ShorelineNet: An Efficient Deep Learning Approach for Shoreline Semantic Segmentation for Unmanned Surface Vehicles

Linghong Yao[1], Dimitrios Kanoulas[2], Ze Ji[3] and Yuanchang Liu[1,*]

***Abstract*— This paper introduces a novel deep learning approach to semantic segmentation of the shoreline environments with a high frames-per-second (fps) performance, making the approach readily applicable to autonomous navigation for Unmanned Surface Vehicles (USV). The proposed ShorelineNet is an efficient deep neural network of high performance relying only on visual input. ShorelineNet uses monocular visual input to produce accurate shoreline separation and obstacle detection compared to the state-of-the-art, and achieves this with real-time performance. Experimental validation on a challenging multi-modal maritime obstacle detection dataset, the MODD2 dataset, achieves a much faster inference (25fps on an NVIDIA Tesla K80 and 6fps on a CPU) with respect to the recent state-of-the-art methods, while keeping the performance equally high (73.1% F-score). This makes ShorelineNet a robust and effective model to be used for reliable USV navigation that require real-time and high-performance semantic segmentation of maritime environments.**

I. INTRODUCTION

Unmanned Surface Vehicles (USV) are intelligent marine platforms vital in various practical applications, such as maritime environment monitoring [1] and coastal waters patrolling [2]. Compared to large-scale manned vessels, the improved mobility and portability of USVs allow them to autonomously inspect challenging areas that would be otherwise difficult to reach. Typical small-scale USVs are several meters long (under 5 meters for most platforms), equipped with on-board video cameras as their main perception sensing system, and supported by small computing units [3].

Safe and efficient autonomous operations are required for full autonomy of USVs. To achieve this, USVs should be capable of observing the environment, detecting obstacles, and navigating around them to avoid collisions in real-time. In the past, several studies have explored the use of various sensors for observing the environment, such as marine radar, LIDAR, and sonar [4]. However, these sensors may have several disadvantages, such as limited detection accuracy, impaired capability in detecting submerged obstacles, and high prices especially for LiDAR sensing. By far, visual cameras have shown to be the most effective, information-dense, and affordable sensory modules for USVs [5].

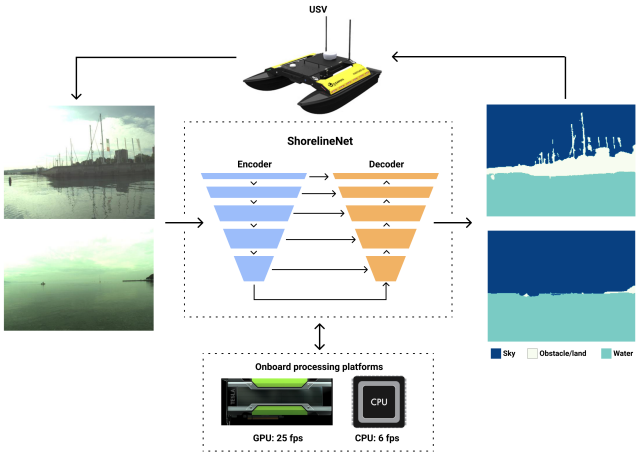


Fig. 1. The proposed ShorelineNet structure. ShorelineNet takes in images acquired in various conditions and outputs predicted masks for segmentation and obstacle detection. A real-time performance of ShorelineNet can be well achieved on both GPU (25fps) and CPU (6fps).

Vision-based obstacle detection in shoreline scenes poses several challenges, namely the non-flat water-sky separation line, large variation in water patterns, and significantly different scene appearances in varying weather and lighting conditions. Traditional camera-based obstacle detection utilizes background subtraction methods. However, they are unsuitable for USV settings due to the large scene variation and high dynamics of marine environments [6]. Stereo reconstruction methods [7] are able to reliably detect objects that protrude the water surface, but fail to detect obstacles submerged underwater. In addition, such methods are range-constrained and are not able to detect visible obstacles in faraway distances. Structured models, such as semantic segmentation methods (SSM) [5], have demonstrated acceptable results with real-time inference. It utilizes a monocular camera's image input and a Markov random field to segment scenes into sky, land, and ocean. However, since only simple features are extracted from the image, the method's performance is bottlenecked in environments with harsh lighting conditions, high degree of visual ambiguity (such as haze), and presence of high pitching and rolling.

There are currently two state-of-the-art approaches to overcome the shortcomings of traditional algorithms. One approach is to increase the number of sensors and inputs. Bovcon et al. [8] proposed an inertial measurement unit

[1]Linghong Yao and Yuanchang Liu are with the Department of Mechanical Engineering, University College London (UCL), WC1E 7JE, United Kingdom. leon.yao.18@ucl.ac.uk, yuanchang.liu@ucl.ac.uk

[2]Dimitrios Kanoulas is with the Department of Computer Science, University College London (UCL), WC1E 6BT, United Kingdom. d.kanoulas@ucl.ac.uk

[3]Ze Ji is with the School of Engineering, Cardiff University, Cardiff, UK. jiz1@cardiff.ac.uk

∗ Corresponding author: Yuanchang Liu

(IMU) assisted semantic segmentation method (ISSM) that uses a stereo camera and an IMU sensor to achieve a significant boost in the detection performance (F-score). In particular, false positive results were significantly reduced from stereo verification. However, this approach imposes the necessity to add additional sensors and a calibration procedure, and doubles the computational time required per image.

The second approach is to learn richer features using deep convolutional neural networks, inspired by the recent developments in deep learning. These deep neural networks are able to learn rich features and achieve desirable vision-based semantic segmentation results [9]. Bovcon et al. [10] proposed a decoder-encoder network and achieved the state-of-the-art performance on the public marine environment Multi-modal Marine Obstacle Detection Dataset 2 (MODD2) dataset [11]. However, a significant drawback of those segmentation networks is the long inference time. For example, within the network proposed in [10], the adopted backbone encoder is predominantly ResNet101, which is a relatively heavy encoder with large number of parameters and long inference time. Consequently, despite of the improvement in accuracy, ResNet-based segmentation neural networks also require high computational time, making them impractical to be implemented for real-world USV applications.

In this paper, we propose a novel efficient segmentation network (shown in Fig. 1), named ShorelineNet, to improve the shortcomings of the aforementioned approaches. ShorelineNet is capable of achieving real-time semantic segmentation of shoreline environments, using only visual inputs, while maintaining high performance. Our main contributions include: 1) demonstrating that the proposed ShorelineNet is a new real-time network that produces significantly higher accuracy compared to model-based real-time methods, and 2) showing that the improvement in speed is achieved with very little compromises in accuracy by retaining a high F-score compared to other state-of-the-art neural network approaches. We validate our proposed ShorelineNet on the public MODD2 marine environment dataset and demonstrate that segmentation provided by ShorelineNet can be performed in real-time with 25fps on a GPU (an NVIDIA Tesla K80) and 6fps on a single-core CPU, which enables a real-time post-processing and facilities full autonomy for USVs.

The rest of the paper is organised as follows. We describe the model architecture of ShorelineNet and its implementation in Sec. II. Experimental validations of the ShorelineNet are carried out in Sec. III together with detailed comparative results with other state-of-the-art methods. Sec. IV concludes the paper and provide directions for future work.

II. METHODOLOGY

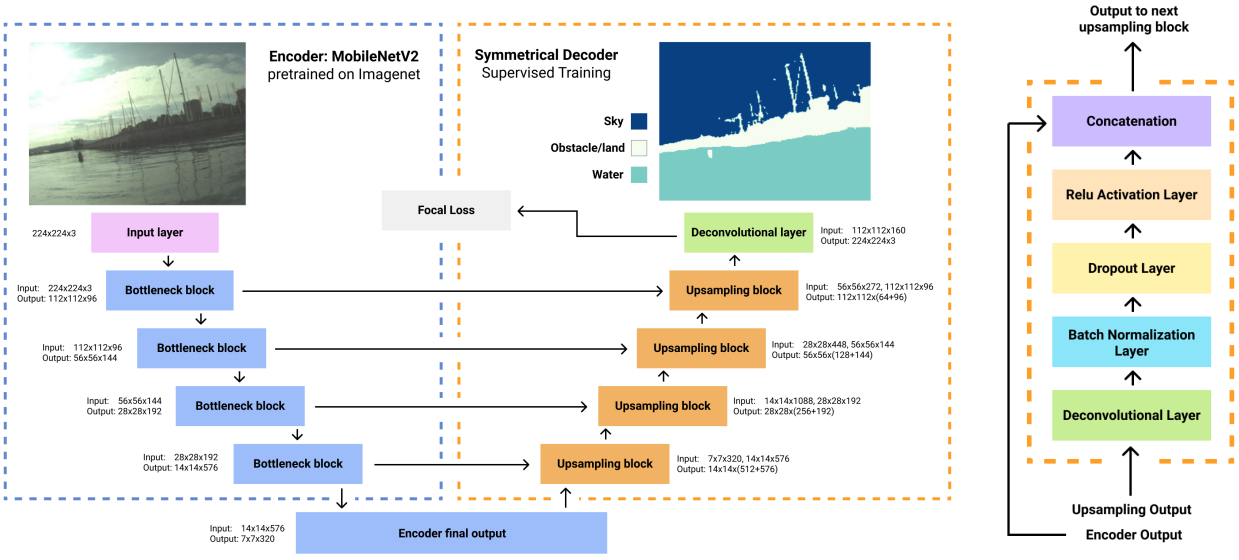
A. Problem Formulation

ShorelineNet aims to semantically segmenting image per pixel to three general classes, i.e. sky, obstacles/land, and water. This segmentation subsequently contributes to accurate shoreline separation and obstacle detection that enables full autonomous navigation of USVs in complex environments. The general goal of semantic segmentation is the minimization of per-pixel difference between the ground truth mask and the predicted mask. However, within the context of USVs, this is an inappropriate problem framing, as inaccuracies of even just a few pixels can lead to false positive or false negative detection of small obstacles. Extensive false positive predictions will lead to frequent misdirection of a USV; whereas a large number of false negatives will lead to the collision of a USV with obstacles. ShorelineNet is therefore developed to specifically tackle the problem of obstacle detection in shoreline scenes, and to overcome the shortcomings of the traditional semantic segmentation problem framing.

B. ShorelineNet Architecture

The proposed model adopts a UNet-like architecture with symmetrical encoder and decoder (shown in Fig. 2a). UNet is an image segmentation network that is originally designed for biomedical image segmentation, but has since shown to be an effective model for many other segmentation tasks [12]. Several desirable characteristics of UNet make it feasible for a fast and accurate detector of the maritime environment. First, UNet has a lightweight structure compared to other main-stream networks, and is therefore associated with low inference time to output fast predictions. Second, UNet has a high performance in segmentation tasks, especially with small numbers of classes. Such property makes UNet extremely suitable for segmenting the shoreline scene, where all scenes can be segmented into three classes in a top-to-bottom order: sky, obstacles/land, and water. Lastly, the connections between the outputs of the encoder to the decoder makes the network capable of retaining deep features, hence suitable when there are small number of training samples. This is desirable for the maritime dataset where the amount of publicly available data is limited. Adopted from UNet's architecture, the proposed ShorelineNet is less likely to overfit on the MaSTr1325 dataset [13], which is designed specifically for segmentation tasks of small-sized coastal USVs, but only includes 1325 images.

We constructed ShorelineNet with a contracting encoder path and an expanding decoder path, shown in Fig. 2a. The purpose of the encoder is to extract deep features from the image using convolutional layers. This can produce rich features and dense spatial information that cannot be obtained using model-based algorithms, such as SSM [5] and background subtraction [6]. A suitable CNN would be the one which extracts the highest quality features with the fewest number of parameters. We have selected the MobileNetV2 [14] as the encoder for the ShorelineNet, as we find it to encompass both of these qualities. When compared to an equivalent UNet structure using ResNet101 as the backbone similar to the WaSR, and using the network for inference on the MaSTr1325 dataset [9] with 224×224 image inputs, ShorelineNet demonstrates an 8-fold reduction in the number of parameters, as shown in Table I. This increases the GPU inference speed by 43 percent and more than doubles



(a) (b)

Fig. 2. (a) Architecture of ShorelineNet adopts a UNet-like structure, the encoder (left, blue) is a pre-trained MobileNetV2 model, and the decoder (right, orange) is a custom decoder constructed with upsampling blocks symmetrical to the encoder outputs. (b) A detailed schematic of the upsampling block is shown, the block first scales up the output of the previous upsampling block, and then concatenates with the outputs from the encoder.

the CPU inference time. In addition, it also reduces the memory space required to store these networks. As we will show in Sec. III, these benefits are achieved with very little compromise in performance.

TABLE I
ARCHITECTURE PARAMETERS AND RUNTIME COMPARISON

	ResNet101+Unet	ShorelineNet (ours)
Encoder Parameters	42.66M	2.26M
Decoder Parameters	4.23M	4.23M
Total Parameters	46.29M	6.50M
GPU Runtime (fps)	18.0	25.7
CPU Runtime (fps)	2.2	5.8

The decoder is composed of four upsampling blocks and a deconvolution layer with spatial symmetry to the encoder outputs, shown in Fig. 2b. Each upsampling block is composed with a deconvolution layer followed by a batch normalization layer, a dropout layer, and a ReLu activation layer. The output of the activation layer is then concatenated along the last channel with the corresponding output from the encoder. Following the advice in [12], we connect the outputs of each block in MobileNetV2 to the decoder, which allows the rich features extracted from each block of the encoder to be transferred to the decoder without extra computation; whereas without these connections, some features would be lost in the pooling layers of the network. It should be noted that such interactive connections make the network both more robust and efficient. The input and outputs of each upsampling block is shown in Fig. 2a.

The final output of the network is $224 \times 224 \times 3$ in size, and we reduce it to a single channel for each pixel by selecting the class that the network assigns the highest probability for. It is important to note that dropout layers are crucial components in our network design (shown in Fig. 2b) as they reduce overfitting by forcing the network to utilize on a wider set of neurons rather than relying on a few important ones. Experiments (Sec. III) showed that dropout layers significantly reduce the number of false positive detections in form of noise around the shoreline edge.

C. Implementation

A main challenge in training on a small dataset is the prevention of overfitting. We train the network on the MaSTr1325 dataset [13], which contains 1325 semantically segmented images of shoreline scenes. The results are post processed and evaluated on the MODD2 dataset [8], which is the most challenging marine dataset publicly available, and allows us to compare to the state-of-the-art methods. Heavy image augmentation is performed in parallel to training in order to prevent overfitting. Our adopted augmentations include randomly flipping the images horizontally, rotating by up to 15 degrees, scaling by 60 to 90 percent, and color changes in hue, saturation, brightness, and contrast.

Our experiments have shown that traditional cross entropy function is inadequate for the task of obstacle detection as it does not produce high penalty for noisy false positive predictions near the edge of the water or between the border of two segments. In addition, the shoreline scene is usually skewed in class distribution where there are more sky and water pixels than obstacle pixels (Fig. 3). This limitation effectively generated high number of false positive predictions near the shoreline and noisy predictions around obstacles. To accommodate this limitation and increase the reliability of the ShorelineNet in detecting obstacles near water and obstacle edges, a new loss function based on the

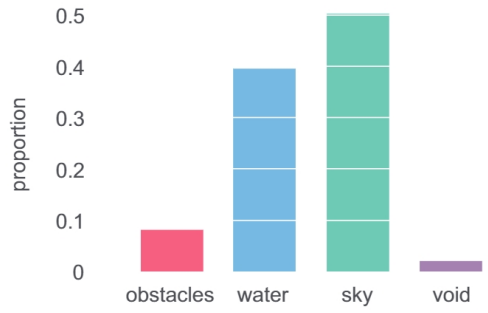


Fig. 3. The data distribution of the MaSTr1325 dataset. [9]

focal cross entropy loss function has been adopted in our model training as:

$$FL(p_t) = -\alpha_t(1-p_t)^{\gamma} log(p_t), \tag{1}$$

where $p_t \in [0,1]$ describes the probability that the model assigns to a specific pixel, γ is a modulating parameter, and α is a balancing parameter.

Focal loss extends the range at which low loss is assigned to easy examples, and penalizes the model for wrong predictions with high confidence scores. This effectively forces the model to distinguish the pixel differences between water and object, and minimizes false predictions of obstacles in water areas. We use $\gamma = 2, \alpha_t = 0.25$ in practice, adopted from [15]. By putting a higher penalty on falsely predicted pixels, the new focal loss helps the model to predict the water edge more accurately and also reduces the number of false positive predictions.

The encoder is a MobileNetV2 model pretrained on ImageNet. The model architecture is implemented and all training is done in Tensorflow using the Keras API. The decoder weights in each of the upsampling blocks are initialized using the Glorot normal initializer, which has shown to be especially effective for gradient descent of deconvolution layers. The network is trained using Google Colab's NVIDIA Tesla K80 GPU.

III. EXPERIMENTAL VALIDATION

A. Dataset and Baseline

There is a limited number of publicly available marine-time datasets that are fully semantically segmented for every pixel. The Marine Semantic Segmentation Dataset [9] (MaSTr1325) is a recent high-quality dataset designed specifically for training neural networks for semantic segmentation and has produced promising results with state-of-the-art neural networks. We use the MaSTr1325 to train our network, with heavy image augmentation described in Sec. II to increase variance in the dataset.

The trained network is then evaluated on the Multimodal Marine Obstacle Detection Dataset (MODD2), which consists of 28 continuous video sequences with labelled shoreline and bounding boxes around obstacles. However, since this dataset is not annotated with a per-pixel mask, post processing is required to transform the annotation masks into the shoreline and the bounding boxes that indicate where the network predicted obstacles.

TABLE II
OBSTACLE THRESHOLD COMPARISON

Architecture	μ_{edge}	TP	FP	FN	F-score
ShorelineNet	15.3	2711	**1505**	**490**	**73.1**
ShorelineNet (low threshold)	15.3	**4597**	2446	1720	68.8

The post processing procedures follow the methodology used in [10] for a consistency in comparison. From the pixel-wise masks outputted by the network, the largest connected component within the water region is treated as the navigable water surface, and the upper boundary of the region corresponds to the shoreline. The obstacles are obtained by marking bounding boxes around regions that are predicted as obstacles within the water region. The performance is calculated with the F-score (also known as F1 [16], [17]) defined as follows:

$$F_1 = 2 \times \frac{precision + recall}{precision \times recall}, \tag{2}$$

$$precision = \frac{TP}{TP+FP}, \qquad recall = \frac{TP}{TP+FN} \tag{3}$$

where TP are the true positives, FP the false positives, and FN the false negatives. Note that in order to speed up the inference time, the input for the ShorelineNet (224×224 pixels) is smaller than that of the original image (1278×958 pixels). Such reduction in image resolution imposes a bigger challenge on the task, as some small obstacles may only appear in a few pixels in size. For fair comparison, an object size threshold has been used in proportion to our input space by removing a small set of annotations, such that no obstacle will appear smaller than 2×2 pixels in size(the same obstacle size used in [9] and [10]). Note that we also show in Table II that a high F-score can also be achieved by ShorelineNet without increasing this threshold with the details discussed in following sub-sections.

The experiments were undertaken in three steps: 1) we set our baseline network as the ShorelineNet architecture without the dropout layer and train the baseline model using sparse cross entropy loss, 2) we then train a network with dropout layers and with focal loss, respectively, to observe what kind of improvements each provides, and 3) we train a network by applying both dropout layers and utilizing focal loss to achieve the best results. Despite differences in accuracy, all of these networks have the same inference speed and network size.

The results from ShorelineNet are then compared to the state-of-the-art methods, such as ISSM [8] and WaSR [10], where we show that the high speed performance of ShorelineNet does not compromise its accuracy.

B. Experiment results and discussions

The experiments results are shown in Table III. For all our experiments, fully converged results are presented to reveal

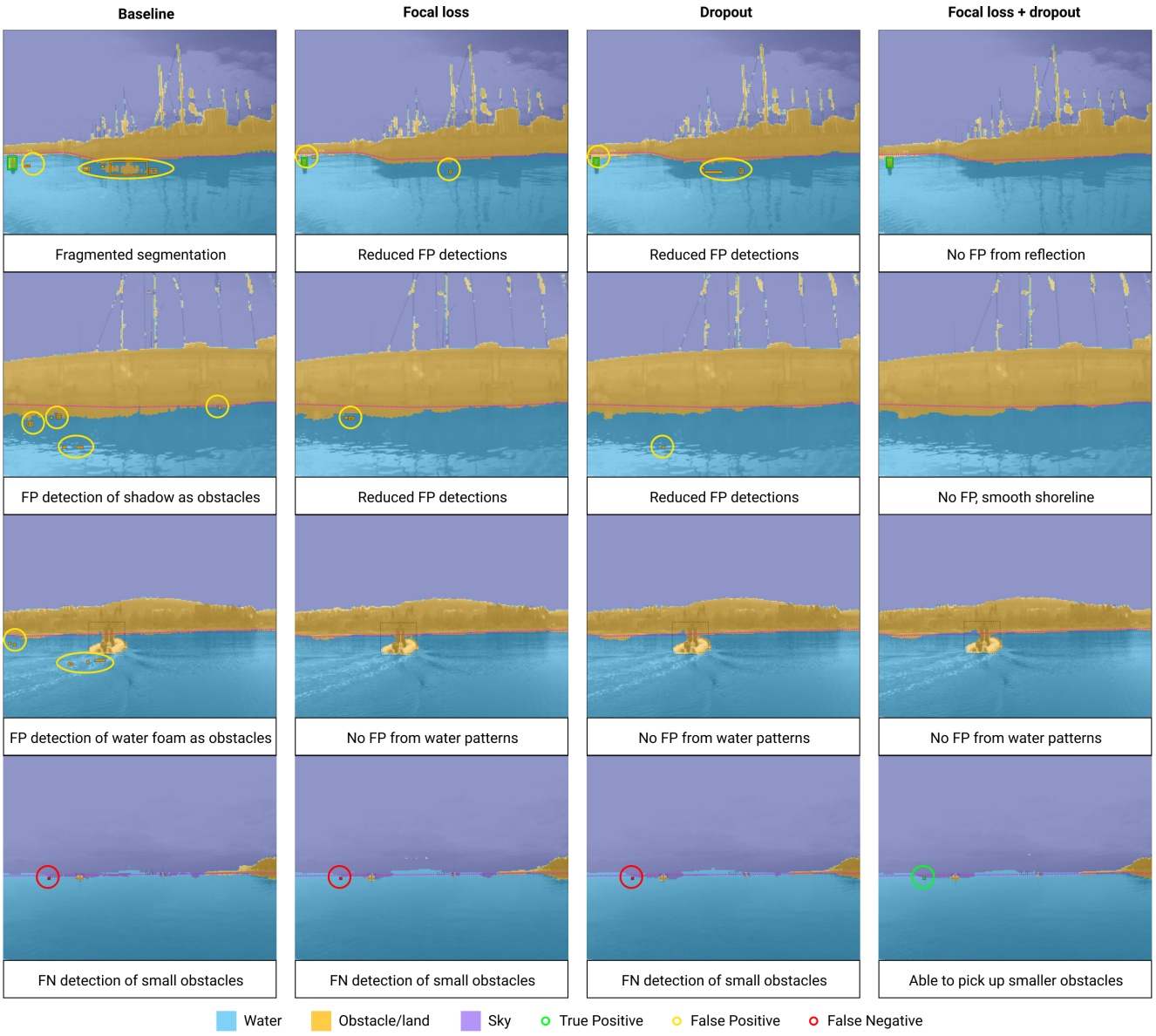


Fig. 4. Qualitative inspection of the results from each experiments. In column one, we see that the baseline network outputs highly fragmented segmentation and an unacceptable amount of false positive predictions. In column two and three, where focal loss and dropout layers are applied respectively, we see a decrease of the false positive predictions present in these images. This indicates that focal loss and dropout layers have made the network become more robust against reflections and glitters. Lastly, the fourth column shows the best result of ShorelineNet with very few false positive predictions and smoother water edges.

TABLE III
EXPERIMENTS WITH DROPOUT AND FOCAL LOSS

Architecture	μ_{edge}	TP	FP	FN	F-score
Baseline	15.2	**2826**	3849	**358**	57.3
Dropout	15.8	2439	**1392**	735	69.5
Focal loss	17.3	2480	2039	704	64.4
Dropout + Focal loss	15.3	2711	1505	490	**73.1**

the best performance. In addition, since both dropout layers and focal loss increase the time it takes for the model to converge, the number of epochs trained for each experiment is different.

Our experiments revealed that dropout layers are extremely effective in reducing the number of false positive results, increasing the F-score from 57 percent to 69 percent compared to the baseline (shown in Table III). Specifically, when implemented with dropout layers, the model reduced false positive predictions by three-fold due to the capability of dropout layers to effectively enable the network to utilize all the neurons. As a consequence of this, the implementation of dropout layers makes ShorelineNet not lenient on a small set of connections that achieve high accuracy in the training set. On the other side, the network learns a larger set of deep features which is more likely to produce generalized results

TABLE IV

COMPARISON TO STATE-OF-THE-ART MODEL-BASED METHODS

Architecture	μ_{edge}	F-score	fps (CPU)	fps (GPU)
ISSM [5]	52.8	34.7	**29**	NA
ISMM (stereo) [8]	52.8	49.5	11	NA
ShorelineNet (ours)	**15.3**	**73.1**	6	**25**

TABLE V

COMPARISON TO STATE-OF-THE-ART NEURAL NETWORK METHODS

Architecture	μ_{edge}	TP	FP	FN	F-score	fps (GPU)
WaSR [10]	9.2	6166	679	151	93.7	10
WaSR (no WS) [10]	12.3	4149	710	2168	74.2	10
PSPNet [18]	13.8	5886	4359	431	71.7	17
DLv2s [19]	14.1	5834	227	483	75.2	1.6
ShorelineNet (ours)	15.3	2711	1505	490	73.1	**25**

applicable to data beyond the training set.

In Table III, we also show that focal loss achieves large improvement when compared to the standard per-pixel cross entropy loss. Similar to dropout layers, focal loss significantly reduces the number of false positive predictions and increased F-score by 7 percent from the baseline. This is achieved by forcing the network to prioritise feature learning of difficult obstacles, as well as falsely predicted obstacles from glitter and reflections.

The best result is achieved when both dropout layers and focal loss are used. We observe an 16 percent increase in F-score from the baseline from the last row in Table III, with a very low number of false positive examples. In comparison to the baseline network, the selected network is much more prone to overfitting, indicated by the significant reduction in false positive values while maintaining similar true positive and false negative scores. We note that the reduction in true positive generally represent temporal inconsistencies in detection of small and faraway obstacles. Overall, such a configuration achieves the most robust and reliable performance whilst retaining a real-time performance.

Qualitative inspection of the results is illustrated in Fig. 4. We observe that without focal loss or dropout layers, the baseline model produces very fragmented segmentation of the image. Specifically, there are a high number of false positive predictions near the shoreline boundary, object boundaries, and reflections and glitters. Focal loss and dropout layers illustrate similar qualitative improvements when implemented. We can see from the first and second row in Fig. 4 that there is a reduction in false positive predictions due to water reflections when focal loss and dropout layers are applied. In both of these experiments, the network is much more robust compared to the baseline and is less likely to predict shadows and reflections as objects. The best result is achieved when both focal loss and dropout layers are applied. We can see in the last column in Fig. 4 that the network predicts smoother water edges with very little noises presented. In addition, we also observed that ShorelineNet is able to reliably detect even extremely small obstacles, which can only be achieved when both focal loss and dropout layers are applied, as shown in last row of Fig. 4.

Recall that in Table II, we show that even with lower threshold for obstacle size, ShorelineNet still achieves a high 68.8 percent F-score. More specifically, we lower the obstacle size threshold to the same threshold used in WaSR [10], which is performed on an input space with higher resolution. Even though ShorelineNet should be severely disadvantaged in such a setting as some obstacles are less than 2×2 pixels in the input space, the network still achieves a high performance nonetheless. This well demonstrates that ShorelineNet is extremely robust, and is able to pick up an additional 1800 true positive detection while outputting an high F-score still comparable to state-of-the-arts results.

C. Comparative study with state-of-the-art model-based methods

In Table IV, we compare the results of ShorelineNet with ISSM and stereo ISSM [8], as these methods are the state-of-the-art real-time efficient models for detecting obstacles in shoreline scenes. We show that ShorelineNet achieves a much higher F-score when compared to both ISSM and stereo ISSM, as well as a much higher water edge accuracy μ_{edge} (15.3 for ShorelineNet and 52.8 for ISSM). Whilst ShorelineNet's CPU fps (6) is lower, its GPU fps (25) has the same performance to that of the ISSM (29fps on CPU), and is much higher than ISSM with stereo verification (11fps on CPU). Furthermore, both ISSM and stereo ISSM rely on additional sensors such as as IMU and stereo camera, whilst ShorelineNet operates only from monocular input images making ShorelineNet much more efficient at converting sensor inputs into accurate detection.

D. Comparative study with the state-of-the-art neural network methods

In Table V, we present our comparative analysis with the state-of-the-art segmentation methods, including the WaSR network [10], the WaSR network without water seperation loss (WaSR without WS) [10], the PSPNet [18] and the DLv2s [19]. The following results can be obtained. First, when compared with PSPNet, WaSR without WS and DLv2s [19], ShorelineNet achieves a similar accuracy but with a superior advantage in speed. For example, a 25fps can be achieved on a GPU using ShorelineNet compared to 1.62fps, 10fps and 17fps achieved by DLv2s, WaSR without WS and PSPNet, respectively. In particular, ShorelineNet's inference time is computed on a NVIDIA Tesla K80 GPU whilst other networks' inference times are computed on a NVIDIA GTX 1080 Ti GPU with values taken from [9], [10]. Note that Tesla K80 is a slightly slower GPU for neural network inferences, which further proves the high inference efficiency achieved by ShorelineNet.

When compared with the WaSR network, even though ShorelineNet provides a compromised performance in detection accuracy with a lower F-score and water edge accuracy, an evident improvement in speed is achieved, where ShorelineNet is able to operate with a 2.5 times higher

speed making a real-time operation possible. In addition, we argue that a compromised performance on the standard evaluation metric used in [8], [9], [10] does not indicate inferiority of the ShorelineNet for the following reasons. First, as ShorelineNet's input (224×224 pixels) is lower than that of WaSR (384×512 pixels), when the network's output is scaled up using bilinear interpolation in the post processing stage, the value of μ_{edge} inevitably increases as the water edge inaccuracy is amplified. However, this does not interfere with the quality of obstacle detection as ShorelineNet still achieves a high F-score. Second, most of the false positive predictions of ShorelineNet are due to noises near the water edge. This often does not pose an immediate threat or misdirection to the control system of a USV, as the land segment can be accurately predicted by ShorelineNet and would be avoided in the first place. Third, the current standard evaluation metric does not count the detection of obstacles that overlap the shoreline as a true positive. If these obstacles are accounted for, the number of true positive predictions would increase by approximately 3-fold. Lastly, the ability for ShorelineNet to achieve high accuracy at a significant increase in speed means that it is a more efficient model, and makes it a practical network and can be readily integrated into the autonomous systems of USVs.

IV. CONCLUSIONS AND FUTURE WORK

We presented ShorelineNet, an efficient deep neural network that accurately detects obstacles in the shoreline scene with real-time performance. The performance of ShorelineNet has been well validated in dataset taken in practical maritime environments. The comparison with the state-of-the-art segmentation networks has demonstrated that ShorelineNet is able to achieve a fast inference capability while retaining a relatively high accuracy in detecting maritime obstacles. To improve the performance of ShorelineNet, our future work will be focusing on reducing the decoder size to further increase the runtime speed, as well as utilizing custom loss functions to further reduce false positive detection and improve accuracy.

REFERENCES

[1] J. Zhuang, L. Zhang, B. Wang, Y. Su, H. Sun, Y. Liu, and R. Bucknall, "Navigating high-speed unmanned surface vehicles: System approach and validations," *Journal of Field Robotics*.

[2] S. Ma, W. Guo, R. Song, and Y. Liu, "Unsupervised learning based coordinated multi-task allocation for unmanned surface vehicles," *Neurocomputing*, vol. 420, pp. 227–245, 2021.

[3] Y. Liu, R. Song, R. Bucknall, and X. Zhang, "Intelligent multi-task allocation and planning for multiple unmanned surface vehicles (usvs) using self-organising maps and fast marching method," *Information Sciences*, vol. 496, pp. 180–197, 2019.

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

[5] 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, 2015.

[6] 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*, vol. 20, no. 5, pp. 1787–1802, 2018.

[7] 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.

[8] 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.

[9] 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.

[10] B. Bovcon and M. Kristan, "A water-obstacle separation and refinement network for unmanned surface vehicles," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 9470–9476.

[11] ——, "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.

[12] 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.

[13] 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.

[14] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4510–4520.

[15] 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.

[16] A. Nguyen, D. Kanoulas, D. G. Caldwell, and N. G. Tsagarakis, "Object-based Affordances Detection with Convolutional Neural Networks and Dense Conditional Random Fields," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 5908–5915.

[17] A. Nguyen, D. Kanoulas, L. Muratore, D. Caldwell, and N. Tsagarakis, "Translating Videos to Commands for Robotic Manipulation with Deep Recurrent Neural Networks," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 3782–3788.

[18] 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.

[19] 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, 2017.