24th Computer Vision Winter Workshop
Friedrich Fraundorfer, Peter M. Roth, Fabian Schenk (eds.)
Stift Vorau, Austria, February 6–8, 2019

DOI: 10.3217/978-3-85125-652-9-07

Benchmarking Semantic Segmentation Methods for Obstacle Detection on a Marine Environment

Borja Bovcon, Matej Kristan
University of Ljubljana
Faculty of Computer and Information Science
1000 Ljubljana
borja.bovcon@fri.uni-lj.si
matej.kristan@fri.uni-lj.si

Abstract. *Obstacle detection is an important and critical module of autonomous navigation. Majority of modern obstacle detection algorithms are based on semantic segmentation and scene understanding. Most of these methods were developed for autonomous ground vehicles and their performance has not yet been evaluated for autonomous boats. In this paper, we (i) benchmark and analyze the most common segmentation algorithms for autonomous driving on a marine environment, (ii) propose a new, pixel-wise annotated, maritime training set for fine-tuning segmentation methods, (iii) conduct an in-depth study of their performance on Modd2 dataset, pinpoint their drawbacks along with their qualities and (iv) compare the results of classical segmentation metrics against obstacle detection metric in terms of USV safety.*

1. Introduction

Small-sized unmanned surface vehicles (USVs) are an affordable tool for navigating in shallow waters and narrow marinas. They are mainly used for coastal environmental patrol and remote inspection of difficult-to-reach man-made structures. These tasks require a high level of autonomy which primarily depends on timely detection and avoidance of nearby obstacles and floating debris. Lightweight and information-rich sensors, such as cameras, combined with computer vision algorithms are gaining prominence as leading obstacle detection mechanisms.

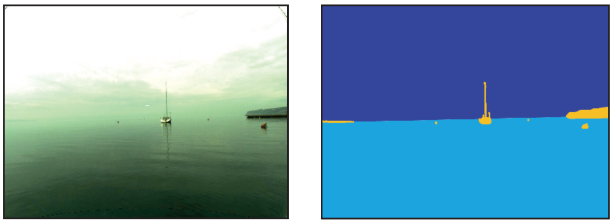

Figure 1 *Sample image and its ground truth segmentation. Sky, obstacles and water are represented with deep blue, yellow and cyan color respectively.*

Obstacles can be detected by various image-processing approaches, for instance background subtraction [33], foreground extraction [12], 3-D reconstruction [34], semantic segmentation [17, 6, 7, 5] etc. Recently, the use of deep learning has contributed significantly to the striking progress in the field of semantic segmentation. The main goal of semantic segmentation methods is to perform a pixel-wise classification of the image, that provides necessary information for scene understanding. Scene understanding is a crucial part for a successful and safe autonomous navigation. Many methods [4, 3, 20, 24, 21], developed for navigation of autonomous ground vehicles, rely on semantic segmentation algorithms to detect obstacles in a scene. Siam *et al.* [32] have done an in-depth comparison of such semantic segmentation methods for autonomous driving and proposed a real-time segmentation benchmarking framework. However, in marine environment different assumptions hold and different segmentation tasks pose a challenge. For instance, the appearance of water varies significantly due to waves and weather conditions. Moreover, submerged and small obstacles might also present a significant threat to the USV.

In this paper we benchmark three commonly used state-of-the-art deep learning semantic segmentation methods (U-Net [31], PSP-Net [37] and DeepLab-v2 [8]) on a marine environment. We evaluate each method based on traditional segmentation metrics and compare the results against metrics used in a marine Multi-modal obstacle detection dataset 2 (Modd2) [7]. The Modd2 is currently one of the largest publicly-available datasets. It consists of a challenging sequences where the sky and the water component are not always distinguishable due to the unfavouring weather conditions. To accurately train selected deep learning segmentation methods, we provide 280 representative images of a marine environment with pixel-wise ground truth annotations.

2. Related work

Obstacle detection for unmanned surface vehicles is still a relatively young research area. A common practice for obstacle detection in marine environment is the use of range sensors - for instance radar [2, 25], sonar [15] etc. Range sensors have difficulties discriminating between water and land in the far field [11] and suffer from scanning rate limitations. Moreover radar also has problems detecting small, non-metallic obstacles. Larson *et al.* [18] presented advances in obstacle avoidance for USVs and pointed out the use of camera as an affordable and information rich alternative. Prasad *et al.* [27] have done an extensive survey of various background subtraction methods and evaluated their performance on Singapore Marine Dataset (SMD) [28] as obstacle detection mechanisms. Analysis shows that spurious dynamics of water and wakes are a leading cause of multiple false detections.

Estimating the water-edge in an image can significantly limit the region of interest (ROI) where obstacles occur. Wang *et al.* [35] combine saliency detection and motion estimation to search for obstacles below the estimated water edge. Their assumption of a sharp boundary between water and sky when estimating the water edge is in practice often violated. In [34], Wang *et al.* introduced the use of a stereo camera system to perform 3-D reconstruction of the scene, which enables them to detect obstacles above the water surface. However, only obstacles that significantly protrude through the water can be detected. Another problem arises in the state of a calm sea, where water lacks a texture, thus leading to a degraded 3-D reconstruction of the scene and consequently inaccurate water surface estimation. Alternatively Kristan *et al.* [17] proposed a graphical model (SSM) for monocular obstacle detection via semantic segmentation. The algorithm generates a water-segmentation mask and treats all blobs inside the water region as obstacles. SSM successfully detects both obstacles protruding through the surface and the floating ones, it does not assume a straight water edge and runs in real-time. Nevertheless, it still fails in the presence of visual ambiguities. For example, when the boat faces open water and the horizon is obscured by haze.

The line separating the water and sky component might not be clear due to the unfavouring weather conditions like haze. Bovcon *et al.* [6] addressed this issue by introducing measurements from the on-board IMU into the segmentation model. The IMU measurements are used to project the horizon into camera view and automatically adjust the priors and hyper-priors of the segmentation model. Their algorithm can correctly estimate the horizon even when obscured. In their recent work [5] the problem of numerous false positive detections has been addressed by a joint stereo image segmentation, where corresponding pixels in the left and right image are assigned to the same semantic region which consequently improves obstacle detection through enforced segmentation consistency. Paccaud *et al.* [26] focus on a lake-deployed USVs, where surrounding land is visible most of the time and the water surface is predominantly calm and without distinct waves. Similar to [6] they use IMU sensor to project the horizon line to the image and define the ROI in which they search for the water edge with RANSAC. On the obtained water component area they use Sobel operator along x- and y-axis in combination with threshold to find blobs representing obstacles. Detected blobs are tracked within consecutive frames to identify false detections caused by glint and reflections. Method assumes that obstacles have sharp edges and is thus unable to detect partially submerged obstacles. Jeong *et al.* [16] use a scene parsing network (PSPNet [37] pre-trained on ADE20k dataset) to perform general segmentation of the image. The horizon approximation is obtained by searching for maximal vertical location corresponding to the sea component in each column of the segmentation mask. Location of the horizon is refined by iteratively applying least-squares regression on its points. The method was evaluated on SMD [28] where it achieved promising results. However, the SMD does not contain images with intense fog, where line between sea and sky is not visible. Lee *et al*. [19] proposed using deep learning network to detect and classify ships. They use a general Faster R-CNN [30], in combination with Bayesian framework to detect ships. Method is able to detect and classify seven different types of ships and cannot be used to distinguish arbitrary obstacles in the water without providing a large amount of additional training data.

3. Semantic Segmentation CNNs

In this section we present three commonly used neural network architectures for semantic segmentation. In Section 3.1 we outline the architecture of the U-Net [31], in Section 3.2 we mark out scene parsing network PSP-Net [37], while in Section 3.3 we describe the model of the DeepLab-v2 [8].

3.1. U-Net [31]

The U-Net, proposed by Ronneberger *et al*. [31], was initially designed for bio-medical image segmentation. Since then, it was used for various segmentation purposes ranging from segmentation of urban planning maps [13] to the road detection through segmentation [22, 36]. Its architecture, shown in Figure 2 top, incorporates an encoder which captures context and a symmetric decoder that provides precise localization. The encoder part consists of a repeated application of convolutions and a max pooling operation which halves the feature map size. After each down-sampling, the number of feature channels is doubled. In contrast, the decoder part of the network is comprised of an up-sampling of the feature map size, followed by a convolution that halves the number of feature channels. A skip connection, in form of a concatenation which combines the information from a corresponding layer in the encoder part, is followed by two convolutions. Each convolution in the network is followed by a rectified linear unit (ReLU). With a proper data augmentation, the network can be trained end-to-end and pixel-to-pixel on a set of very few images and still produce good results [31].

3.2. PSP-Net [37]

Zhao *et al*. [37] designed a state-of-the-art scene parsing network PSP-Net. Its architecture is visualized in Figure 2 middle. They use a pre-trained ResNet-50 [14] backbone with a dilated network strategy to extract features from the input image. The extracted feature map is then fed to pyramid pooling module, where features are fused under four different pyramid scales. After each pyramid level, a convolution is applied to reduce the dimension of context representation and maintain the weight of a global feature. Low dimension feature maps are up-sampled to the size of the original feature map via bi-linear interpolation and concatenated with the initial feature map. Concatenation is sent through a convolution to generate the final prediction map, which is further up-sampled to the original resolution.

3.3. DeepLab-v2 [8]

Chen *et al*. [8] proposed a segmentation model that uses ResNet-101 [14] backbone with atrous convolutions to extract features from the input image. Atrous convolutions enable them to explicitly control the resolution at which feature responses are computed and to enlarge the field-of-view (FOV) of filters. The main benefit of a larger FOV is obtaining a preponderant context without increasing the number of parameters. Passing multiple rescaled versions of the original image to parallel CNN branches allows them to perform a multi-scale semantic segmentation. The responses are combined with a fully connected Conditional Random Field (CRF) which improves the localization of object boundaries. Tuning of the CRF is done separately as a post-processing step. The architecture of DeepLab-v2 is shown in Figure 2 bottom.

4. Experimental setup

The dataset and evaluation protocol are described in Section 4.1 while implementation details of evaluated methods are given in Section 4.2.

4.1. The dataset and evaluation protocol

The performance of segmentation methods was analyzed on Modd2 [7], which consists of 11675 stereo images captured by a small-sized USV in the coastal waters of Marina Koper, Slovenia. The onboard cameras can accurately estimate the depth up to 185 m and their frame-rate is limited to 10 frames per second. Obstacles and water-edge in the dataset were manually annotated with bounding boxes and a polygon respectively. The segmentation CNNs from Section 3 require sufficient training data to produce satisfactory results. We have captured and handpicked 280 images under different weather conditions from Marina Koper using the acquisition system of [7]. These images were pixel-wise annotated

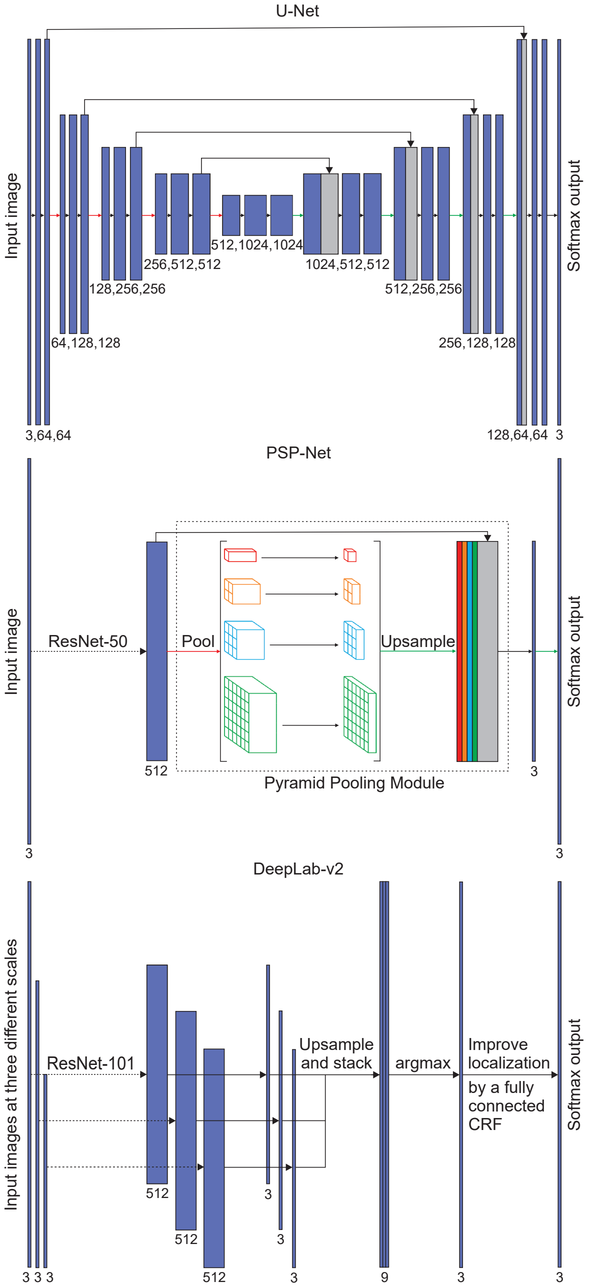

Figure 2 *Architecture illustration of tested CNNs. Black arrows represent 2-D convolutions, red arrows denote max pooling, while unpooling is marked with green arrows. The blue box corresponds to a multi-channel feature map, beneath which a number of feature channels is written. The gray box represents a copied feature map.*

for three classes (water, sky and environment) by human annotators (see Figure 1). The annotated images were further color augmented to increase diversity of the training set and replicate weather conditions from Modd2. For color augmentation we have handpicked seven descriptive images from Modd2 which were used as target images in the color transfer method [29] proposed by Reinhard *et al.* With data augmentation we have generated 1960 new training samples with accurate ground truth annotations. Figure 3 shows a sample image from the train set and its color augmentations. Timely and accurate obstacle detection is of central importance for autonomous navigation, so we rescaled images from Modd2 on two different resolutions - low (512×288) and high (896×512) to test the detection accuracy against processing speed.

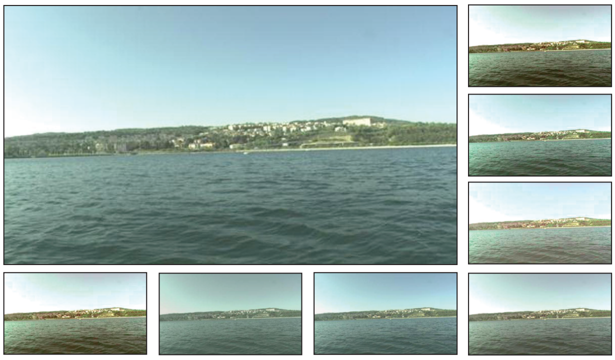

Figure 3 *Sample image from the train set (top left) surrounded by its color augmentation variations.*

For image segmentation evaluation purposes we have used metrics inspired by Long *et al.* [23]. These metrics are mean pixel accuracy $\left(\frac{\sum_i n_{ii}}{\sum_i t_i}\right)$, mean IOU $\left(\frac{1}{n_{cl}} \sum_i \frac{n_{ii}}{t_i+\sum_j n_{ji}-n_{ii}}\right)$ and frequency weighted IOU $\left(\left(\sum_k t_k\right)^{-1} \sum_i \frac{t_i n_{ii}}{t_i+\sum_j n_{ji}-n_{ii}}\right)$, where n_{cl} denotes the number of classes in the ground truth, n_{ji} represents the number of pixels of the class j predicted to belong to the class i, while t_i stands for the total number of pixels of class i in the ground truth segmentation. Segmentation metrics do not provide information on how many obstacles were detected, neither how accurately the sea-edge was approximated. For this task, the evaluation protocol of [7] was used. It measures the accuracy of the pixel-wise water-edge estimation by mean-squared error over all sequences, while the accuracy of obstacle detection is measured by the number of true positives (TP), false positives (FP), false negatives (FN) and by the overall F-measure, i.e., a harmonic mean of precision and recall.

4.2. Implementation details

A softmax-cross-entropy loss function and a stochastic gradient descent (SGD) optimization were used to train the segmentation networks. The initial learning rate was set to a low value of 0.009 and a weight decay factor of 0.7 was applied after every 5th epoch.

In our implementation of U-Net (Section 3.1) we have employed batch-normalization after each convolution and before ReLU activation to speed-up the training process. In the PSP-Net (Section 3.2) implementation we have initialized ResNet-50 weights, which were pre-trained on the ADE20k [38, 39] dataset, while in the DeepLab-v2 (Section 3.3) implementation we have restored ResNet-101 weights, which were pre-trained on the ImageNet [10] dataset. Additionally, we have modified the number of output channels in the last layer of both DeepLab-v2 and PSP-Net according to our dataset. The fine-tuning process was carried out for 60 epochs. A single-scale version of DeepLab-v2 (Section 3.3) is denoted as DeepLab-v2S, while its multi-scale counterpart is denoted as DeepLab-v2M.

The semantic segmentation methods were implemented in Python and use Tensorflow [1] back-end. All experiments were run on a desktop computer with Intel Core i7-7700 3.6 GHz CPU and nVidia GTX1080 Ti GPU.

5. Experimental results

We begin our analysis with Section 5.1 where we analyze semantic segmentation results, in Section 5.2 we interpret obstacle detection results, while Section 5.3 serves for qualitative comparison. Results of methods from Section 3 were compared against a baseline method ISSM [7]. The speed of the tested methods is analyzed in Section 5.4.

5.1. Semantic segmentation results

The semantic segmentation results are summarized in Table 1. On low-resolution images DeepLab-v2S achieves the highest accuracy, followed by DeepLab-v2M, PSP-Net and U-Net in the order given. The differences in results between DeepLab-v2S and DeepLab-v2M are 0.11%, 0.14% and 0.10% for the mean pixel accuracy, the mean IOU and the frequency weighted IOU, respectively.

On high-resolution images, DeepLab-v2M achieves the highest accuracy based on the mean pixel accuracy and the frequency weighted IOU, followed by DeepLab-v2S, PSP-Net and U-Net. The differences in results between the top two methods are 0.09% and 0.10% for the mean pixel accuracy and the frequency weighted IOU, respectively. Based on the mean IOU metric, DeepLab-v2S outperforms DeepLab-v2M by 0.11%.

Resolution 512 × 288

	Mean Pixel Accuracy	Mean IOU	Frequency Weighted IOU
U-Net [31]	93.12	88.82	86.27
PSPNet [37]	96.32	93.33	93.02
DeepLab-v2S [8]	**98.07**	**96.18**	**95.93**
DeepLab-v2M [9]	97.96	96.04	95.83

Resolution 896 × 512

	Mean Pixel Accuracy	Mean IOU	Frequency Weighted IOU
U-Net [31]	90.91	85.59	82.30
PSPNet [37]	94.69	90.42	89.98
DeepLab-v2S [8]	96.91	**94.26**	93.56
DeepLab-v2M [9]	**97.00**	94.15	**93.66**

Table 1 *Semantic segmentation results with traditional metrics - mean pixel accuracy, mean intersection-over-union and frequency weighted intersection-over-union. All reported results are in percentages.*

Additional smaller input images of DeepLab-v2M only detriment its performance compared to DeepLab-v2S, because bouys and other tiny obstacles disappear in the process of re-scalling. This is substantiated by a lower number of detections (shown in Table 2).

U-Net is very sensitive to reflections and sun-glitter in water, which causes a lot of false positive detections (Table 2), subsequently leading to a low segmentation accuracy. Based solely on given segmentation metrics and their results we cannot fully determine which method detects more obstacles and how well it approximates navigable surface.

5.2. Obstacle detection results

Table 2 summarizes results based on metrics used in [7]. On low-resolution images DeepLab-v2S approximates the water-edge the most accurately, fallowed by DeepLab-v2M, PSP-Net, ISSM and U-Net. DeepLab-v2S outperforms its multi-scale counterpart DeepLab-v2M by 3.6% on the water-edge estimation task. The highest F-measure score is achieved by PSP-Net, followed by DeepLab-v2S, ISSM, DeepLab-v2M and U-Net. PSP-Net outperforms second-best DeepLab-v2M by 14.1% on the obstacle detection task.

On high-resolution images DeepLab-v2M approximates the water-edge the most accurately, followed by DeepLab-v2S, ISSM, PSP-Net and U-Net in the order given. DeepLab-v2M outperforms its single-scale counterpart DeepLab-v2S by 2.2% on the water-edge estimation task. It also obtains

Baseline

	μ_{edg}	TP	FP	FN	F-measure
ISSM [5]	0.056 (0.066)	538	1641	144	0.376

Resolution 512×288

	μ_{edg}	TP	FP	FN	F-measure
U-Net [31]	0.098 (0.090)	296	2329	383	0.179
PSP-Net [37]	0.050 (0.063)	**322**	203	**357**	**0.535**
DeepLab-v2S [8]	**0.027** (0.035)	245	121	434	0.469
DeepLab-v2M [8]	0.028 (0.041)	121	**25**	558	0.293

Resolution 896×512

	μ_{edg}	TP	FP	FN	F-measure
U-Net [31]	0.128 (0.115)	153	4686	526	0.055
PSPNet [37]	0.073 (0.101)	318	**94**	361	0.583
DeepLab-v2S [8]	0.045 (0.065)	**388**	447	**291**	0.513
DeepLab-v2M [8]	**0.044** (0.058)	361	117	318	**0.624**

Table 2 *Modd2 [7] reports water-edge estimation error μ_{edg} and its standard deviation, the number of true positive (TP), false positive (FP), false negative (FN) detections and the F-measure.*

the highest F-measure score, followed by PSP-Net, DeepLab-v2S, ISSM and U-Net. DeepLab-v2M outperforms PSP-Net by approximately 7% on the task of obstacle detection.

In general DeepLab-v2 variations approximate the water-edge most accurately. The difference in the number of detections between DeepLab-v2S and DeepLab-v2M is significant, especially on low-resolution images, where multiple re-scalled inputs of DeepLab-v2M suppress small obstacles. This causes a reduction of true positive as well as false positive detections. The difference in the water-edge approximation is less significant, because the water edge does not disappear in the process of re-scaling. PSP-Net is able to detect a lot of true positives, yet it has problems with over- and under-estimating the water edge when overlooking the open sea. Similarly to U-Net, the ISSM method is also sensitive to sun-glitter and reflections, causing a considerable amount of false positive detections and poor water-edge approximation compared to DeepLab-v2, regardless of having an additional IMU sensor. ISSM detects significantly more true positives than any method from Section 3, but its high number of false positive detections deteriorates its overall F-measure score.

5.3. Qualitative comparison

In this section we present a qualitative comparison of methods from Section 3. We limit ourselves to the input resolution of 512×288, where the difference between single-scale and multi-scale version of DeepLab-v2 is most prominent. Figure 4 depicts segmentation performance in various challenging scenarios.

The first row in Figure 4 shows a problem of a small obstacle detection. DeepLab-v2S detects a smaller buoy, while its multi-scale version suppressed the detection. The water-edge is also better estimated in a single-scale version. The water-edge estimation of U-Net is severely over-estimated, however its sensitivity allows it to correctly detect the buoy. PSP-Net is unable to detect obstacles (boat and buoy) in the scene and it drastically under-estimates the water-edge. ISSM correctly detects all obstacles in the scene. These observations are reflected in quantitative results (Table 1,Table 2) as well.

The second row in Figure 4 portraits the difficulty of water segmentation in presence of significant sun-glitter. As stated in Section 5.1, U-Net and ISSM are sensitive to sun-glitter. This causes a lot of false positive detections and poor water-edge estimation. Most of the falsely classified patches are relatively large, which has a negative effect on a segmentation accuracy presented in Table 1. In general, PSP-Net and DeepLab-v2 do not have problems with sun-glitter which is reflected in segmentation (Table 1) and obstacle detection (Table 2) results.

The third row in Figure 4 depicts a challenge of detecting an obstacle (i.e., a green buoy) whose color resembles the surrounding water. U-Net detects only a top part of the obstacle, however the bottom is the more important part for safe navigation. It also drastically over-estimates the water edge, which has a significant negative impact on the segmentation accuracy. The obstacle in a scene is big enough to not get suppressed in a multi-scale version of DeepLab-v2. Moreover, the various scales of DeepLab-v2M allow it to refine the outline of an obstacle more precisely. PSP-Net does not detect obstacle at all and its water-edge approximation is severely over-estimated. ISSM approximates the water-edge the most precisely. It detects obstacle as a whole plus a part of its reflection in the water.

The last row in Figure 4 shows a scene in a harbour with water droplets on a camera lens. The water droplets were correctly ignored by all methods. They have also correctly estimated the water-edge, however none of the CNN methods was able to detect a pole in close proximity, which is a critically dangerous misclassification. On the other hand, ISSM is able to correctly detect the pole, but its water-edge estimation is affected by sun-glitter.

5.4. Speed analysis

The processing speed of methods, described in Section 3, is presented in Table 3. On low-resolution images U-Net is the fastest, followed by DeepLab-v2S, PSP-Net and DeepLab-v2M. Similarly, U-Net is also the fastest on high-resolution images, followed by PSP-Net, DeepLab-v2S and DeepLab-v2M.

U-Net is the fastest method due to its low-complexity architecture and fewer parameters compared to those of PSP-Net and DeepLab-v2. Both PSP-Net and DeepLab-v2 use ResNet backend architecture, however PSP-Net uses ResNet-50 architecture, while DeepLab-v2 uses ResNet-101 architecture. Besides this DeepLab-v2 also has a fully convolutional CRF layer, which explains the slower performance. Despite the segmentation of multi-scale images in DeepLab-v2M is done parallel, we witness a slow-down of approximately 50% compared to DeepLab-v2S. The ISSM method is the fastest, however its performance was measured on images of size 100×100.

The on-board cameras from Modd2 [7] are limited to 10 frames-per-second, meaning that all of the methods from Section 3 would be capable of running in real-time when inputted with low-resolution images. However, only U-Net and PSP-Net would be able to run at real-time when using high-resolution images.

Baseline

	t_{seg} [ms]	ω [fps]
ISSM [7]	33.8	29.6

Resolution 512×288

	t_{seg} [ms]	ω [fps]
U-Net [31]	**37.6**	**26.6**
PSPNet [37]	57.9	17.3
DeepLab-v2S [8]	48.5	20.6
DeepLab-v2M [8]	98.6	10.1

Resolution 896×512

	t_{seg} [ms]	ω [fps]
U-Net [31]	**93.2**	**10.7**
PSPNet [37]	98.1	10.2
DeepLab-v2S [8]	114.8	8.7
DeepLab-v2M [8]	218.7	4.6

Table 3 *Times required for single image segmentation, measured in milliseconds, is denoted with t_{seg}, while the corresponding frame-rate, measured in frames-per-second (fps), is denoted as ω.*

6. Conclusion

In this paper, we benchmarked three popular semantic segmentation methods on a marine environment and prepared an in-depth analysis of their performances. As expected, the results showed that complex networks are able to estimate the water-edge more accurately. DeepLab-v2 produced the most promising results for the task of water-edge estimation as well as for the obstacle detection task. This could be due to deeper backbone model (ResNet-101) compared to PSP-Net (ResNet-50). U-Net performed the worst, which could be a consequence of training it from scratch.

On the task of water-edge approximation CNN methods, described in Section 3, mostly over-estimate the water-edge location. In contrast, non-CNN ISSM does not over-estimate the water-edge location due to embedded IMU sensor, which serves for horizon calculation and segmentation restriction. Nevertheless, due to its sensitivity to sun-glitter, it under-estimates the water-edge location in special cases. This reduces the potential navigable surface, but it does not cause dangerous instances. On the task of obstacle detection, certain obstacles, which visual appearance is similar to water, remain undetected in all compared methods. Detection of buoys far away also proved to be difficult, but such misclassification do not pose an immediate danger to USVs. False positive detections are mainly caused by reflections and prominent sun-glitter.

When processing low-resolution images, all methods are capable of running in real-time. However, low-resolution images also produce low F-measure scores. When processing high-resolution images, presented CNN methods achieve higher F-measure scores due to mostly larger number of true positive detections. DeepLab-v2 cannot run in real-time when processing high-resolution images, while other methods are on the verge of running in real-time.

In our future work, we plan a deeper analysis of tested methods, accompanied by additional state-of-the-art segmentation methods. For a fair comparison we plan on re-train all methods on the same

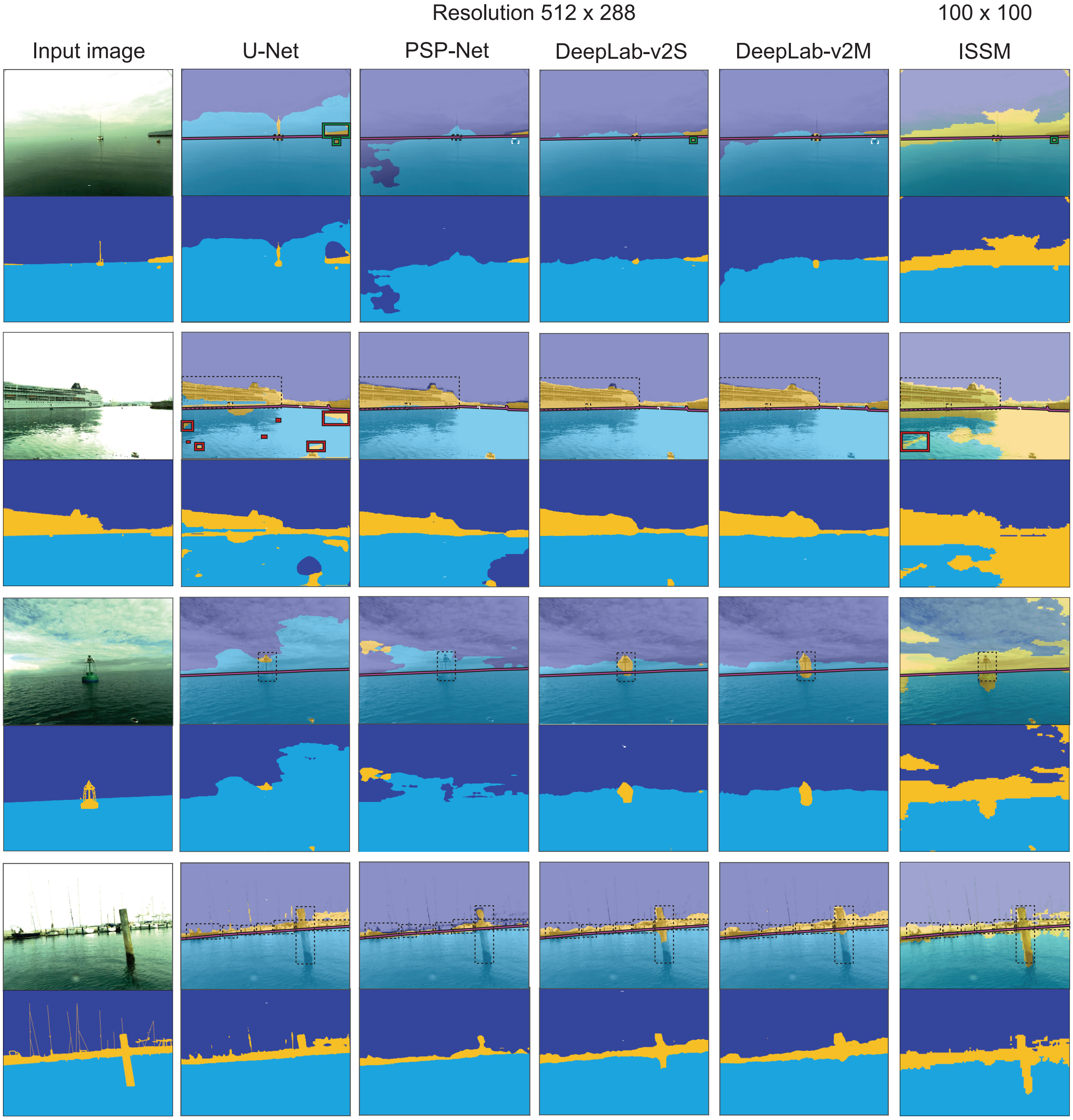

Figure 4 *Qualitative comparison of methods for resolution* 512 × 288*. The sky, obstacles and water components are denoted with deep-blue, yellow and cyan color, respectively. The ground truth sea edge is annotated with a pink line, while ground truth obstacles are outlined with a dotted bounding box. False positives are marked with a red bounding box, whereas correctly detected obstacles are marked with a green bounding box.*

dataset and use a significantly larger training set for fine-tunning. We will explore a new evaluation metrics, specifically designed for a marine environment, which takes into account the size of obstacles and their distances from the USV. We also plan to experiment with optimization of the segmentation process and embedding different sensor modalities into deep-learning segmentation algorithms.

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 ViAMaRo J2-8175.

References

[1] M. Abadi et al. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[2] C. Almeida, T. Franco, H. Ferreira, A. Martins, R. Santos, J. M. Almeida, J. Carvalho, and E. Silva. Radar based collision detection developments on USV ROAZ II. In *OCEANS - EU*, pages 1–6, May 2009.

[3] J. M. Alvarez, T. Gevers, Y. LeCun, and A. M. Lopez. Road scene segmentation from a single image. In *European Conference on Computer Vision*, pages 376–389. Springer, 2012.

[4] J. M. Alvarez, Y. LeCun, T. Gevers, and A. M. Lopez. Semantic road segmentation via multi-scale ensembles of learned features. In *European Conference on Computer Vision*, pages 586–595. Springer, 2012.

[5] B. Bovcon and M. Kristan. Obstacle detection for usvs by joint stereo-view semantic segmentation. 2018.

[6] B. Bovcon, R. Mandeljc, J. Perš, and M. Kristan. Improving vision-based obstacle detection on USV using inertial sensor. In *ISPA*, pages 1–6, Sept 2017.

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

[8] 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 TPAMI*, 40(4):834–848, 2018.

[9] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. *arXiv preprint arXiv:1802.02611*, 2018.

[10] 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*, pages 248–255. Ieee, 2009.

[11] L. Elkins, D. Sellers, and W. R. Monach. The autonomous maritime navigation (AMN) project: Field tests, autonomous and cooperative behaviors, data fusion, sensors, and vehicles. *Journal of Field Robotics*, 27(6):790–818, 2010.

[12] Y. Guo, M. Romero, S. H. Ieng, F. Plumet, R. Benosman, and B. Gas. Reactive path planning for autonomous sailboat using an omni-directional camera for obstacle detection. In *ICM*, pages 445–450, 2011.

[13] Z. Guo, H. Shengoku, G. Wu, Q. Chen, W. Yuan, X. Shi, X. Shao, Y. Xu, and R. Shibasaki. Semantic segmentation for urban planning maps based on U-Net. In *IGARSS 2018-2018 IEEE International Geoscience and Remote Sensing Symposium*, pages 6187–6190. IEEE, 2018.

[14] 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*, pages 770–778, 2016.

[15] H. K. Heidarsson and G. S. Sukhatme. Obstacle detection and avoidance for an autonomous surface vehicle using a profiling sonar. In *ICRA 2011*, pages 731–736, May 2011.

[16] C. Y. Jeong, H. S. Yang, and K. D. Moon. Horizon detection in maritime images using scene parsing network. *Electronics Letters*, 54(12):760–762, 2018.

[17] M. Kristan, V. S. Kenk, S. Kovačič, and J. Perš. Fast image-based obstacle detection from unmanned surface vehicles. *IEEE TCYB*, 46(3):641–654, 2016.

[18] J. Larson, M. Bruch, R. Halterman, J. Rogers, and R. Webster. Advances in autonomous obstacle avoidance for unmanned surface vehicles. Technical report, SPAWAR San Diego, 2007.

[19] 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.

[20] D. Levi, N. Garnett, E. Fetaya, and I. Herzlyia. Stixelnet: A deep convolutional network for obstacle detection and road segmentation. In *BMVC*, pages 109–1, 2015.

[21] J. Li, X. Liang, S. Shen, T. Xu, J. Feng, and S. Yan. Scale-aware fast R-CNN for pedestrian detection. *IEEE Transactions on Multimedia*, 20(4):985–996, 2018.

[22] L. Liu and Y. Zhou. A closer look at U-Net for road detection. In *ICDIP 2018*, volume 10806, page 108061I. International Society for Optics and Photonics, 2018.

[23] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015.

[24] G. L. Oliveira, W. Burgard, and T. Brox. Efficient deep models for monocular road segmentation. In *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*, pages 4885–4891. IEEE, 2016.

[25] C. Onunka and G. Bright. Autonomous marine craft navigation: On the study of radar obstacle detection. In *ICCAR 2010*, pages 567–572, Dec 2010.

[26] P. Paccaud and D. Barry. Obstacle detection for lake-deployed autonomous surface vehicles using RGB imagery. *PloS one*, 13(10):e0205319, 2018.

[27] 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*, pages 1–16, 2018.

[28] 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*, 18(8):1993–2016, 2017.

[29] E. Reinhard, M. Adhikhmin, B. Gooch, and P. Shirley. Color transfer between images. *IEEE Computer graphics and applications*, 21(5):34–41, 2001.

[30] 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*, pages 91–99, 2015.

[31] O. Ronneberger, P. Fischer, and T. Brox. U-Net: Convolutional networks for biomedical image segmentation. In *MICCAI*, pages 234–241. Springer, 2015.

[32] M. Siam, M. Gamal, M. Abdel-Razek, S. Yogamani, M. Jagersand, H. Zhang, N. Vallurupalli, S. Annamaneni, G. Varma, C. Jawahar, et al. A comparative study of real-time semantic segmentation for autonomous driving. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 587–597, 2018.

[33] D. Socek, D. Culibrk, O. Marques, H. Kalva, and B. Furht. A hybrid color-based foreground object detection method for automated marine surveillance. *LNCS*, 3708:340, 2005.

[34] H. Wang and Z. Wei. Stereovision based obstacle detection system for unmanned surface vehicle. In *ROBIO*, pages 917–921, 2013.

[35] H. Wang, Z. Wei, C. S. Ow, K. T. Ho, B. Feng, and J. Huang. Improvement in real-time obstacle detection system for USV. In *ICARCV*, pages 1317–1322, 2012.

[36] W. Xia, Z. Chen, Y. Zhang, and J. Liu. An approach for road material identification by dual-stage convolutional networks. In *IGARSS 2018 - 2018 IEEE International Geoscience and Remote Sensing Symposium*, pages 7153–7156, July 2018.

[37] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia. Pyramid scene parsing network. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 2881–2890, 2017.

[38] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba. Semantic understanding of scenes through the ADE20k dataset. *arXiv preprint arXiv:1608.05442*, 2016.

[39] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba. Scene parsing through ADE20K dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.