A Survey on the Deployability of Semantic Segmentation Networks for Fluvial Navigation

Reeve Lambert, Jianwen Li, Jalil Chavez-Galaviz, Nina Mahmoudian

Purdue University

{lamber53, li3602, jchavezg, ninam}@purdue.edu

Abstract

Neural network semantic image segmentation has developed into a powerful tool for autonomous navigational environmental comprehension in complex environments. While semantic segmentation networks have seen ample applications in the ground domain, implementations in the surface water domain, especially fluvial (rivers and streams) deployments, have lagged behind due to training data and literature sparsity issues. To tackle this problem the publicly available River Obstacle Segmentation En-Route By USV Dataset (ROSEBUD) was recently published. The dataset provides unique rural fluvial training data for the water binary segmentation task to aid in fluvial scene autonomous navigation. Despite new dataset sources, there is still a need for studies on networks that excel at both understanding marine and fluvial scenes and efficiently operating on the computationally limited embedded systems that are common on autonomous marine platforms like ASVs. To provide insight into state-of-the-art network capabilities on embedded systems a survey of twelve networks encompassing 8 different architectures has been developed. Networks were trained and tested on a combination of three existing datasets, including the ROSEBUD dataset, and then implemented on an NVIDIA Jetson Nano to evaluate performance on real-world hardware. The survey's results lay out recommendations for networks to use in autonomous applications in complex and fast-moving environments relative to network performance and inference speed.

1. Introduction

Autonomous Surface Vehicles (ASV) have shown to be exceptionally versatile and have rapidly advanced the scientific communities' understanding of natural and man-made phenomena in the world's oceans and lakes. With the ability to affordably, rapidly and efficiently collect data, scientists have been able to complete more in depth research in oceanography [1, 2], bathymetry [3], magnetic surveys [4], and environmental phenomena [5].

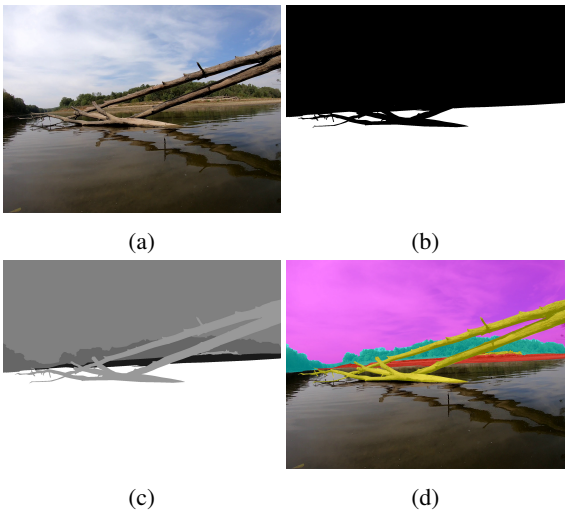


Figure 1: An example image and subsequent annotations pulled from the ROSEBUD dataset showing a) the original RGB image, b) the hand annotation of the binary classification between water and non-water classes, c) the hand annotation of fluvial classes, d) a colored overlay of the annotated fluvial classes onto the original image.

Recent advances in neural networks have further advanced the capabilities of ASVs by providing new methods of data driven model predictive controllers that allow precise vehicle control [6, 7], vehicle dynamics identification [8,9], environmental perception for navigation [10–12], and identification of obstacles for avoidance [13, 14]. The abilities of complex scene understanding lent to ASVs equipped with neural networks have advanced their deployment opportunities for tasks in environments previously thought of as beyond the scope of ASVs.

ASVs have seen limited use in fluvial environments (rivers, streams, and creeks) and estuaries due to naviga-

tion and control complexities arising from the complex environment (Fig. 1) and dynamic time-varying disturbances present in flowing water. While ASVs such as the Jetyak [15, 16] and ROAZ [3] have shown the ability to operate in rivers, they have only seen limited deployments [3, 17, 18] that leverage a-prioris of the operational environment. In fluvial environments, a-prioris can easily be invalidated due to time varying obstacle dispersion, course changes due to river morphology, dynamic disturbances due to flowing water, and rapid changes in operational depth.

For successful fluvial deployments, a system should be able to identify navigable areas without relying on environmental a-prioris. This becomes a problem for navigational systems that would need to generalize between fluvial and marine systems due to the vast differences in navigable space, obstacles density, water clarity, and other environmental factors. The generalized navigation problem can be simplified by grouping all obstacles (shore, rocks, logs, etc.) and other non-water entities (trees, sky, etc.) into a single non-water category. When this is done the navigational problem becomes one of semantic segmentation network water identification in the environment from ASV onboard imagery.

While the image segmentation task is simplified when formulated as a binary problem the generalization problem still persists as water can appear vastly different in varying lights, environments, and seasons. Semantic segmentation neural networks are supervised learning methods that learn to classify human annotated training data. Such implementations of supervised perception networks are exceptionally dependent on the availability of training data obtained from the targeted deployment environment, due to their inability to rapidly generalize between dissimilar data. While fluctuations in lighting and coloration can be augmented to aid in network generalization, environmental fundamentals can not be easily synthesized. Thus, the need for training and testing on annotated data in varying environments is required to successfully train a network for real-world implementation. Semantically annotated datasets for unmanned ground and aerial applications are publicly available and abundant in number and size across many differing environments [19–24]. However, few publicly available marine datasets exist, leading to data sparsity problems. Of the existing datasets many are not semantically annotated and even fewer contain semantically annotated images of fluvial scenes.

This lack of high quality training data has led to a dearth in real-world deployments and forward work towards deploying semantic segmentation networks on ASVs for navigation of complex environments. This latter results in a lack of knowledge within the marine robotics community regarding networks that are: 1) implementable on embedded computers typically used in robotic systems, 2) fast enough to allow real-world deployments, and 3) capable of learning complex datasets comprising fluvial, marine, and inland waterways. To alleviate this problem, the authors recently have created and published the publicly available River Obstacle Segmentation En-route By USV Dataset (ROSEBUD) [25] that contains images of fluvial scenes from the Wabash River and Sugar Creek in the U.S. state of Indiana, Fig. 1.

Within this work an evaluation of twelve semantic segmentation network architecture and encoder backbone combinations are trained and evaluated on a dataset comprised of existing semantically annotated marine images [10], inland lakes and canals [26], and fluvial scenes [25]. Results are presented for network capabilities on the representative datasets as well as performance on an embedded system. This paper specifically reports on: 1) how well varying state of the art semantic architectures and encoder-pairs handle marine semantic data; and 2) the performance of the survey networks on a resource constrained NVIDIA Jetson Nano embedded system commonly used in robotic development and deployments of deep learning algorithms.

In the remainder of this work, a review of existing marine datasets for navigational tasks and semantic segmentation networks is presented in Sec. 2. A brief review of the datasets being used for training and testing of said networks are described in Sec. 3, while the network architectures and encoders utilized within this work are explained in Sec. 4. The network training process and evaluation metrics are discussed in Sec. 5 then the results of network testing are presented in Sec. 6. A final conclusion and future implementation work is explored in Sec. 7.

2. Background

Semantic Segmentation is frequently used to provide contextualized location information from RGB based sensing modalities (stereo and monocular RGB and RGB-D). Unlike traditional neural network approaches for image identification, semantic segmentation not only identify classes within an image as a whole, but identifies where within an image classes are. Semantic segmentation labels individual pixels within an image using supervised neural networks such as Convolutional Neural Networks (CNN) or Fully Convolutional Networks (FCN). These networks utilize an encoder and decoder combination where an encoder performs repeated convolutions into a feature rich latent space, and a decoder upsamples the representative latent space into a classified output at the same resolution of the input image. Semantic segmentation networks have been investigated and developed previously for uses ranging from the marine navigation task in shipping lanes [11], to indoor scene recognition [27]. These capabilities have been employed together with simultaneous mapping and localization (SLAM) to create semantic maps that provide anno-

tated data clouds for autonomous navigation and decision making while simultaneously allowing human robotic insight [28]. Furthermore, none of the existing implementations the authors are aware of are explicitly done on embedded systems meant for operation on mobile marine platforms. This work investigates the application of supervised segmentation to the fluvial navigation task through the use of existing and established networks in the literature.

Within this work networks have been selected that use various encoders, decoders, and network architectures to reduce the spatial losses from feature encoding that are common in semantic segmentation processes. This is because the act of reducing spatial resolution to increase the depth of features inherently leads to a loss of knowledge of the spatial location of the features at the input resolution.For example, Residual Networks [29] have residual outputs that are used as inputs in subsequent convolutional layers. Unet [30] is a special architecture that takes convolutional layers and saves the feature space at the resolution of each layer for incorporation with decoding deconvolutional layers of comparable resolution. Deeplabv3+ [31] uses atrous (or dilated) convolutions along with atrous spatial pyramid pooling to extract and decode features at different spatial strides to incorporate various spatial resolutions in the process.

ESPNet [32] implements a particular convolution factorization module called Efficient Spatial Pyramid (ESP), which is based on point-wise convolution and spatial pyramid pooling of dilated convolutions. The ESP custom module is used to build the entire network structure, reducing the memory footprint and increasing execution speed. The Bilateral Segmentation Network (BiSeNet) is a dual path network that consists of a low-stride spatial path to preserve spatial information and a context path with fast downsampling to improve the receptive field. In the end, the two paths are merged efficiently with a feature fusion module, thus allowing a good balance between speed and segmentation performance. SegNet [33] consists of an encoder-decoder core architecture using 13 identical convolutional layers (convolution, batch normalization and pooling) from the VGG16 network as the encoder, followed by a decoder composed of another 13 convolutional layers (convolution, batch normalization, and pooling) to upsample the encoder feature space to full input resolution; finally, the decoder output is fed into a multi-class softmax classifier to do pixel-wise classification.

A survey of semantic segmentation networks and their applicability to marine surveillance using datasets was presented by Cane Et. Al [34], specifically the identification of objects in marine images from the ground and air in the Seagull [35], and SMD [36] datasets. This work turns such a survey on its head by investigating network ability to recognize only water in a binary classification schema for navigation instead of obstacle classification for surveillance.

Furthermore since 2018, new network architectures and semantically annotated datasets have become publicly available for the semantic classification task. Examples include the MaSTr1325 [10], Tampere-WaterSeg [37], Waterline [38], Modd2 [39], Seagull [35] and SMD [36] datasets. However, only the first three contain publicly available ground truth pixel-wise annotations explicitly meant for aiding in surface navigation through semantic network training. Furthermore, none of the aforementioned datasets contain surface level imagery annotated for fluvial navigation. To the authors knowledge, [25] is one of the only publicly available semantically annotated fluvial segmentation datasets built to aid in non-urban fluvial navigation.

3. Dataset

To capture the wide variety of images from an ASV operating in the presence of different obstacles, and especially in fluvial environments; a dataset was built consisting of annotations and images from the ROSEBUD [25], MaSTr1325 [10], and Tampere-WaterSeg [26, 37] datasets. The dataset combination provides a diverse spectrum of scenarios such as littoral, harbor, lake, canal, and fluvial imagery. The environmental spectrum provides a variety of scenes for training and represents many of the scenes an ASV may encounter on varying rivers such as harbors, rural creeks with heavy debris fields, urban rivers with minimal debris and estruaries.

While the Tapere-Waterseg dataset and ROSEBUD datasets already have binary masks for supervised training, the MaSTr1325 annotations are multi-class; thus the annotations were converted to binary masks by combining all non-water classes into a single ”non-water” class. All three dataset together create 2474 unique images for use. In order to assess the performance of the networks, the dataset has been randomly divided into three subsets: training, validation, and testing. The training represents 70 % (1732 images) of the data, validation 20 % (495 images), and testing 10 % (247 images). Across the training subset the split between MaSTr1325, Tampereseg, and ROSEBUD was 54.2 - 23.5 - 22.3%, with the validation and testing split being 52.9 - 25.3 - 21.8% and 50.6 - 27.1 - 22.3% respectively. The images were then augmented as in [40] with each image being augmented for color, orientation, exposure, and blur for a total of 13 augmentations. This brings the total number of images to 34636, with 24248, 6930, 3458 for training, validation, and testing respectively.

4. Segmentation Networks

To tackle the binary semantic segmentation task within the training and test dataset within this work, eight different semantic segmentation architectures are utilized with three

of the eight architectures being employed with three different encoder architectures for latent space generation. The eight semantic architectures used are: 1) DeepLabV3+ [31], 2) Unet [30], 3) PSPNet [41], 4) WaSR [42,43], 5) WODIS [12], 6) BiSeNet [44], 7) SegNet [33], 8) ESPNet [32].

WODIS was modified from that reported in [12], to adapt the network to the binary segmentation task at hand. This was done by using Binary Cross Entropy (BCE) loss instead of negative log likelihood loss and by passing the network output through a sigmoid function rather than a softmax. A custom dataloader for the binary dataset was also created and used for training and testing of WODIS and all other networks. Similar changes were made to the WaSR network; however, as shown in Sec. 6, the FPS for this network is not reported since it was not implementable on the NVIDIA Jetson Nano due to its high memory footprint requirements.

Of the eight architectures DeppLabv3+, Unet, and PSPNet were implemented multiple times with different encoders for latent space generation. ResNet was chosen for its frequent use within the literature, and ability to use residual layers to maintain feature and spatial resolutions. Two ResNet implementations [29], ResNet 101 and ResNet 50, were used to test the affect of encoder depth on the performance of the network. Finally EfficientNet [45] was also implemented as a encoder build to balance size and computational cost with accuracy.

Each of the seven architecture-encoder combinations were implemented using the Segmentation Models Pytorch repository [46]. While the implementations of BiSeNet was from [47], SegNet [33], ESPNet from [32], WODIS from the link in [12], and WaSR from the link present in [42]. Thus, the total of twelve implementations were: 1) DeepLabV3+ - ResNet101, 2) DeepLabV3+ - ResNet50, 3) Unet - ResNet101, 4) Unet - ResNet50, 5) Unet - EfficientNet-b4, 6) PSPNet - ResNet101, 7) PSPNet - ResNet50, 8) WaSR, 9) WODIS, 10) BiSeNet, 11) SegNet, 12) ESPNet.

5. Network Training and Evaluation

All twelve of the networks detailed in Sec. 4 were trained on desktop computers running NVIDIA 3000 series GPU's. BiSeNet, SegNet, ESPNet, and all networks that utilize ResNet101 and EfficientNet as encoders were trained on a NVIDIA RTX3090 with 24 GB of VRAM and clock speed of 1.4GHz. WODIS and all networks that utilized ResNet50 as an encoder were trained on a NVIDIA RTX3080 with 8 GB of VRAM and a clock speed of 1.4 GHz.

All Unet, DeepLabV3+,and PSPNet networks were trained for 30 epochs with a learning rate of $1e-4$, Dice coefficient loss [48], and early stopping implemented with respect to validation water Intersection Over Union (IOU) given as $IOU = \frac{TP}{TP+FP+FN}$ where TP, FP, and FN are the standard classification of network inferences of True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) with respect to the networks binary classification of water. The early stopping had an early stopping patience of seven epochs, and delta of $5e-4$. WaSR was implemented in a similar fashion, but with the custom loss function detailed in [42] across both the water, and resulting non-water class.

WODIS was trained over 30 epochs with a learning rate of $1e-4$ and BCE loss [49] but without early stopping. BiSeNet, SegNet, WaSR, and ESPNet were all also implemented with BCE loss and without early stopping criteria and trained for 1, 50, 10, and $10+10$ (encoder, decoder) epochs respectively. Across BiSeNet, SegNet, WaSR, and ESPNet the learning rates were also different as $5e-3$, $1e-3$, $1e-6$, and $5e-4$ with exponential weight decay of $1e-4$, $1e-8$, $1e-6$, and $5e-4$. BiSeNet used an SGD optimizer [50] while others used the Adam optimizer [51].

To evaluate how well the networks handled the test dataset four macro metrics are reported: 1) Mean Pixel Accuracy (MPA), 2) Mean Intersection Over Union (MIOU), F1 Score (or Dice coeffecient), and the Specificity (Spec), using the following definitions: $MPA = \frac{TP+TN}{TP+TN+FP+FN}$, $MIOU = \frac{1}{n}\sum_1^n IOU_n$ where n is the number of classes (in this case two), $Spec = \frac{TN}{TN+FP}$, $F1Score = \frac{Pr \times Se}{Pr+Se}$ where precision (pr) and sensitivity (se) are defined as $pr = \frac{TP}{TP+FP}$ and $se = \frac{TP}{TP+TN}$.

6. Results

The evaluating metrics of the training as discussed in Sec. 5 are presented as two separate sections to show the quantitative and qualitative results separately in Sec. 6.1 and Sec. 6.2. The quantitative results section discusses the metrics reported from testing of trained networks and the implementation speed on the embedded platform NVIDIA Jetson Nano that features a Quad-core ARM A57 @ 1.43 GHz with 4 GB memory and a NVIDIA Maxwell GPU that allows efficient implementation of neural networks. The qualitative results section presents a visual comparison of the network output masks to complement the results obtained in the quantitative section. The results are presented and discussed separately to provide a macro (dataset) and a micro (image) view of network performance. Table 1 shows that all networks except BiSeNet and SegNet have MPA, MIOU, F1 Score, and Spe over 0.96.

6.1. Quantitative results

In general, all of the networks performed exceptionally well at a macro dataset level. This is in part due to the complexity and size of the multi-class networks being used for binary segmentation tasks as well as the significant amount of augmentations and training data utilized.

Network	MPA	MIOU	F1 Score	Spe	Avg. FPS	Size [MB]	Parameter [Million]
Unet ResNet50	0.9942	0.9881	0.9930	0.9943	1.17	124.27	32.5
Unet ResNet101	0.9948	0.9893	0.9938	0.9954	0.87	196.92	51.5
Unet EffecientNet4	0.9867	0.9730	0.9842	0.9821	1.05	77.64	20.2
Deeplabv3+ ResNet50	0.9942	0.9881	0.9931	0.9944	1.21	101.99	26.7
Deeplabv3+ ResNet101	0.9942	0.9882	0.9931	0.9943	0.88	174.64	45.7
PSPNet ResNet50	0.9930	0.9857	0.9916	0.9949	3.42	92.91	24.3
PSPNet ResNet101	0.9930	0.9856	0.9916	0.9937	3.39	165.56	43.3
WODIS	0.9923	0.9843	0.9908	0.9914	1.41	187.5	49.0
BiSeNet	0.9730	0.9377	0.9679	0.9692	0.74	20.01	5.2
SegNet	0.9741	0.9382	0.9681	0.9937	0.43	117.9	29.4
ESPNet	0.9840	0.9706	0.9804	0.9863	18.04	15	0.35
WaSR	0.9957	0.9949	0.9962	0.9899	-	272.89	71.4

Table 1: Quantitative comparison of networks after completing training on the combined dataset as detailed in Section 3. Additionally, each network was implemented on the Nvidia Jetson Nano and executed to evaluate the speed in FPS, and the memory footprint in terms of the network size and number of parameters.

BiSeNet and SegNet have a relatively low F1 Score of around 0.93 but still have high MPA, MIOU and Spe. The result also shows that besides the WaSR network all of the networks can be implemented on the embedded NVIDIA Jetson Nano system at varying inference speeds. WaSR was not able to be implemented due to the limited memory present on the embedded NVIDIA Jetson system. Most networks implemented within this survey have a FPS below 1.5 due to their size and complexity. PSPNet with ResNet 50 and 101 both have a performance over 3 FPS. The lightweight ESPNet architecture has the best inference performance with an operation rate of 18.04 images per second on the embedded system. However, lightweight architectures are not always fast in every implementation as BiSeNet, a lightweight network, has a slower inference rate than even larger networks such as WODIS and PSPNet, and was still less accurate than ESPNet.

The results show that network size and complexity for accuracy is not the end goal for autonomous deployments on embedded systems. This shows up in the data as an example of diminishing returns, where for the data presented increasing the network size from 101.99 MB (DeeplabV3+ ResNet50) to 174.65 MB (DeeplabV3+ ResNet101) did not yield a significant increase in network performance, while decreasing the systems average inference speed by 0.33 to less than a frame per second. Such decreases in inference speed can be detrimental to autonomous system control structures operating at speeds at or greater than 10Hz in dynamic environments such as rivers more so than a small decrease in output accuracy.

All of the Unet, Deeplab, and PSPNet implementations had exceptionally high F1 scores as they were all trained with dice coefficient loss and with early stopping criteria relative to their F1 score on the validation dataset. Finally, as expected with images such as those present in complex scenes as shown in Fig. 1, all of the networks struggled with MIOU, likely due to the complex entanglement between water and object reflections and water when present behind obstacles (Fig. 2f right) as well as submerged objects (Fig. 2h center).

(a) Unet ResNet50

(b) Unet ResNet101

(c) Unet EffecientNet4

(d) Deeplabv3+ ResNet50

(e) Deeplabv3+ ResNet101

(f) PSPNet ResNet50

(g) PSPNet ResNet101

(h) WODIS

(i) BiSeNet

(j) SegNet

(k) ESPNet

(l) WaSR

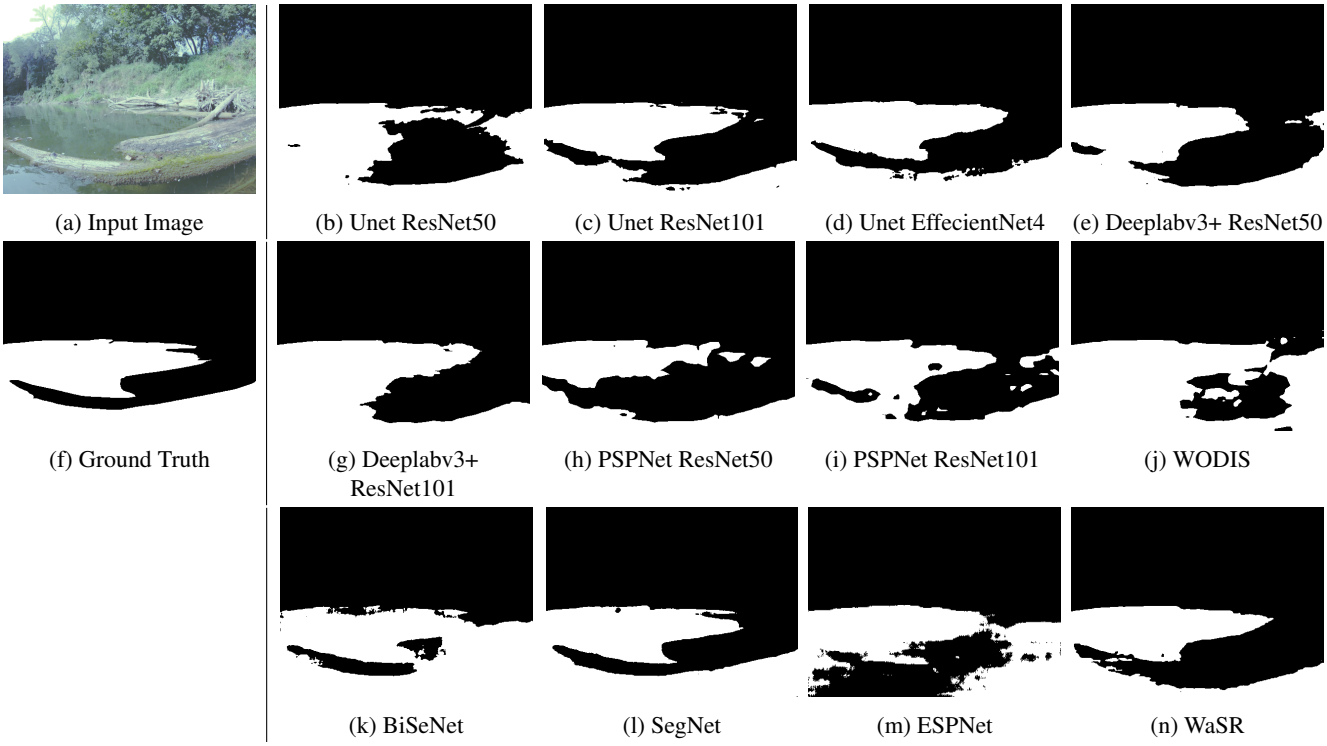
Figure 2: Illustration of network outputs on ROSEBUD dataset [40]. Columns from left to right are the test image, the ground truth mask, and the output of the network.

6.2. Qualitative results

The macro level results show that the networks perform exceptionally well at identifying scenes and many qualitative outputs such as that shown in Fig. 2d which illustrates the networks' performance in complex scenes such as images with fallen trees in a river system. However, the networks occasionally struggle with complex fluvial scenes as shown in Fig. 2. As many obstacles present within fluvial environments, especially natural and rural areas, are small in size compared to the environment but can be detrimental to ASV navigation and vehicle safety. The ability of a network to recognize small obstacles such as thin branches, partially submerged rocks, and stumps is hard to gauge from dataset wide quantitative results.

An example of network difficulties that does not show up in macro data is shown in Fig. 2b. In the figure an image is taken of a dry riverbed, that the Unet ResNet101 fails to properly parse into water and non-water areas. Such an encounter is not out of the question when traversing remote rivers with seasonally varying river depths. In other instances such as that shown in Fig. 2i, reflections in the water from the sky can create confusion for networks, in this case the BiSeNet network.

Despite these issues, the networks can perform well in circumstances with great variance in exposure and image hue, such as that shown in Fig. 2j, Fig. 2l, and 2e. This is due to the ability to augment image hue and brightness to enhance the training data, thus future data sets aimed at improving navigation should focus on varying environments



(a) Input Image (b) Unet ResNet50 (c) Unet ResNet101 (d) Unet EffecientNet4 (e) Deeplabv3+ ResNet50

(f) Ground Truth (g) Deeplabv3+ ResNet101 (h) PSPNet ResNet50 (i) PSPNet ResNet101 (j) WODIS

(k) BiSeNet (l) SegNet (m) ESPNet (n) WaSR

Figure 3: Illustration of network outputs on a same image from ROSEBUD dataset [40]

rather than varying seasons.

The qualitative comparison in Fig. 3 shows the performance of the networks when segmenting a complex scene containing water, trees, sky, and a partially submerged log. The proximity between the camera and obstacle ridden foreground also creates difficulty due to the cameras lens distortions. WaSR, Segnet, Unet ResNet101, and Unet EffecientNet produced masks close to the ground truth. PSPNet Resnet50 detected the submerged part of the log even though it is marked as water in the ground truth. Although this classification is a FN detection, it gives a higher factor of safety for the navigation task due to the high specificity of the image. Unet ResNet50, Deeplabv3+, PSPNet Resnet101, WODIS, and BiseNet were prone to poor detection of the log above water. ESPNet not only failed to detect half of the log, but also made abundant FP predictions.

7. Conclusion

In this paper, a survey of twelve networks was performed with training on a combination of datasets including ROSEBUD, MaSTr1325, and Tampere-WaterSeg to provide qualitative and quantitative results for each network. Additionally, all the networks were implemented on an NVIDIA Jetson Nano to evaluate the performance on real-world hardware.

Since all the evaluated networks exhibited good results in terms of MPA, MIOU, and F1 score, and Specificity. Determining the best network depends on other aspects such as the environment and the computational resources available. Although the application in general is fluvial navigation, the environment plays an important role in the selection of the most suitable network. If the segmentation network is going to be used in a dynamic environment with a lot of different obstacles, then a good trade-off between speed and accuracy is needed, which in this survey was the case for PSPNet50 with 3.42 FPS and an MIOU of 0.9857. If on the other hand, the environment requires a fast reaction due to currents or wind, then the obvious option is ESPNet which reached 18.04 FPS and an MIOU of 0.9706.

In the future, the selected networks will be integrated into the navigation stack of a real ASV platform and tested on different rivers to evaluate their performance. The newly collected data will be used to assess the performance of the networks in an unseen environment. The results from these experiments will serve to determine what properties of the selected networks are more appropriate to have in real-world applications.

References

[1] C. Sabine, A. Sutton, K. McCabe, N. Lawrence-Slavas, S. Alin, R. Feely, R. Jenkins, S. Maenner, C. Meinig, J. Thomas, E. van Ooijen, A. Passmore, and B. Tilbrook, "Evaluation of a New Carbon Dioxide System for Autonomous Surface Vehicles," *Journal of Atmospheric and Oceanic Technology*, vol. 37, no. 8, pp. 1305–1317, 07 2020. [Online]. Available: https://doi.org/10.1175/JTECH-D-20-0010.1

[2] C. Meinig, N. Lawrence-Slavas, R. Jenkins, and H. M. Tabisola, "The use of saildrones to examine spring conditions in the bering sea: Vehicle specification and mission performance," in *OCEANS 2015 - MTS/IEEE Washington*, 2015, pp. 1–6.

[3] H. Ferreira, C. Almeida, A. Martins, J. Almeida, N. Dias, A. Dias, and E. Silva, "Autonomous bathymetry for risk assessment with roaz robotic surface vehicle," in *OCEANS 2009-EUROPE*, 2009, pp. 1–6.

[4] M. Kurowski, J. Thal, R. Damerius, H. Korte, and T. Jeinsch, "Automated survey in very shallow water using an unmanned surface vehicle," *IFAC-PapersOnLine*, vol. 52, no. 21, pp. 146–151, 2019. [Online]. Available: https://doi.org/10.1016/j.ifacol.2019.12.298

[5] M. Bălănescu, G. Suciu, A. Bădicu, A. Birdici, A. Pasat, C. Poenaru, and I. Zătreanu, "Study on unmanned surface vehicles used for environmental monitoring in fragile ecosystems," in *2020 IEEE 26th International Symposium for Design and Technology in Electronic Packaging (SIITME)*, 2020, pp. 94–97.

[6] G. Lv, Z. Peng, L. Liu, and J. Wang, "Barrier-certified distributed model predictive control of under-actuated autonomous surface vehicles via neurodynamic optimization," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, pp. 1–13, 2022.

[7] J. Wang, J. Wang, and Q.-L. Han, "Neurodynamics-based model predictive control of continuous-time under-actuated mechatronic systems," *IEEE/ASME Transactions on Mechatronics*, vol. 26, no. 1, pp. 311–322, 2021.

[8] J. Woo, J. Park, C. Yu, and N. Kim, "Dynamic model identification of unmanned surface vehicles using deep learning network," *Applied Ocean Research*, vol. 78, pp. 123–133, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0141118717304819

[9] P.-F. Xu, C.-B. Han, H.-X. Cheng, C. Cheng, and T. Ge, "A physics-informed neural network for the prediction of unmanned surface vehicle dynamics," *Journal of Marine Science and Engineering*, vol. 10, no. 2, p. 148, Jan. 2022. [Online]. Available: https://doi.org/10.3390/jmse10020148

[10] B. Bovcon, J. Muhovic, J. Pers, 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, Nov. 2019. [Online]. Available: https://doi.org/10.1109/iros40897.2019.8967909

[11] D. Qiao, G. Liu, W. Li, T. Lyu, and J. Zhang, "Automated full scene parsing for marine ASVs using monocular vision," *Journal of Intelligent &amp Robotic Systems*, vol. 104, no. 2, Feb. 2022. [Online]. Available: https://doi.org/10.1007/s10846-021-01543-7

[12] X. Chen, Y. Liu, and K. Achuthan, "WODIS: Water obstacle detection network based on image segmentation for autonomous surface vehicles in maritime environments," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–13, 2021. [Online]. Available: https://doi.org/10.1109/tim.2021.3092070

[13] J. Woo and N. Kim, "Collision avoidance for an unmanned surface vehicle using deep reinforcement learning," *Ocean Engineering*, vol. 199, p. 107001, Mar. 2020. [Online]. Available: https://doi.org/10.1016/j.oceaneng.2020.107001

[14] C. Zhou, Y. Wang, L. Wang, and H. He, "Obstacle avoidance strategy for an autonomous surface vessel based on modified deep deterministic policy gradient," *Ocean Engineering*, vol. 243, p. 110166, Jan. 2022. [Online]. Available: https://doi.org/10.1016/j.oceaneng.2021.110166

[15] P. Kimball, J. Bailey, S. Das, R. Geyer, T. Harrison, C. Kunz, K. Manganini, K. Mankoff, K. Samuelson, T. Sayre-McCord, F. Straneo, P. Traykovski, and H. Singh, "The whoi jetyak: An autonomous surface vehicle for oceanographic research in shallow or dangerous waters," in *2014 IEEE/OES Autonomous Underwater Vehicles (AUV)*, 2014, pp. 1–7.

[16] J. Moulton, N. Karapetyan, S. Bukhsbaum, C. McKinney, S. Malebary, G. Sophocleous, A. Q. Li, and I. Rekleitis, "An autonomous surface vehicle for long term operations," in *OCEANS 2018 MTS/IEEE Charleston*, 2018, pp. 1–10.

[17] J. Moulton, N. Karapetyan, M. Kalaitzakis, A. Quattrini Li, N. Vitzilaios, and I. Rekleitis, "Dynamic autonomous surface vehicle controls under changing environmental forces," in *Field and Service Robotics*, G. Ishigami and K. Yoshida, Eds. Singapore: Springer Singapore, 2021, pp. 381–394.

[18] N. Karapetyan, J. Moulton, and I. Rekleitis, "Meander-based river coverage by an autonomous surface vehicle," in *Field and Service Robotics*, G. Ishigami and K. Yoshida, Eds. Singapore: Springer Singapore, 2021, pp. 353–364.

[19] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[20] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 3354–3361.

[21] F. Yu, H. Chen, X. Wang, W. Xian, Y. Chen, F. Liu, V. Madhavan, and T. Darrell, "Bdd100k: A diverse driving dataset for heterogeneous multitask learning," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[22] A. G. Perera, Y. Wei Law, and J. Chahl, "Uav-gesture: A dataset for uav control and gesture recognition," in *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, September 2018.

[23] B. Kellenberger, D. Marcos, and D. Tuia, "Detecting mammals in UAV images: Best practices to address a substantially imbalanced dataset with deep learning," *Remote Sensing of Environment*, vol. 216, pp. 139–153, Oct. 2018. [Online]. Available: https://doi.org/10.1016/j.rse.2018.06.028

[24] Z. Wang, L.-F. Wu, and N. Mahmoudian, "Aerial fluvial image dataset (afid) for semantic segmentation," Jul 2022. [Online]. Available: https://purr.purdue.edu/publications/4105/1

[25] R. D. Lambert, J. Li, J. F. C. Galaviz, Z. Wang, and N. Mahmoudian, "River obstacle segmentation enroute by usv dataset (rosebud)," Jun 2022. [Online]. Available: https://purr.purdue.edu/publications/4072/1

[26] J. Taipalmaa, "Tampere-waterseg," http://urn.fi/urn:nbn:fi:att:eafdb99c-4396-4591-80e0-24219875b5b6, 10 2019, jussi Taipalmaa.

[27] D. Seichter, M. Köhler, B. Lewandowski, T. Wengefeld, and H.-M. Gross, "Efficient rgb-d semantic segmentation for indoor scene analysis," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 13 525–13 531.

[28] I. Kostavelis and A. Gasteratos, "Semantic mapping for mobile robotics tasks: A survey," *Robotics and Autonomous Systems*, vol. 66, pp. 86–103, 2015.

[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] 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.

[31] 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.

[32] 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.

[33] 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.

[34] 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, Nov. 2018. [Online]. Available: https://doi.org/10.1109/avss.2018.8639077

[35] R. Ribeiro. Seagull dataset – VisLab – computer and robot vision laboratory. [Online]. Available: https://vislab.isr.tecnico.ulisboa.pt/seagull-dataset/

[36] 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.

[37] J. Taipalmaa, N. Passalis, H. Zhang, M. Gabbouj, and J. Raitoharju, "High-resolution water segmentation for autonomous unmanned surface vehicles: a novel dataset and evaluation," in *2019 IEEE 29th International Workshop on Machine Learning for Signal Processing (MLSP)*. IEEE, Oct. 2019. [Online]. Available: https://doi.org/10.1109/mlsp.2019.8918694

[38] 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, Feb. 2020. [Online]. Available: https://doi.org/10.1016/j.robot.2019.103346

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

[40] R. Lambert, J. Chavez-Galaviz, J. Li, and N. Mahmoudian, "ROSEBUD: A deep fluvial segmentation dataset for monocular vision-based river navigation and obstacle avoidance," *Sensors*, vol. 22, no. 13, p. 4681, Jun. 2022. [Online]. Available: https://doi.org/10.3390/s22134681

[41] 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 (CVPR)*, July 2017.

[42] B. Bovcon and M. Kristan, "WaSR–a water segmentation and refinement maritime obstacle detection network," *IEEE Transactions on Cybernetics*, pp. 1–14, 2021. [Online]. Available: https://doi.org/10.1109/tcyb.2021.3085856

[43] L. Žust and M. Kristan, "Learning maritime obstacle detection from weak annotations by scaffolding," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2022, pp. 955–964.

[44] 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.

[45] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *International conference on machine learning*. PMLR, 2019, pp. 6105–6114.

[46] P. Iakubovskii, "Segmentation models pytorch," https://github.com/qubvel/segmentation_models.pytorch, 2019.

[47] CoinCheung, "Bisenet," https://github.com/CoinCheung/BiSeNet, 2018.

[48] C. H. Sudre, W. Li, T. Vercauteren, S. Ourselin, and M. Jorge Cardoso, "Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations," in *Deep learning in medical image analysis and multimodal learning for clinical decision support*. Springer, 2017, pp. 240–248.

[49] P. Contributors, "Bceloss." [Online]. Available: https://pytorch.org/docs/stable/generated/torch.nn.BCELoss.html

[50] ——, "Sgd." [Online]. Available: https://pytorch.org/docs/stable/generated/torch.optim.SGD.html?highlight=sgd#torch.optim.SGD

[51] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.