Compact and Fast Underwater Segmentation Network for Autonomous Underwater Vehicles

Jiangtao Wang[1][0000−0001−6630−3248], Baihua Li[1][0000−0002−4930−7690], Yang Zhou[1][0000−0001−5474−9605], Emanuele Rocco[2], and Qinggang Meng[1]

[1] Department of Computer Science, Loughborough University, Loughborough, UK
{J.Wang5, B.Li, Y.Zhou5, Q.Meng}@lboro.ac.uk.com
[2] Witted Srl, Piazza della Manifattura 1, 38068 Rovereto, Italy
emanuele@witted.it

Abstract. Reliable and real-time semantic segmentation is crucial for vision-based navigation tasks undertaken by AUVs (Autonomous Underwater Vehicles). However state-of-art deep learning segmentation networks could not be deployed on embedded devices with limited onboard resources, due to the required high computation capacity and the lack of capability to deal with poor underwater image quality. In this work we present a new deep underwater segmentation network, featured by a compact encoder and a lightweight decoder. We use only one step upsampling block to recover features maps from the encoder to significantly speed up the inference time. Furthermore, we adopt three strategies to improve the network accuracy. Firstly, in parallel with the main decoder path, we introduce a branch path to extract additional low-level features. Secondly, we use position attention module to enhance the high-level semantic information and use channel attention module to introduce extra global context as well as refine the inter-dependencies of each features. Thirdly, we proposed to use two additional auxiliary loss and smooth loss functions to better train the network, such that it will be more robust in segmenting images at varying resolutions and generating smooth boundaries. We validate our network accuracy on two different underwater segmentation datasets, a generalistic and a specialist one, and our model achieves the same level of accuracy of state-of-art networks. We also tested the network speed on different embedded platforms, and we showed it reaches real-time inference speed on both Nvidia Jetson GPU platforms TX2 and Nano, with respectively around 24 and 18 FPS (Frame Per Second). The proposed network inference is up to 27 times faster than other considered networks. Its high accuracy and speed will so pave the way for its deployment and application on AUVs systems.

1 Introduction

In the past years, Autonomous underwater vehicles (AUVs) have been developed to autonomously carry out various missions in the underwater environment, which would be otherwise too expensive and dangerous for the human labor. These range from wreckage search and localization, to marine science, and environmental protection. To be autonomous, those AUVs need similar abilities of

Fig. 1. Examples of underwater semantic segmentation. The first row presents the underwater images and the second row presents their corresponding segmentation ground truth. The first three columns from the left are related to the seagrasses segmentation task [1] while the others comes from SUIM dataset [2] for general underwater segmentation task.

UGVs (Unmanned Ground Vehicles) or UAVs (Unmanned Aerial vehicles), for example, free navigation, obstacle avoidance, semantic simultaneous localization and mapping. Scene understanding is the key to support the above tasks and, depending on water conditions, semantic segmentation driven by RGB imagery of the underwater environment could provide it in a cheap, efficient and reliable way. However underwater segmentation cannot be yet widely adopted on AUVs due to that it is still challenging to run segmentation algorithms that are both accurate and, due to computational capability constrains, still achieve real-time inference on the AUVs embedded platforms.

In fact, on one side the underwater environment negatively affects the accuracy of semantic segmentation by continuously modifying the outlook and appearances of the same underwater biological entities or instances. Fig. 1 presents six examples of underwater images and their corresponding segmentation ground truth. Those images come with various color shift in the green or blue and different levels of haziness and illumination; those depend on the sea physical parameters, such as depth, salinity or temperature, the presence and activity of biological entities, as plankton or algae, or density of dispersed particles. As in the case of seagrasses at shallow depths, water mass flows given by underwater currents or wave motion keep as well the underwater scenarios in continuous motion. All those unavoidable effects exacerbate the challenge of underwater segmentation.

On the hardware side, the AUVs have only the limited computational capabilities of their GPU embedded platforms, such as Nvidia Jetson TX2 or Nvidia Jetson Nano. On those deep learning-based segmentation methods, which achieve real time state-of-art performance on desktop GPU, do not reach the same inference speed. This speed loss could become a problem for AUVs navigation and control system since it requires vision perceptions - such as object detection and segmentation - to provide scene understanding information fast enough to make real-time decisions for navigation. Given the slow motion of AUVs (2-3 m/s) even during fast surveys, we consider a safe time delay threshold around 100ms.

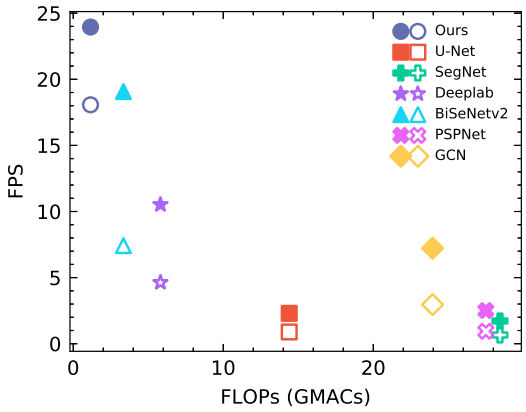

Fig. 2. This figure shows the FLOPs of ours and the other compared segmentation networks (small is better), and their inference speed as frame per second (high is better) on two different platforms. The solid symbols denote the speed test results on Nvidia Jetson TX2, while the hollow symbols relate to the ones on the Nvidia Jetson Nano. It shows that our proposed network is the fastest one with respect to other six networks.

To address the above challenges, we present so a lightweight segmentation neural network, which has a reduced number of parameters and FLOPs(floating point operations) and achieves realtime inference speed. We use also multiple segmentation predictions with different resolutions to help our proposed network address the challenges of underwater images taken at disparate seafloor distances with various resolutions. This approach helps also to optimize network training, as well as to improve the segmentation accuracy. We evaluated the proposed network on two datasets,the Seagrass [1] and SUIM [2] datesets, against which we validated its accuracy, and measured its inference speed on two Nvidia embedded platforms.

In summary, the main contributions of this work are:

1) we present a new segmentation network which can achieve high accuracy for underwater segmentation. In particular, we employ two additional loss functions, a smooth loss and auxiliary loss, to help train the network reach even higher accuracies. On both the Seagrass [1] and SUIM [2] datasets, our network respectively reaches 89.74 and 51.87 mIoU (mean Intersection Over Union), which are at the state-of-art level.

2) we design such network with a lightweight and well-designed decoder to reduce its total computational demands. Our network has just 1.153M parameters and 0.278G FLOPs which are respectively at least to 65% and 92% smaller than the ones of the considered alternatives as showed in Fig. 2.

3) we showed then the modest computational needs of the proposed network enable it to overcome the computational limitations of embedded platforms. Such network reaches real-time inference with 24 and 18 FPS respectively on Nvidia

Jetson TX2 and Nvidia Jetson Nano GPU platforms; those are respectively at least 9.54x and 19.22x faster than the most accurate network, PSPNet [3].

The proposed network is so optimal for deployment on AUVS embedded platforms for accurate and realtime segmentation inference.

2 Related Work

Underwater Segmentation Seagrass meadows coverage is a key index to measure for evaluating the health status of the marine ecological environment [4]. To automatically measure it from benthic RGB images, [5], [6], and [7] developed patches classification based methods. Those split the whole benthic images into super-pixels and patches at first, then use traditional machine learning methods, such as SVM (Support Vector Machine), to classify each patch and obtain the segmentation. Those authors deployed their segmentation methods on underwater robotics and AUVs where they tested their performances on embedded devices. However, these method cannot achieve the realtime inference, and had weak segmentation accuracies. A few authors, [8], [9] and [10], proposed instead end2end deep learning segmentation networks to attain high accuracies on seagrass region detection and segmentation. [11] developed a fast seagrass segmentation network running on GPU Desktop platform and, as well as, compared the existing state-of-art methods against a public seagrass dataset [1]. Instead, for general underwater segmentation on various semantic classes, [2] presented a light-end segmentation network and published its companion generalist underwater segmentation dataset, the SUIM dataset. Our proposed network is an end2end deep learning network optimised for embedded GPU platforms.

Semantic Segmentation Network Although current state-of-art segmentation networks have been originally proposed for medical image analysis, driverless cars or other surface applications, those successful networks could be applied as well to underwater segmentation tasks. The authors of [2] and [11] already proved that segmentation networks such as U-Net [12], deeplab [13], SegNet [14], PSPNet [3], FCN [15] can achieve excellent segmentation accuracy on different underwater datasets, while [16], [17], [18], [19] and [20] proposed a fast inference segmentation networks for real time use on Desktop GPU, which could be potentially deployed on AUVs while still achieve high accuracy.

Light weight encoder Classic Segmentation networks generally consist of two parts, an encoder (or backbone) and a decoder. The encoder is used to downsample the input image to low resolution to generate high-level features, while the decoder is used to restore the resolution of feature maps to achieve pixel-wise segmentation. To compress the segmentation network, it is a common strategy to adopt light weight convolutional networks as encoder, such as MobileNet [21], EfficientNet [22], and ShuffleNet [23]. For example employing MobileNet as backbone in PSPNet [3] or deeplab [13] can significantly reduce

the inference FLOPs and accelerate its speed with respect to using ResNet [24] as backbone. In our work, we utilize the mobilenet backbone [21] as encoder to optimize computational demands.

Attention Modules Attention modules have been wildly used in convolution neural networks. CAM (Channel Attention Modules), such as SENet (Squeeze Excitation Network) [25], CBAM (Convolutional Block Attention Module) [26], and SKNet (Selective Kernal Network) [27] can generate channel-wise weight to demarcate the feature maps across channel. In particular, CAM can enhance the essential and important feature maps to let the network focus on learning those. In our network we applied CAM to differentiate the inter-dependencies of low-level features maps across its channel dimensions, and re-rank relative importance of the features.

General position based attention (PAM) modules instead can generate instead pixel-wise attention maps. Those have abundant high level and global context information, which can significantly help to refine the segmentation prediction. In particular, works such as the PAM of [28] and the non-local block of [29] can generate position attention weights of one pixel to all other pixels. However, even if non-local block and other modules can improve the segmentation accuracy by introducing some global context, they will also increase the network computational demands due to increased network complexity. For a more efficient solution, it is possible to adopt other PAM structures such as GC-Net(Global Context Network) [30], CCNet (Criss Cross Network) [31], and ANN (Asymmetric Non-local Neural Network) [32], to improve segmentation accuracy with global context awareness but with contained computational demands. In our network we also employs PAM to generate high level context information at the decoder level.

3 Our Approach

We present a lightweight segmentation network which follows an encoder-decoder architecture. Its overall architecture is shown in Fig. 3. The network encoder can down sample the input images and generate different level feature maps, while the decoder can recover the resolution of the feature maps and make the pixel-wise segmentation prediction. Even if the proposed network can accept input images of any resolution, we experimentally choose the input resolution to be 320x256, as a balance between segmentation accuracy and computational overhead.

3.1 Encoder

We utilize MobileNet V2 [21] as encoder, as the blue box shown in Fig 3, which is an optimal network for mobile and embedded platforms. This encoder has a low parameter number and FLOPs, allowing fast inference on embedded platforms. In particular, the encoder down-samples the input image for five times (the

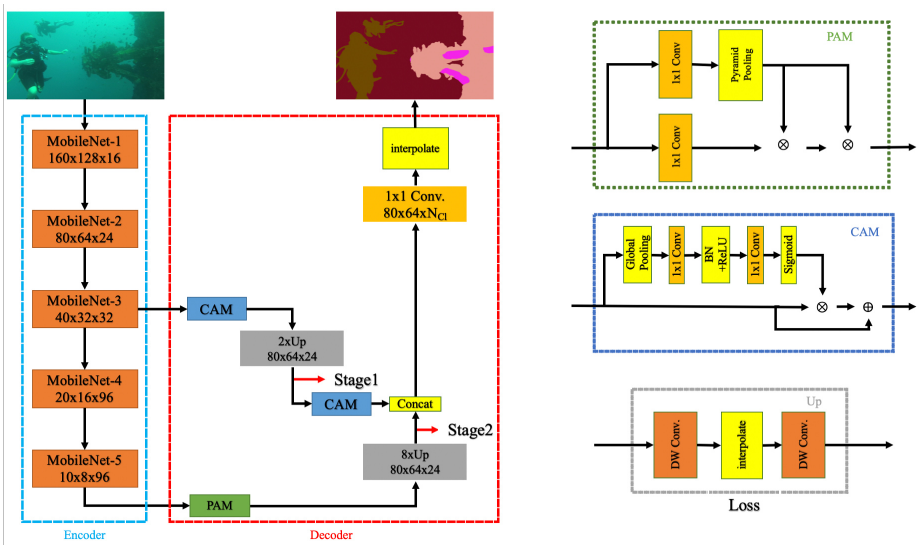

Fig. 3. This figure shows the overall architecture of our proposed segmentation network, which follows an encoder and decoder architecture. It has CAM (Channel Attention Module) to enhance the feature maps via channel-wise. It also has PAM (Position Attention Module) to get the high-level context information associated with pixel-to-pixel relationship. The two stages of segmentation prediction are shown as red arrows in decoder. Numbers under the block name refers to the width, height and channels number of output feature maps.

orange block in the encoder as shown in Fig 3), reducing the resolution by half after each down sampling step. Those down sampling convolutions generate also different resolution feature maps, from low-level to high-level features. We feed the final feature maps into the decoder. Since the high-level features have, in general, abundant context information, they can help improve the pixel-wise accuracy of the decoder. Additionally, other low-level and low resolution features can provide extra low-level information to the decoder for segmentation result refining.

3.2 Decoder

As shown in Fig 3 in the red box, the decoder consists of two paths: the one showed as starting with the PAM block, which is main path for the segmentation; a side path, showed as starting with the CAM block, which helps to refine the segmentation results. The main trunk quickly recovers the resolution of feature maps for the segmentation prediction by implementing an 8x up-sample convolution block at once. The PAM of the main trunk generates the pixel-to-pixel relationship matrix which contains high level context information. Our position attention module is based on the asymmetric pyramid non-local block of [32]

which uses 1x1, 2x2, 4x4 and 8x8 sizes for the pyramid pooling output to gain the spatial context information. After the PAM, the main trunk continue with an 8x up-sample block, quickly producing segmentation results just with three layers: this block starts with a 1x1 depth-wise convolution layer, followed by an 8x bi-linear interpolate layer, and ends up with a 1x1 depth-wise convolution layer. The efficient and quick resolution recovery achieved by the main trunk is the key to speed up the full segmentation prediction. However, such high expansion up-sampling inevitably leads to the loss of low level detail information.

To address such information loss, we adopt an additional parallel branch to take the low-level information from the encoder to the decoder which can help the main trunk generate precise pixel-wise segmentation. Such side branch starts with a CAM, and the follows with a 2x up-sampling block and ends with another CAM. The detailed structure of such channel attention module is shown within the blue box in Fig. 3. We calculate the channel attention according to global pooling and convolution. This method obtain the inter-dependencies of each feature maps, and also capture the global semantic context. By doing so, the CAM enriches the final inference with both the low-level detail information and the high-level context information. Both main path and branch path generate so the recovered 8x feature maps with respect to the encoder output.

To make the final segmentation prediction, we then concatenate both feature maps together and connect with an additional convolution layer and a 4x bi-linear interpolate layer, which recovers the full resolution segmentation prediction.

As presented, the proposed light decoder structure reduces the computational burden of the inference. This will speed up the segmentation inference on the limited computational capabilities of embedded platforms, while keep the accuracy performances.

3.3 Loss function

Our total loss function combines three different loss estimates, of which two are based on weighted cross entropy (WCE). We define as the $WCE(y, p)$ of the network prediction p and its corresponding segmentation ground truth y as:

$$WCE(y,p) = \frac{1}{N}\sum_{i} -\beta^{y_i} \log\left(\frac{\exp(p^{y^i})}{\sum_j \exp(p^j)}\right) \tag{1}$$

where the cross entropy calculation is based on the log softmax of the prediction p, β is the class weight, i is the pixel index and j is the class index. Both p and y should have the same resolution with N pixels.

Cross Entropy Loss Cross entropy loss l_{ce} is the main loss function used for the segmentation network training and is defined as follows:

$$l_{ce} = WCE(GT, Seg(X)) \tag{2}$$

where GT is the ground truth label towards input X, and $Seg(X)$ is the segmentation prediction generated by the segmentation network.

Smooth Loss From Seg(X), we calculate the edge smooth loss as [33] to let the network generate the smooth segmentation boundaries. Smooth loos is defined as follows:

$$l_{smooth} = \left|\partial_x \frac{Seg(X)}{Seg(X)}\right| e^{-|\partial_x X|} + \left|\partial_y \frac{Seg(X)}{Seg(X)}\right| e^{-|\partial_y X|} \tag{3}$$

Auxiliary Loss As indicated in Fig. 3 with red arrows, the up-sampling blocks generate two additional low-resolution segmentation predictions, which we use to calculate two auxiliary losses for the network training. Their loss function is defined as following:

$$l^i_{aux} = WCE(GT, Seg_i(X)) \tag{4}$$

where $Seg_i(X)$ is the i-th stage prediction of segmentation network.

Total loss We define so the total loss as weighted sum of the previous losses as eq. (5):

$$loss = l_{ce} + \sum_{i}^{2} \lambda^i_{aux} l^i_{aux} + \lambda_{Smooth} l_{smooth} \tag{5}$$

where λ_{aux} and λ_{Smooth} are the weights of the respective loss functions.

4 Experiment

In this section, we evaluate our network on two different underwater segmentation datasets: Seagrass [1] and SUIM [2]. The experiments compare our proposed method with six existing state-of-art segmentation approaches. The experiment results validate the advantages of our proposed segmentation network with respect to segmentation accuracies, network parameter numbers, computational demands (FLOPs) and inference speeds.

4.1 Dataset

Seagrass Dataset The seagrass dataset proposed by [1] involves 12682 images in total, which were taken by underwater cameras at different distances to the sea floor. 6037 of them, taken within the 0m to 6m range, have been labeled with ground truth information by human experts. They indicated two possible classes for each pixel: either 1 for seagrass meadows or 0 for the background. The first column of Fig 4 presents six examples taken from the Seagrass dataset, while the second column presents their ground truth. The first three row of these seagrass examples refers to close range (0m - 2m) images, while the other three

were taken at higher seafloor distances (2m - 6m). As shown in fig. 4, with the increase of the distance to seafloor and seagrasses, the imaging conditions such as luminosity, hue and haziness can rapidly change depending on water and weather conditions. This make the seagrass images have varying visual patterns as, for example, different colours, outlines and feature appearances. All this imaging variability increases the challenges of seagrass segmentation.

SUIM Dataset [2] recently published the SUIM (Segmentation of Underwater Imagery) which is instead a more general underwater semantic segmentation dataset. In this dataset, there are 1630 labelled images in total, which have eight unbalanced classes for pixel-wise annotation: BW (background and waterbody, 31%), HD (human divers, 1.9%), PF (Aquatic plants and sea-grass, 2%), WR (wrecks or ruins,7.3%), RO (robots, 0.3%), RI (reefs and invertebrates, 35.7%), FV (fish and vertebrates, 7.8%), and SR (seafloor and rocks, 13.9%). The example segmentation images and their annotations are shown in Fig. 5.

4.2 Implementation Details

To increase the data available for the network training, we combined several different image processing methods: we randomly crop a 320x256 patch from the original images (Crop); we rotate the images by a random angle from -20 to 20 degree (Rotation); we horizontally flip the image with 50% probability (Flip). After the data augmentation processing, we normalize each image with pre-calculated mean and variance. In experiments, we used PyTorch 1.5 [34] to implement and train all the networks on a single Nvidia RTX 2080ti GPU, with 100 (500) epochs on Seagrass (SUIM) dataset with Adam optimizer [35]. The batch size is set to 32 during training, momentum is 0.6 and weight decay is 0.001. Initial learning rate is 0.001, and we use linear schedule for the learning rate decay, as $\left(1 - \frac{E_{Current}}{E_{Total}+1}\right) * lr_{initial}$ to update learning rate after each training epoch.

4.3 Segmentation accuracy results

Segmentation Metrics We calculate mIoU (mean Intersection Over Union) and F1 Score ($\mathcal{F}$) to validate the network segmentation accuracy as in [2] and [1]. These two metrics are defined as following equations:

$$mIoU = \frac{1}{Classes} \sum_{class} \frac{|GT \cap Seg(X)|}{|GT \cup Seg(X)|} \tag{6}$$

$$\mathcal{F} = 2 * \frac{1}{Classes} \sum_{class} \frac{|GT \cap Seg(X)|}{|GT| + |Seg(X)|} \tag{7}$$

where $Classes$ refers to the total number of segmentation classes.

Table 1. All networks comparison of the segmentation metrics on the Seagrasses dataset

Range	Metrics	Ours	U-Net [12]	SegNet [14]	Deeplab [13]	BiSeNetv2 [36]	PSPNet [3]	GCN [37]
0-2m	mIoU	88.63	87.73	83.92	88.05	88.34	88.98	87.77
	$\mathcal{F}$	93.93	93.42	91.22	93.61	93.77	94.14	93.45
2-6m	mIoU	89.31	73.42	82.93	89.33	89.85	88.76	89.39
	$\mathcal{F}$	94.35	84.58	90.60	94.36	94.64	94.04	94.39

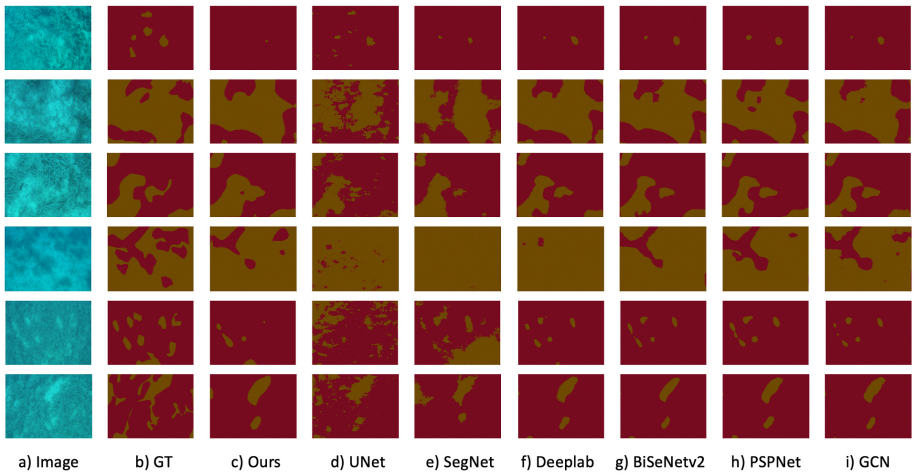

Fig. 4. Segmentation qualitative results on the Seagrass dataset. First three rows are taken from 0m to 2 m distance ranges and last three rows are taken from 2m to 6m distance ranges. The red pixels on segmentation present seagrass and yellow pixels present the seabed (background).

Seagrass Segmentation To compare the seagrass segmentation accuracy on Seagrass dataset [1], we equally train all the networks with 100 epochs, using the l_{ce} loss only. As shown in Table 1 and Fig. 4, our network achieves the same segmentation accuracy level of the others, in particular 88.63 mIoU and 93.93 $\mathcal{F}$, the second best accuracy over 0-2m range Seagrass dataset. Over the 2-6m seafloor distance dataset, it achieves instead 89.31 mIoU and 94.35 $\mathcal{F}$, which is the fifth best accuracy on such dataset. The gap between our proposed network and the best accurate network is 0.35 mIoU on 0-2m range and 0.54 mIoU on 2-6m range. However, the ablation experiments of the end of this section will show that our networks can also be improved and achieve the 2nd best accuracy on 2-6m range dataset when trained with the two additional proposed loss functions.

SUIM Segmentation For a more general evaluation, we also trained the networks on the SUIM dataset [2] with 500 epochs, and compared their segmenta-

tion accuracies. For fairness, our network was initially trained with only the l_{ce}. Fig. 5 and Table 2 present the segmentation results on such dataset.

In general the estimated segmentation accuracies are smaller than the ones obtained on the Seagrass dataset, which means the generalist underwater segmentation is a much more challenging task. On the SUIM dataset, our network achieves 50.60 mIoU and 66.14 $\mathcal{F}$, which is the 4th best accuracy when compared with all other networks. However, the mIoU gap between ours and the 2nd best network is just 2.08%; as we will show in the next section, this gap can be reduced to less than 0.8% by including the two additional loss functions for training. Yet, on the total 8 segmentation classes of this dataset, our network achieves the highest miou on the BW (background and waterbody) and SR (seafloor and rocks) classes.

Ablation Experiment As we mentioned earlier, we investigated as well the effects of our proposed two additional loss functions on the segmentation accuracies. We trained all networks with same epoch number but considered different combinations of loss functions for the two datasets. The combinations of these loss functions and results are reported in Table 3. We experimentally set λ^1_{aux} to 0.0001, λ^2_{aux} to 0.001 and λ_{Smooth} to 0.01. As shown in Table 3 and Fig. 6, using either additional l_{aux} or l_{smooth} improves the segmentation accuracy on both Seagrass and SUIM dataset. On the seagrass dataset, the mIoU increases from 88.63 up to 88.96 on 0-2m range images, and from 89.31 to 89.74 on 2-6m range images. On SUIM, such improvement is stronger: just using only l_{aux} loss, our network mIoU increases from 50.6 to 51.87. With the accuracy gains given by

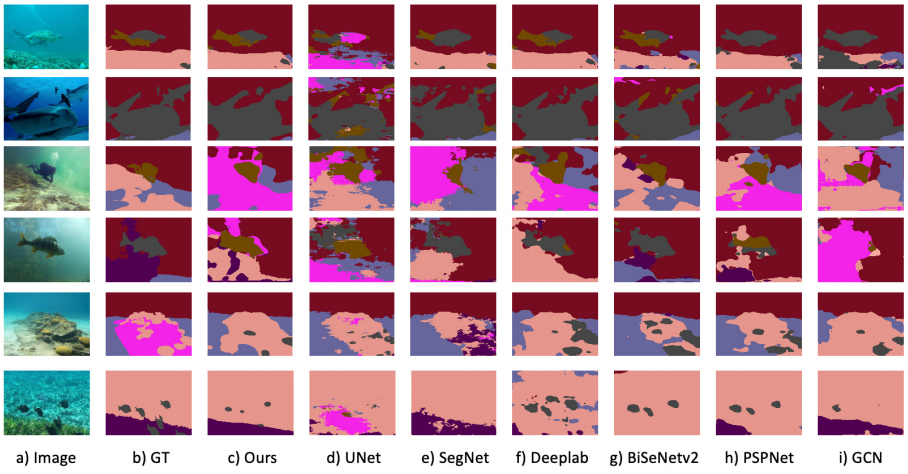

Fig. 5. Segmentation qualitative results on the SUIM dataset. The first column shows the test images and the second cloumn shows segmentation ground-truth. The third columns presents the segmentation prediction generated by our proposed network while the other columns refer to the results given by other networks.

Table 2. All networks comparison of the segmentation metrics on the SUIM dataset

	Ours	U-Net [12]	SegNet [14]	Deeplab [13]	BiSeNetv2 [36]	PSPNet [3]	GCN [37]
BW	84.62	79.46	80.63	81.82	83.67	82.51	79.32
HD	52.99	32.24	45.69	50.26	59.29	65.06	38.57
PF	11.46	21.86	17.45	17.06	11.27	28.54	15.09
WR	41.84	33.94	32.24	43.32	39.58	46.55	30.38
RO	49.67	23.66	55.74	63.63	56.54	62.88	54.25
RI	53.70	50.30	47.60	57.15	58.16	55.81	49.94
FV	45.98	38.15	43.93	43.60	56.00	46.75	36.09
SR	60.30	42.16	51.50	55.34	56.93	55.98	52.02
mIoU	50.60	39.85	46.85	51.52	52.68	55.51	44.46
$\mathcal{F}$	66.14	57.10	62.61	68.35	68.58	71.75	61.43

these two additional loss functions, our networks reaches the 2nd best accuracy on Seagrass dataset and 3nd best accuracy on SUIM dataset.

Table 3. Results from the ablation experiment evaluating the effect of different loss functions on our network mIoU

l_{ce}	l_{aux}	l_{smooth}	Seagrass 0m-2m mIoU	Seagrass 2m-6m mIoU	SUIM mIoU
✓			88.63	89.31	50.60
✓	✓		88.93	89.40	51.87
✓		✓	88.66	89.54	50.87
✓	✓	✓	88.96	89.74	50.79

4.4 Computational need and speed results

Of all the networks, we measured the FLOPs with respect to a single input image of 320x256 resolution, and the total parameter numbers. The FLOPs estimate directly measures the total computational overhead of network inference, which is inversely proportional to inference speed. We timed so also the inference speeds of the networks on the GPUs of two different embedded platforms, an Nvidia Jetson TX2 and Nvidia Jetson Nano. Table 4 presents the measurement results.

We found that our proposed network has less parameters and FLOPs than all other networks. Our network has just 1.153M parameters and 0.278G FLOPs which are respectively at least 65% and 92% smaller than the ones of the considered alternatives as showed in Fig. 4.Thanks to this, can execute the segmentation inferences with the highest FPS. By averaging over 100 inferences, our proposed network achieves 23.95 FPS and 18.04 FPS respectively on the

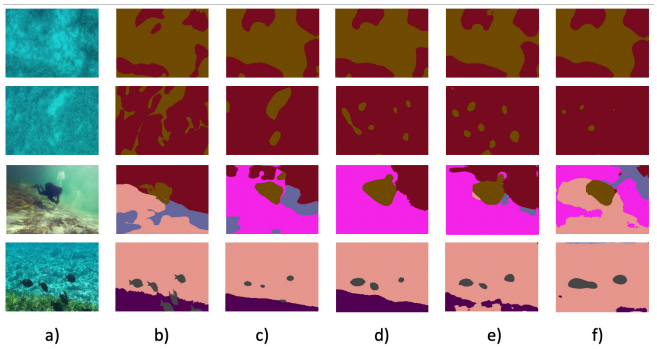

Fig. 6. The results of ablation experiment. Column a) presents test image samples and b) their corresponding ground truth; column c) shows the segmentation results by our network trained with l_{ce} only, d) with l_{ce} and l_{aux} only, and e) with l_{ce} and l_{smooth} only. The last column shows the segmentation results by our network trained with all the three loss functions.

TX2 and Nano. As comparison, our network inference is, on average, 1.255 and 2.43 times faster than bisenetv2 (2nd fastest network), and 13.9 and 27.37 times faster than SegNet (the slowest one) respectively on the TX2 and Nano. Furthermore, we observed that with respect to PSPNet, which is the most accurate network on the SUIM dataset and on the Seagrasses dataset on the 0-2m range, our network is 9.5 and 19 times faster on the Nvidia Jetson TX2 and Nano.

On the Nvidia TX2 and, in particular, on the Nano these measurements show that the existing networks, although they achieve high accuracy on underwater segmentation tasks, they are not optimal to be deployed on AUVs for real-time use. Our proposed network instead reaches high accuracies without compromising on inference speed.

Table 4. All network comparison of total parameter number, FLOPs and inference speed achieved on two Nvidia Jetson GPU embedded platforms

		Ours	U-Net [12]	SegNet [14]	Deeplab [13]	BiSeNetv2 [36]	PSPNet [3]	GCN [37]
Params (M)		1.153	14.396	28.442	5.813	3.347	27.501	23.952
FLOPs (GMACs)		0.278	38.793	61.390	8.278	3.830	49.782	7.087
FPS	TX2	23.95	2.297	1.723	10.52	19.07	2.509	7.224
GPU	NANO	18.07	0.89	0.66	4.63	7.42	0.94	2.96

5 Conclusion

In this paper, we present a lightweight underwater segmentation network well suited for deployment on the embedded platforms of AUVs. On two datasets, the generalist SUIM dataset and the specialist Seagrasses dataset, we showed that the new proposed network achieves the same accuracy level of other six state-of-art segmentation networks. We proposed also two additional loss functions to help further our network training and we demonstrated that they let it reach even higher segmentation accuracies. Our network attains 51.87 mIoU (3rd best) on the generalist dataset SUIM and 89.74 mIoU (2nd best) on 2-6m range Seagrass dataset. Anyhow, given a light encoder with mobilenet and simple decoder with single step resolution recovery, the proposed network is characterized by a much smaller parameter number and requires much less FLOPs than the other considered segmentation networks. Our network has at least 65% less parameter number and 92% less FLOPs than the ones of the considered alternatives. Contrary to those, it is so much less limited by the computational constrains of the embedded platform and achieves faster inferences. The speed tests shows in fact that the reduced computational requirements of our network allow it to attain much higher FPS than the others segmentation network on different Nvidia embedded platforms; for example it reaches up to 24 FPS on Nvidia Jetson TX2 which is 14 times faster than SegNet. The advantages of such fast inference speed and high segmentation accuracy make so our segmentation network optimal for deployment and real-time use on the embedded platforms of AUVs.

References

1. Reus, G., Möller, T., Jäger, J., Schultz, S.T., Kruschel, C., Hasenauer, J., Wolff, V., Fricke-Neuderth, K.: Looking for seagrass: Deep learning for visual coverage estimation. In: 2018 OCEANS-MTS/IEEE Kobe Techno-Oceans (OTO), IEEE (2018) 1–6
2. Islam, M.J., Edge, C., Xiao, Y., Luo, P., Mehtaz, M., Morse, C., Enan, S.S., Sattar, J.: Semantic segmentation of underwater imagery: Dataset and benchmark. arXiv preprint arXiv:2004.01241 (2020)
3. Zhao, H., Shi, J., Qi, X., Wang, X., Jia, J.: Pyramid scene parsing network. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2017) 2881–2890
4. Boudouresque, C.F., Bernard, G., Pergent, G., Shili, A., Verlaque, M.: Regression of mediterranean seagrasses caused by natural processes and anthropogenic disturbances and stress: a critical review. Botanica Marina **52** (2009) 395–418
5. Gonzalez-Cid, Y., Burguera, A., Bonin-Font, F., Matamoros, A.: Machine learning and deep learning strategies to identify posidonia meadows in underwater images. In: OCEANS 2017-Aberdeen, IEEE (2017) 1–5
6. Bonin-Font, F., Burguera, A., Lisani, J.L.: Visual discrimination and large area mapping of posidonia oceanica using a lightweight auv. Ieee Access **5** (2017) 24479–24494
7. Bonin-Font, F., Campos, M.M., Codina, G.O.: Towards visual detection, mapping and quantification of posidonia oceanica using a lightweight auv. IFAC-PapersOnLine **49** (2016) 500–505

8. Martin-Abadal, M., Guerrero-Font, E., Bonin-Font, F., Gonzalez-Cid, Y.: Deep semantic segmentation in an auv for online posidonia oceanica meadows identification. IEEE Access **6** (2018) 60956–60967
9. Martin-Abadal, M., Riutort-Ozcariz, I., Oliver-Codina, G., Gonzalez-Cid, Y.: A deep learning solution for posidonia oceanica seafloor habitat multiclass recognition. In: OCEANS 2019 - Marseille. (2019) 1–7
10. Sengupta, S., Ersbøll, B.K., Stockmarr, A.: Seagrassdetect: A novel method for detection of seagrass from unlabelled under water videos. Ecological Informatics (2020) 101083
11. Weidmann, F., Jäger, J., Reus, G., Schultz, S.T., Kruschel, C., Wolff, V., Fricke-Neuderth, K.: A closer look at seagrass meadows: Semantic segmentation for visual coverage estimation. In: OCEANS 2019-Marseille, IEEE (2019) 1–6
12. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical image computing and computer-assisted intervention, Springer (2015) 234–241
13. Chen, L.C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H.: Encoder-decoder with atrous separable convolution for semantic image segmentation. In: Proceedings of the European conference on computer vision (ECCV). (2018) 801–818
14. Badrinarayanan, V., Kendall, A., Cipolla, R.: Segnet: A deep convolutional encoder-decoder architecture for image segmentation. IEEE transactions on pattern analysis and machine intelligence **39** (2017) 2481–2495
15. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2015) 3431–3440
16. Wu, T., Tang, S., Zhang, R., Zhang, Y.: Cgnet: A light-weight context guided network for semantic segmentation. arXiv preprint arXiv:1811.08201 (2018)
17. Zhang, X., Chen, Z., Wu, Q.J., Cai, L., Lu, D., Li, X.: Fast semantic segmentation for scene perception. IEEE Transactions on Industrial Informatics **15** (2018) 1183–1192
18. Poudel, R.P., Liwicki, S., Cipolla, R.: Fast-scnn: Fast semantic segmentation network. arXiv preprint arXiv:1902.04502 (2019)
19. Paszke, A., Chaurasia, A., Kim, S., Culurciello, E.: Enet: A deep neural network architecture for real-time semantic segmentation. arXiv preprint arXiv:1606.02147 (2016)
20. Lo, S.Y., Hang, H.M., Chan, S.W., Lin, J.J.: Efficient dense modules of asymmetric convolution for real-time semantic segmentation. In: Proceedings of the ACM Multimedia Asia. MMAsia '19, New York, NY, USA, Association for Computing Machinery (2019)
21. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.C.: Mobilenetv2: Inverted residuals and linear bottlenecks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2018) 4510–4520
22. Tan, M., Le, Q.V.: Efficientnet: Rethinking model scaling for convolutional neural networks. arXiv preprint arXiv:1905.11946 (2019)
23. Ma, N., Zhang, X., Zheng, H.T., Sun, J.: Shufflenet v2: Practical guidelines for efficient cnn architecture design. In: Proceedings of the European conference on computer vision (ECCV). (2018) 116–131
24. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2016) 770–778
25. Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2018) 7132–7141

26. Woo, S., Park, J., Lee, J.Y., So Kweon, I.: Cbam: Convolutional block attention module. In: Proceedings of the European conference on computer vision (ECCV). (2018) 3–19
27. Li, X., Wang, W., Hu, X., Yang, J.: Selective kernel networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2019) 510–519
28. Fu, J., Liu, J., Tian, H., Li, Y., Bao, Y., Fang, Z., Lu, H.: Dual attention network for scene segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2019) 3146–3154
29. Wang, X., Girshick, R., Gupta, A., He, K.: Non-local neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2018) 7794–7803
30. Cao, Y., Xu, J., Lin, S., Wei, F., Hu, H.: Gcnet: Non-local networks meet squeeze-excitation networks and beyond. In: Proceedings of the IEEE International Conference on Computer Vision Workshops. (2019) 0–0
31. Huang, Z., Wang, X., Huang, L., Huang, C., Wei, Y., Liu, W.: Ccnet: Criss-cross attention for semantic segmentation. In: Proceedings of the IEEE International Conference on Computer Vision. (2019) 603–612
32. Zhu, Z., Xu, M., Bai, S., Huang, T., Bai, X.: Asymmetric non-local neural networks for semantic segmentation. In: Proceedings of the IEEE International Conference on Computer Vision. (2019) 593–602
33. Heise, P., Klose, S., Jensen, B., Knoll, A.: Pm-huber: Patchmatch with huber regularization for stereo matching. In: Proceedings of the IEEE International Conference on Computer Vision. (2013) 2360–2367
34. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al.: Pytorch: An imperative style, high-performance deep learning library. In: Advances in neural information processing systems. (2019) 8026–8037
35. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In Bengio, Y., LeCun, Y., eds.: 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings. (2015)
36. Yu, C., Gao, C., Wang, J., Yu, G., Shen, C., Sang, N.: Bisenet v2: Bilateral network with guided aggregation for real-time semantic segmentation. arXiv preprint arXiv:2004.02147 (2020)
37. Peng, C., Zhang, X., Yu, G., Luo, G., Sun, J.: Large kernel matters–improve semantic segmentation by global convolutional network. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2017) 4353–4361