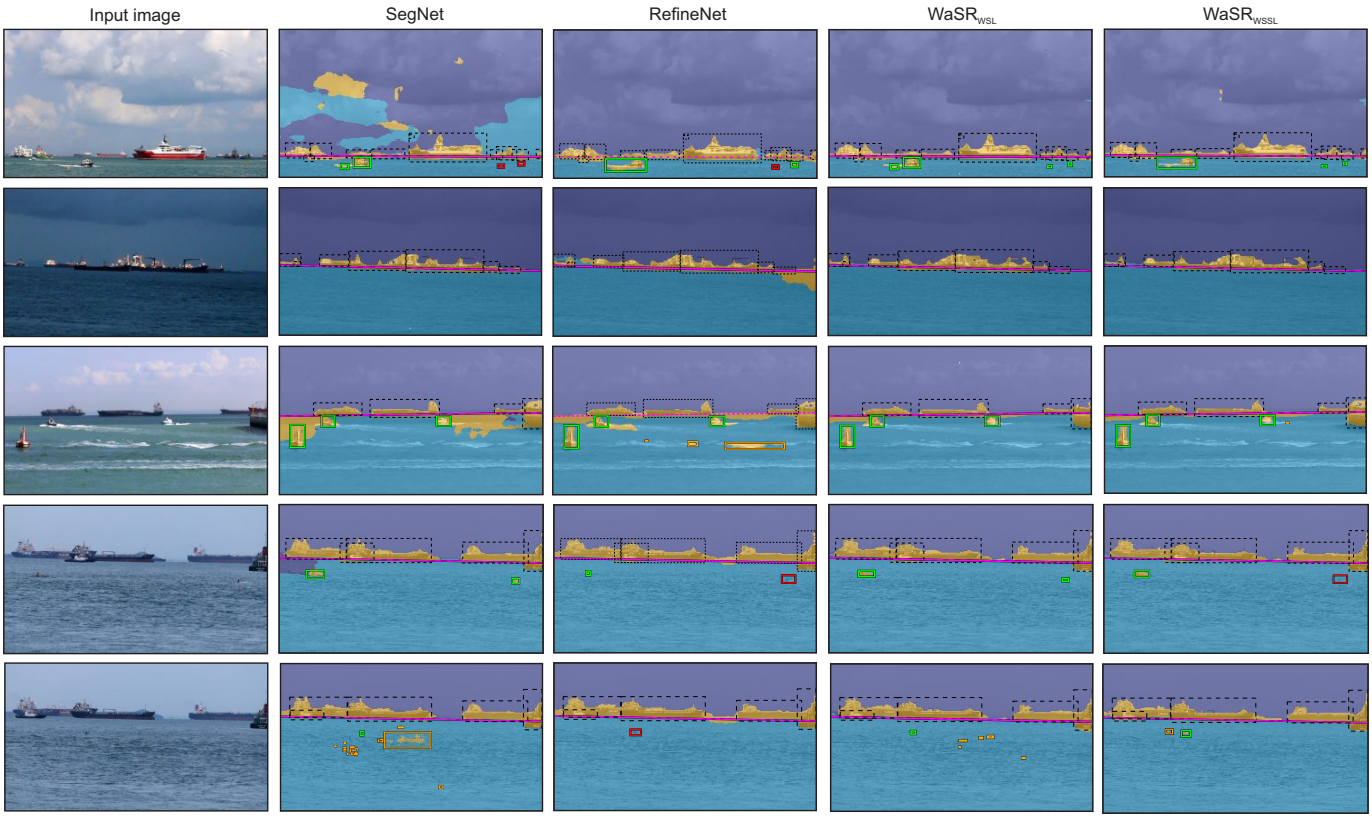


Fig. 12: Quality of segmentation generalisation on SMD [18]. The sky, obstacles, and water components are marked with deep blue, yellow, and cyan colours, respectively. Correctly detected obstacles are marked with a green bounding box, false positive detections are marked with an orange bounding box, and undetected obstacles are marked with a red bounding box. The ground-truth water edge is represented by a dotted pink polygon.

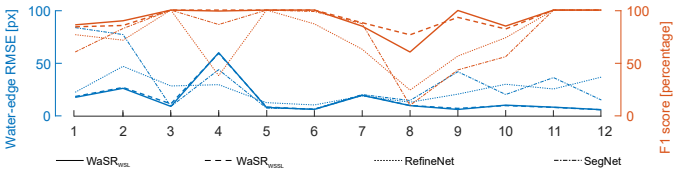


Fig. 13: Per-sequence generalization performance for the four best methods (WaSR$_{WSL}$, WaSR$_{WSSL}$, RefineNet [12], and SegNet [34]) on SMD [10].

WaSR has been analyzed in detail and compared with the current state of the art. The results show that WaSR outperforms the best performing state-of-the-art segmentation network BiSeNet [13] by more than two pixels on average, which means that the obstacle localization error is significantly reduced by several hundred meters for the obstacles and the land near the horizon. In the obstacle detection task, WaSR outperforms the state of the art by 4% in F1 score. Compared to the best related method RefineNet [12], WaSR significantly improves the precision and recall scores by halving the number of FPs and increasing the average TP rate by about two percentage points. Moreover, the WaSR variant that does not use IMU data also outperforms the compared methods, but not to this extent.

The generalizability of WaSR was evaluated on the SMD [18] sequences. WaSR outperformed the best related method RefineNet [12] by approximately nine pixels in water edge estimation. It also achieved a significantly better F1 score in obstacle detection. This improvement is due to a higher number of TP detections, which increase the recall score, and a significant decrease in FP detections, which improve the precision score.

An ablation study showed that each part of our network (a deep ResNet101 [29] backbone with atrous convolutions, combined with a novel fusion block decoder architecture with multiple IMU fusions and the semantic separation loss) contributes to performance improvements - the best performance is achieved with all parts together. Detailed analysis showed that sufficient depth of the ResNet101 [29] encoder is critical for successful encoding of the various water textures. This affects both the detection rate and the accuracy of the water edge estimation.

Two decoder fusion block variants with different complexity were analyzed. Although the water edge estimation accuracy and the number of TPs were comparable for both variants, the more complex variant with additional FFM and ASPP modules halved the number of FPs and improved the F-measure (Section III-C). Regardless of the increased complexity, WaSR still runs in real-time at nearly 16 frames per second.

Several IMU feature-channel encoding techniques were investigated, and the best results were obtained with binary

encoding (Section III-A). The improvements from IMU fusion are most apparent when the USV is looking at open water, especially in cases where the location of the water's edge is not clear due to haze on the horizon. In these situations, the accuracy of the water edge estimate is significantly improved.

The new semantic separation loss function of Section III-D plays an important role in guiding the learning procedure to map a variety of specific semantic component appearances into a cluster of features that are well separated from those corresponding to the remaining semantic components. Various adaptations of semantic separation loss were investigated. Clustering water features and separating them from the obstacle features (WSL) leads to a better segmentation of the water components, which is reflected in an improved accuracy of the water edge estimation and an increased number of TPs. On the other hand, clustering obstacle features and separating them from water features (OSL) increases accuracy by reducing the number of FPs, but it does not generalize well in TP detections to sequences with water-like obstacles that are significantly different from those observed in training.

We plan to inspect several directions of future work. On standard GPUs, WaSR runs at moderate frame rates. We thus plan exploring network compression techniques and backbone replacements to reduce the number of network parameters and thus speed-up the inference. Overall faster inference and a smaller amount of layers will allow fitting the network to low-end embedded GPUs with small power consumption, which will increase the range of applications on smaller USVs. Despite the excellent robustness, the segmentation output of WaSR is not ideal in the presence of significant wakes and leaves a room for improvement. This opens research in direction of adaptation methods to adjust the network parameters to the current scenes (akin to hand-crafted methods like [4]) and fusion with alternative sensors such as LIDAR to improve the localization accuracy of small and distant obstacles. Although WaSR achieves state-of-the-art generalization performance, many false positives are still triggered due to visual discrepancies when we use different on-board sensing devices. To improve the generalization capabilities, we plan to explore domain adaptation methods based on adversarial learning, such as [57].

Acknowledgment

This work was supported in part by the Slovenian research agency (ARRS) programmes P2-0214 and P2-0095, and the Slovenian research agency (ARRS) research project J2-8175.

References

[1] C. Onunka and G. Bright, "Autonomous marine craft navigation: On the study of radar obstacle detection," in *2010 11th International Conference on Control Automation Robotics & Vision*, Dec 2010, pp. 567–572.

[2] A. R. J. Ruiz and F. S. Granja, "A short-range ship navigation system based on ladar imaging and target tracking for improved safety and efficiency," *IEEE Transactions on Intelligent Transportation Systems*, vol. 10, no. 1, pp. 186–197, March 2009.

[3] H. K. Heidarsson and G. S. Sukhatme, "Obstacle detection and avoidance for an autonomous surface vehicle using a profiling sonar," in *2011 IEEE International Conference on Robotics and Automation*, May 2011, pp. 731–736.

[4] M. Kristan, V. S. Kenk, S. Kovačič, and J. Perš, "Fast image-based obstacle detection from unmanned surface vehicles," *IEEE transactions on cybernetics*, vol. 46, no. 3, pp. 641–654, 2016.

[5] T. Cane and J. Ferryman, "Saliency-based detection for maritime object tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2016, pp. 18–25.

[6] B. Bovcon and M. Kristan, "Obstacle detection for usvs by joint stereo-view semantic segmentation," in *2018 IEEE/RSJ, International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 5807–5812.

[7] D. K. Prasad, C. K. Prasath, D. Rajan, L. Rachmawati, E. Rajabally, and C. Quek, "Object detection in a maritime environment: Performance evaluation of background subtraction methods," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–16, 2018.

[8] J. Muhovič, B. Bovcon, M. Kristan, J. Perš *et al.*, "Obstacle tracking for unmanned surface vessels using 3-d point cloud," *IEEE Journal of Oceanic Engineering*, 2019.

[9] H. Wang and Z. Wei, "Stereovision based obstacle detection system for unmanned surface vehicle," in *Proc. 2013 IEEE Int. Conf. on Robotics and Biomimetics*, 2013, pp. 917–921.

[10] B. Bovcon, J. Perš, M. Kristan *et al.*, "Stereo obstacle detection for unmanned surface vehicles by IMU-assisted semantic segmentation," *Robotics and Autonomous Systems*, vol. 104, pp. 1–13, 2018.

[11] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2881–2890.

[12] G. Lin, A. Milan, C. Shen, and I. Reid, "Refinenet: Multi-path refinement networks for high-resolution semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1925–1934.

[13] C. Yu, J. Wang, C. Peng, C. Gao, G. Yu, and N. Sang, "Bisenet: Bilateral segmentation network for real-time semantic segmentation," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 325–341.

[14] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 801–818.

[15] T. Cane and J. Ferryman, "Evaluating deep semantic segmentation networks for object detection in maritime surveillance," in *2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*. IEEE, 2018, pp. 1–6.

[16] B. Bovcon, J. Muhovič, J. Perš, and M. Kristan, "The mastr1325 dataset for training deep usv obstacle detection models," in *2019 IEEE/RSJ, International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 3431–3438.

[17] B. Bovcon and M. Kristan, "A water-obstacle separation and refinement network for unmanned surface vehicles," in *2020 IEEE/Robotics and Automation Society, International Conference on Robotics and Automation (ICRA)*. IEEE, 2020.

[18] D. K. Prasad, D. Rajan, L. Rachmawati, E. Rajabally, and C. Quek, "Video processing from electro-optical sensors for object detection and tracking in a maritime environment: a survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 8, pp. 1993–2016, 2017.

[19] J. Larson, M. Bruch, R. Halterman, J. Rogers, and R. Webster, "Advances in autonomous obstacle avoidance for unmanned surface vehicles," SPAWAR San Diego, Tech. Rep., 2007.

[20] Q. Fu, C. Hu, J. Peng, F. C. Rind, and S. Yue, "A robust collision perception visual neural network with specific selectivity to darker objects," *IEEE Transactions on Cybernetics*, pp. 1–15, 2019.

[21] L. Steccanella, D. Bloisi, A. Castellini, and A. Farinelli, "Waterline and obstacle detection in images from low-cost autonomous boats for environmental monitoring," *Robotics and Autonomous Systems*, vol. 124, p. 103346, 2020.

[22] T. Huntsberger, H. Aghazarian, A. Howard, and D. C. Trotz, "Stereo vision–based navigation for autonomous surface vessels," *Journal of Field Robotics*, vol. 28, no. 1, pp. 3–18, 2011.

[23] P. Wang, P. Chen, Y. Yuan, D. Liu, Z. Huang, X. Hou, and G. Cottrell, "Understanding convolution for semantic segmentation," in *2018 IEEE winter conference on applications of computer vision (WACV)*. IEEE, 2018, pp. 1451–1460.

[24] X. Bai, Z. Chen, Y. Zhang, Z. Liu, and Y. Lu, "Infrared ship target segmentation based on spatial information improved fcm," *IEEE Transactions on Cybernetics*, vol. 46, no. 12, pp. 3259–3271, 2016.

[25] J. Yang, Y. Li, Q. Zhang, and Y. Ren, "Surface vehicle detection and tracking with deep learning and appearance feature," in *2019*

5th International Conference on Control, Automation and Robotics (ICCAR). IEEE, 2019, pp. 276–280.

[26] S.-J. Lee, M.-I. Roh, H.-W. Lee, J.-S. Ha, I.-G. Woo *et al.*, "Image-based ship detection and classification for unmanned surface vehicle using real-time object detection neural networks," in *The 28th International Ocean and Polar Engineering Conference*. International Society of Offshore and Polar Engineers, 2018.

[27] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.

[28] L. Ma, W. Xie, and H. Huang, "Convolutional neural network based obstacle detection for unmanned surface vehicle." *Mathematical biosciences and engineering: MBE*, vol. 17, no. 1, pp. 845–861, 2019.

[29] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[30] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708.

[31] L. Patino, T. Nawaz, T. Cane, and J. Ferryman, "Pets 2017: Dataset and challenge," in *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, July 2017, pp. 2126–2132.

[32] S. Moosbauer, D. Konig, J. Jakel, and M. Teutsch, "A benchmark for deep learning based object detection in maritime environments," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2019, pp. 0–0.

[33] M. M. Marques, P. Dias, A. S. Ferreira *et al.*, "Unmanned aircraft systems in maritime operations: Challenges addressed in the scope of the seagull project," in *OCEANS 2015-Genova*. IEEE, 2015, pp. 1–6.

[34] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 12, pp. 2481–2495, 2017.

[35] A. Paszke, A. Chaurasia, S. Kim, and E. Culurciello, "Enet: A deep neural network architecture for real-time semantic segmentation," *arXiv preprint arXiv:1606.02147*, 2016.

[36] S. Mehta, M. Rastegari, A. Caspi, L. Shapiro, and H. Hajishirzi, "Espnet: Efficient spatial pyramid of dilated convolutions for semantic segmentation," in *Proceedings of the european conference on computer vision (ECCV)*, 2018, pp. 552–568.

[37] C. Y. Jeong, H. S. Yang, and K. D. Moon, "Horizon detection in maritime images using scene parsing network," *Electronics Letters*, vol. 54, no. 12, pp. 760–762, 2018.

[38] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 4, pp. 834–848, 2018.

[39] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.

[40] H. Kim, J. Koo, D. Kim, B. Park, Y. Jo, H. Myung, and D. Lee, "Vision-based real-time obstacle segmentation algorithm for autonomous surface vehicle," *IEEE Access*, vol. 7, pp. 179 420–179 428, 2019.

[41] W. Zhan, C. Xiao, Y. Wen, C. Zhou, H. Yuan, S. Xiu, Y. Zhang, X. Zou, X. Liu, and Q. Li, "Autonomous visual perception for unmanned surface vehicle navigation in an unknown environment," *Sensors*, vol. 19, no. 10, p. 2216, 2019.

[42] L.-C. Chen, G. Papandreou, I. Kokkinos, A. L. Yuille *et al.*, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 4, pp. 834–848, 2017.

[43] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[44] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2980–2988.

[45] A. Krogh and J. A. Hertz, "A simple weight decay can improve

[47] G. Hinton, N. Srivastava, and K. Swersky, "Neural networks for machine learning," *Coursera, video lectures*, vol. 264, p. 1, 2012.

generalization," in *Advances in neural information processing systems*, 1992, pp. 950–957.

[46] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *Journal of Big Data*, vol. 6, no. 1, p. 60, 2019.

[48] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. Ieee, 2009, pp. 248–255.

[49] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, 2010, pp. 249–256.

[50] L. v. d. Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of machine learning research*, vol. 9, no. Nov, pp. 2579–2605, 2008.

[51] H. Liu, Y. Liu, X. Gu, Y. Wu, F. Qu, and L. Huang, "A deep-learning based multi-modality sensor calibration method for USV," in *2018 IEEE Fourth International Conference on Multimedia Big Data (BigMM)*. IEEE, 2018, pp. 1–5.

[52] S. Jégou, M. Drozdzal, D. Vazquez, A. Romero, and Y. Bengio, "The one hundred layers tiramisu: Fully convolutional densenets for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2017, pp. 11–19.

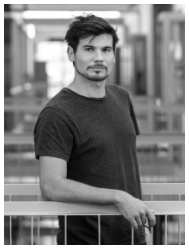
[53] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[54] L. Liu and Y. Zhou, "A closer look at u-net for road detection," in *Tenth International Conference on Digital Image Processing (ICDIP 2018)*, vol. 10806. International Society for Optics and Photonics, 2018, p. 108061I.

[55] L. Teixeira, M. R. Oswald, M. Pollefeys, and M. Chli, "Aerial single-view depth completion with image-guided uncertainty estimation," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1055–1062, 2020.

[56] L. Steccanella, D. Bloisi, J. Blum, and A. Farinelli, "Deep learning waterline detection for low-cost autonomous boats," in *International Conference on Intelligent Autonomous Systems*. Springer, 2018, pp. 613–625.

[57] Q. Wang, J. Gao, and X. Li, "Weakly supervised adversarial domain adaptation for semantic segmentation in urban scenes," *IEEE Transactions on Image Processing*, vol. 28, no. 9, pp. 4376–4386, 2019.



Borja Bovcon Received his M.Sc. degree from the Faculty of Mathematics and Physics at University of Ljubljana in 2017. He is currently working as a researcher at the ViCoS Laboratory, Faculty of Computer and Information Science, University of Ljubljana. His research interests are computer vision, obstacle detection and autonomous systems.

Matej Kristan Received the Ph.D. degree from the Faculty of Electrical Engineering, University of Ljubljana, in 2008. He is an associate professor at the Faculty of Computer and Information Science, University of Ljubljana. His research interests include probabilistic methods for computer vision with focus on visual tracking, semantic segmentation, object detection and online learning, and computer vision for autonomous robots. He is a member of the IEEE.