Few-shot Segmentation and Semantic Segmentation for Underwater Imagery

Imran Kabir[3], Shubham Shaurya[1], Vijayalaxmi Maigur[1], Nikhil Thakurdesai[1],
Mahesh Latnekar[1], Mayank Raunak[1], David Crandall[1], and Md Alimoor Reza[2]

***Abstract*— This paper tackles image segmentation problems for underwater environments. First, we introduce a novel underwater animal-centric dataset with dense pixel-level annotations containing diverse fine-grained animal categories to mitigate the lack of diverse categories in the existing benchmarks. Then, we solve two image segmentation tasks using underwater images in this dataset: (i) *few-shot segmentation*, and (ii) *semantic segmentation*. For the segmentation task in a few-shot learning framework, we propose a novel attention-guided deep neural network architecture by infusing *attention modules* in various stages of our proposed network. We systematically explore how the learned attention maps can improve few-shot segmentation performance for underwater imagery. Finally, we assess the semantic segmentation problem on our proposed dataset by benchmarking it with two state-of-the-art semantic segmentation methods. We believe our new problem setup, i.e., *few-shot segmentation for underwater environments*, will be a valuable addition to the existing underwater semantic segmentation task. We believe our novel dataset will pave the way for developing better algorithms and exploring new research directions for marine robotics and underwater image understanding. We publicly release our dataset and the code to advance image understanding research in underwater environments: https://github.com/Imran2205/uwsnet.**

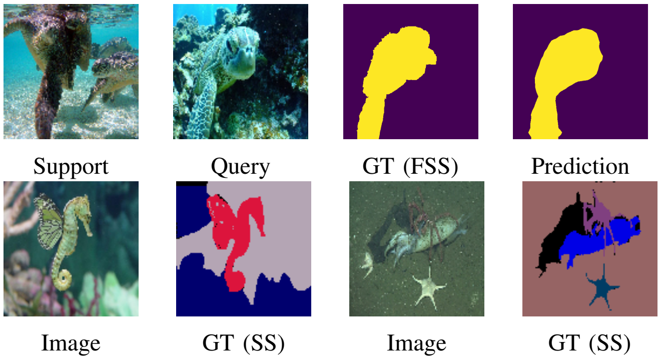

Fig. 1: Top row: Few-shot segmentation for underwater environments where the task is to find the segmentation map of the *query* image given the *support* image. Next to the *query* image, the ground truth, and predicted segmentation using our proposed method. Bottom row: Two pairs of input images and their corresponding semantic segmentation ground truths from our newly proposed underwater dataset with diverse animal categories. FSS: Few-Shot Segmentation. SS: Semantic Segmentation.

I. Introduction

Semantic segmentation is the task of assigning labels to image pixels from a predefined set of object categories. This classic computer vision problem has been extensively explored in both indoor and outdoor environments. Traditional approaches to solving semantic segmentation require the careful design of hand-engineered features [1], [2], [3]. The success of deep learning in various other image understanding problems paved the way for replacing the hand-engineered feature extraction process with deep neural network-based approaches. These end-to-end trainable networks produce more accurate segmentation [4], [5], [6], [7]. Nevertheless, these supervised methods require pixel-level manually annotated images, making the process highly time-consuming and labor-intensive. For example, annotating a single image might require 60-94 minutes, depending on the complexity of the scene [8], [9], [10]. Several possibilities have been explored to reduce the laborious manual annotation efforts [11], [12], [13]. One suggested solution is to annotate a single pixel inside each object [11]. However, the segmentation obtained from a model trained using those annotations exhibits lower levels of accuracy. An alternative approach involves augmenting the current training dataset of hand-annotated images by incorporating a substantial collection of synthetically generated images along with their corresponding annotations [12], [13]. This strategy requires an extensive collection of 3D models, their arrangement in a scene, and the utilization of a graphical rendering pipeline. Nonetheless, these synthetically rendered images may not fully capture the realism present in natural images. When facing a scarcity of large-scale training images, whether manually annotated or synthetically rendered, a viable alternative is to explore segmentation solutions within a few-shot learning framework [14], [15], [16], [17].

Several studies have delved into the few-shot segmentation solution in both indoor and outdoor environments [18], [15]. However, the segmentation in underwater environments has received relatively less attention. One contributing factor is the absence of a suitable dataset that encompasses a diverse range of marine life. In recent times, Islam et al. [19] introduced a semantic segmentation solution, along with a well-suited dataset curated for oceanic exploration and human-robot collaborative experiments. The dataset encompasses

[1] Luddy School of Informatics, Computing, and Engineering, Indiana University, Bloomington, IN 47408, USA {sshubham, vbmaigur, nthakurd, mraunak, mrlatnek, djcran}@iu.edu

[2] Department of Mathematics and Computer Science, Drake University, Des Moines, IA 50311, USA md.reza@drake.edu

[3] College of Information Sciences and Technology, Pennsylvania State University, State College, PA 16801, USA ibk5106@psu.edu

eight semantic categories, with the only animal category being a generic *fish (vertebrates)* as shown in Figure 2. The remaining categories represent common background elements such as *reefs, wrecks, sea floor, etc.*

To this end, we present a segmentation framework designed for underwater environments, boasting diverse applications such as fine-grained aquatic animal monitoring, underwater environment exploration, and robot collision avoidance [20]. Recent efforts to address the scarcity of underwater animal-centric datasets have certain limitations. These datasets either include multiple underwater animals, but with annotations limited to bounding-box level [21], or they have pixel-level annotations but cover only a smaller set of generic animal categories [19] (Fig 2). To overcome these constraints, we present a novel dataset with dense pixel-wise annotations, encompassing a diverse collection of **21** commonly found underwater animals such as *shrimp, crab, turtle, shark, whale, crocodile*. Ochal et al. [22] performed a comparative study of few-shot image classification methods in the underwater environment, employing both optical and sonar images. In contrast, our work goes beyond the scope of classification and delves into the task of dense pixel-wise segmentation within a few-shot learning framework. To the best of our knowledge, our work represents the first attempt to formulate few-shot segmentation within the context of underwater imagery. In summary, our contributions can be outlined as follows:

- Introduce a novel underwater animal-centric dataset comprising 576 images, with complete pixel-level annotations covering a diverse range of animal categories commonly found in underwater environments. Our dataset demonstrates **21** times greater diversity compared to the underwater segmentation dataset [19].
- Formulate the segmentation problem for underwater environments using a few-shot learning framework, and propose an innovative attention-guided deep neural network architecture for few-shot segmentation. We systematically investigate how the learned attention maps can enhance few-shot segmentation performance for underwater imagery. In 1-shot experiments, our top-performing model achieved remarkable improvements of **2.12%**, **0.56%**, **6.47%**, and **4.48%** over PANet [15], PMMs [16], HSNet [23] and ASNet [17], respectively, in mean IoU metric.
- Benchmark the proposed dataset for the task of semantic segmentation by employing two state-of-the-art methods, namely Mask2Former [24] and HRNetV2 [25].

The dataset, code, and trained models are openly accessible to the research community at the following link: https://github.com/Imran2205/uwsnet.

II. Related Work

Image segmentation involves the task of dividing an image into meaningful regions that possess similar visual or geometric properties. In the past, conventional segmentation approaches relied on low-level image cues [26], [27], [1] to group these regions. While these traditional methods effectively partition the image into meaningful regions, the regions themselves do not carry any class information.

Semantic Segmentation. A more meaningful task, known as semantic segmentation, is to assign labels to the partitioned regions. With the advent of deep learning, various deep neural network-based semantic segmentation solutions have been proposed, such as FCN [7], SegNet [6], DeepLab [5], HRNet [25], and Mask2Former [24]. More recently, several semantic segmentation datasets and methods have been proposed for autonomous driving applications [9], [10], [28], [29] or for off-road environments [30].

Few-shot Segmentation. Dong *et al.* [18] explored N-way k-shot segmentation. They employed a two-branch architecture, where the first branch acted as a feature vector extractor (*prototype learner*) and regularizer to avoid over-fitting. The second branch took the query image and the feature vector as input to produce a segmentation mask. Distance metric learning and non-parametric nearest neighbor classifiers were used to further improve the performance. PANet [15] proposed an effective network architecture that interchangeably used support and query images during training to reduce over-fitting. Yang *et al.* [16] proposed a prototype mixture model (PMMs) for few-shot segmentation. This model enforces the prototype-based semantic representation by correlating diverse regions of images with multiple prototypes. PMMs uses two CNNs with shared weights in the query and support branches as their backbone. The work of [31] et al. developed a probabilistic latent variable framework for few-shot segmentation. This model integrates attention to prototype construction. Min *et al.* [23] introduced the idea of squeezing a hypercorrelation with a 4D convolution in a pyramidal network. This method improves performance by gradually squeezing the feature dimension and aggregating local information into a global context. Kang *et al.* [17] extended the idea and improved the performance of the hypercorrelation-based model by introducing global self-attention to it. To this end, we build on the work of PANet [15] and infuse different types of attention modules to construct a novel prototype and assess their efficacy. Our experimental results suggest that adding attention modules helps improve the few-shot segmentation performance in underwater environments.

III. Pixel-wise Densely Annotated Underwater Dataset with Diverse Animals

In contrast to the abundance of datasets available for outdoor and indoor environments, there is a scarcity of such resources for underwater settings. The majority of existing underwater datasets either contain annotations limited to bounding-box level [21] or encompass a limited number of animal categories [19], [21]. We addressed the issue of limited category representation by introducing the Underwater Segmentation (UWS) dataset. This novel dataset consists of **576 images**, each densely pixel-level annotated, encompassing various entities commonly found in underwater environments, whether in the wild or within man-made conservatories like aquariums. All images in the dataset were collected from the internet using query terms such as *shark,* *polar bear, etc*. The dataset includes annotations for two task settings: i) Semantic Segmentation, and ii) Few-shot Segmentation. We provide further details on each of these settings.

Semantic Segmentation Setting. The SUIM dataset [19], featuring dense pixel-level annotations, is specifically designed for oceanic exploration and human-robot collaborative experiments in underwater environments. This dataset comprises only eight semantic categories, namely *fish (vertebrates), reefs (invertebrates), aquatic plants, wrecks/ruins, human divers, robots*, and *sea-floor*. However, it includes only one generic underwater animal category, *fish (vertebrates)*. In contrast, our UWS dataset presents fine-grained underwater animal categories, such as *seal, turtle, starfish, shrimp, crab, etc.*, showcasing twenty-one times more diverse animals. Additionally, the UWS dataset contains an additional 8 background categories, such as *coral, rock, water, sand, plant, human, iceberg, and reef*. The bar charts presented in Figure 2 illustrate the distinctions between the SUIM dataset and our UWS dataset.

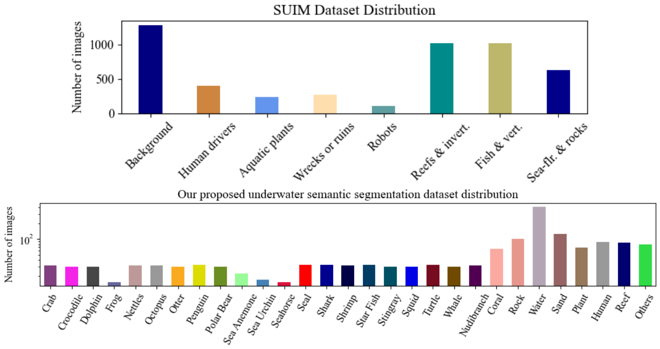

Fig. 2: Class-wise distribution of the images of the SUIM [19] dataset (top bar chart) and our proposed underwater segmentation dataset (UWS) for semantic segmentation setting (bottom bar chart).

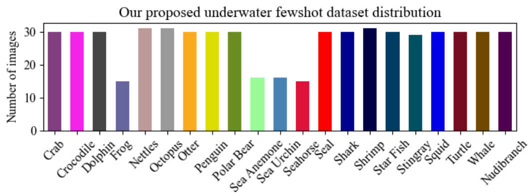

Fig. 3: Class-wise distribution of the images of our proposed underwater segmentation dataset (UWS) for few-shot segmentation setting.

Few-shot Segmentation Setting. For the few-shot segmentation setting, we prepared separate ground truths by retaining only the **21** underwater animal categories while collapsing others into a single background. The distribution of these 21 underwater animal categories is visualized in Figure 3. Figure 1 (top row) presents a sample image along with its corresponding ground truth annotation. We utilized the LabelMe Annotation Tool [32] to annotate the images.

IV. Method

A. *Proposed Approach for Few-shot Segmentation:*

To define the few-shot segmentation task for underwater environments, we are given a set of query images I_{query} for which we want to predict the object segmentation masks. We are also given another set of associated support images $I_{support}$ to help learn these segmentation masks. During training, we have ground truths for both support and query images. The setup is defined in an N*-ways-*n*-shot* format, where N represents the number of classes, and n indicates the number of support images needed to predict a segmentation mask for a query image. Following the prevalent setup found in existing few-shot segmentation literature [15], our underwater segmentation task involves learning to predict 1-way-1-shot and 1-way-5-shot segmentations. In the 1-shot scenario, the model uses one support image, while in the 5-shot scenario, it employs five support images to generate segmentation predictions. In this binary segmentation setup, the foreground refers to various underwater object classes like *polar bear, shark, crocodile*, etc., while the background encompasses elements such as *water, rock, coral, plant, iceberg, reef*, and so on. We build upon the foundations of PANet [15] and extend its capabilities by introducing a novel attention-guided architecture specifically designed for few-shot segmentation tasks. Our network architecture is illustrated in Figure 4. A description of the distinct components of our network architecture is discussed in the subsequent subsections.

1) Shared Feature Extractor with Attention: Given a pair of two sets of images – support image(s) (top branch) and a query image (bottom branch) – the *Shared Feature Extractor with Attention* module extracts 2D convolutional features for both sets. In the *Shared Feature Extractor with Attention*, we used a pretrained VGG-16 [33] as a 2D convolutional backbone and augmented it with an attention submodule to extract the features for the given support image(s) and the query image. The weights of this module are shared between both branches. The outputs of this module would be two sets of CNN feature maps ($F_{support}$, F_{query}).

2) Attention Guided Prototype Construction: Then we focus on constructing a useful compact prototype from the support feature map $F_{support}$. Ideally, this prototype should encode important features of our object of interest, which we want to segment in the query image. One simple idea would be to apply a pooling operation to aggregate the information captured by the support feature map $F_{support}$. More precisely, a global average pooling or max-pooling operation could be applied across the channels of $F_{support}$ [34]. In contrast, motivated by the success of the attention mechanism in various other computer vision tasks, we propose to learn an attention map on the support feature map $F_{support}$. Denote the learned attention map as $A_{support}$. Then, a prototype $P_{support}$ can be constructed based on this newly learned attention map, which can be subsequently matched against the attention map of the query image using the cosine similarity measure (as shown in the bottom branch within the

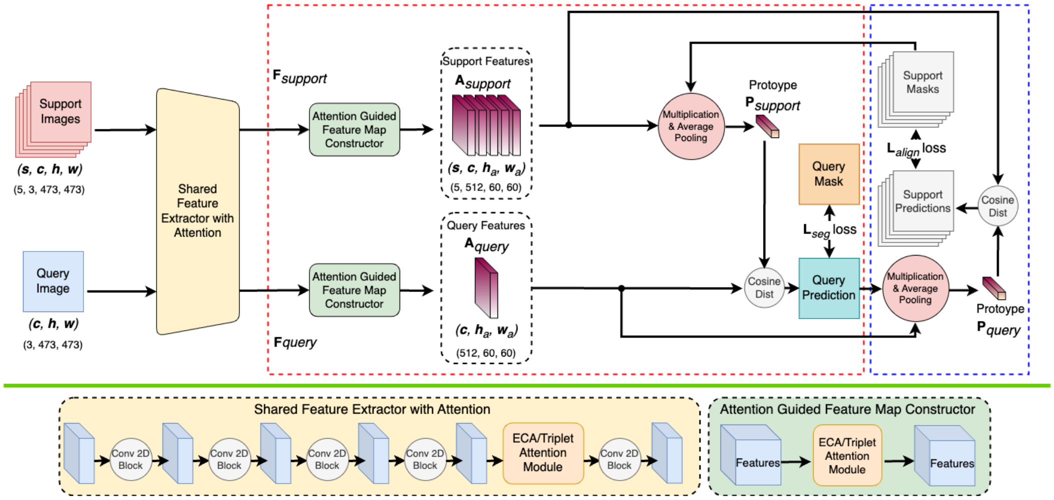

Fig. 4: Top part (above the green line): Our proposed network architecture for few-shot segmentation. Given support set image(s) and a query image, the network uses *Shared Feature Extractor with Attention* module to extract feature maps $F_{support}$ and F_{query}. From these two initial feature maps, *Attention Guided Feature Map Constructor* creates two attention maps $A_{support}$ and A_{query}. We construct our prototype from the attention map $A_{support}$, which we match against A_{query} to generate the segmentation map for the input query image. A symmetric mechanism is followed by reversing the roles of support and query (as depicted on the right-most branch within the blue-dashed rectangle). Bottom part (below the green line): Details of our *Feature Extractor with Attention* and *Attention Guided Feature Map Constructor* modules. Best viewed in color.

red-dashed rectangle in Figure 4). Our prototype $P_{support}$ is a vector of dimension 1xC, where C is the number of channels in the attention map $A_{support}$. We compute average pooling on the attention map to construct the prototype. In 5-shot segmentation, we aggregate the five prototypes into a single prototype for the support images by averaging them. A similar attention map is learned for the query feature map F_{query} before computing the cosine similarity between the support prototype $P_{support}$ and the attention map of the query feature map A_{query}. During the training phase of our model, we follow the above-mentioned method in reverse order (as shown in the right-most branch within the blue-dashed rectangle in Figure 4). In other words, we try to predict the true label for support image(s) using the query image. We want to emphasize that this specific branch of the model is not executed during the inference stage. During our experiments, we investigated two different types of attention mechanisms: i) ECA attention [35], and ii) triplet attention [36].

ECA Attention Module. ECA attention learns attention weights across the channels of an input tensor using a combination of global feature descriptor, adaptive neighborhood interaction, and broadcasted scaling. For additional details, we refer to the work of [35].

Triplet Attention Module. To better utilize different cross-dimensional relationships of the input feature map, we explore the triplet attention module. For additional details, we refer to the work of [36].

3) Loss Function: We train our network using pixel-wise cross-entropy loss. Additionally, we noticed performance gains with a dice loss [37], which measures the intersection over the union between the predicted mask and the ground truth mask. Given T training query images, the dice loss is computed as follows:

$$\mathcal{L} = \sum_{i=1}^{T} -D \tag{1}$$

where D is the dice coefficient computed as follows: $D = \frac{2\sum_{i=1}^{N} p_i g_i}{\sum_{i=1}^{N} p_i^2 + \sum_{i=1}^{N} g_i^2}$, and we took the negative of the dice coefficient because while training, the model tries to minimize the loss. N is the number of pixels in each training sample, p is the foreground segmentation prediction for i^{th} query image, and g_i is the corresponding ground truth. Our final loss is the sum of the loss calculated between the query segmentation prediction and the ground truth, $\mathcal{L}_{seg}$, and the loss calculated between support segmentation prediction and the ground truth of support images during the query to support segmentation, $\mathcal{L}_{align}$. Our final loss is computed as follows:

$$\mathcal{L}_{total} = \mathcal{L}_{seg} + \mathcal{L}_{align} \tag{2}$$

For 5-shot segmentation, $\mathcal{L}_{align}$ is the average of losses calculated for five support images.

B. Semantic Segmentation

Upon introducing a new task – the *few-shot segmentation task* – specifically tailored for underwater images, and proposing a novel method for addressing it, we also address an existing segmentation task – *semantic segmentation* in underwater environments [19]. Mathematically, given an image I, the task is to assign each pixel to one semantic label from a fixed set of semantic categories $\{1, 2, ..., C\}$,

Model	Shared Feature Extraction with Attention	Attention Guided Feature Map (for Support)	Attention Guided Feature Map (for Query)	Dice Loss	1-shot					5-shot				
					Split 1	Split 2	Split 3	Split 4	Mean	Split 1	Split 2	Split 3	Split 4	Mean
UWSNetV1		ECA	ECA		0.6881	0.6145	0.6679	0.6760	0.6616	0.7187	0.6394	0.6939	0.6878	0.6850
UWSNetV2	ECA	ECA	ECA		0.7048	0.6236	0.6750	0.7022	**0.6764**	0.7420	0.6291	0.7009	0.7077	0.6949
UWSNetV3		Triplet	Triplet		0.7028	0.6094	0.6720	0.6639	0.6620	0.7145	0.6344	0.6809	0.6952	0.6813
UWSNetV4		Triplet	Triplet	✓	0.6989	0.6096	0.6577	0.6760	0.6606	0.7179	0.6384	0.6904	0.6921	0.6847
UWSNetV5	Triplet	Triplet	Triplet		0.7068	0.6017	0.6705	0.7016	0.6702	0.7402	0.6197	0.6947	0.7115	0.6915
UWSNetV6	Triplet	Triplet	Triplet	✓	0.7066	0.6175	0.6813	0.6981	0.6759	0.7421	0.6396	0.7008	0.7048	**0.6968**

TABLE I: Ablation study for 1-shot and 5-shot segmentations. Each row denotes a specific version of our model, where we include or omit different components eg, attention modules, and dice loss in different stages of the proposed network (as discussed in Section IV). Split i denotes the model's performance on the images of Split i, following training on the images of the remaining three splits. The best performances are highlighted in boldfaces, while the second best are underlined.

Model	1-shot					5-shot					
	Split 1	Split 2	Split 3	Split 4	Mean	Split 1	Split 2	Split 3	Split 4	Mean	diff.
PANet (ICCV2019) [15]	0.6911	0.5992	0.6576	0.6729	0.6552	0.7157	0.6296	0.6822	0.6959	0.6809	0.0257
PMMs (ECCV2020) [16]	0.6876	0.6386	0.6570	0.7003	0.6708	0.7149	0.6379	0.6826	0.7091	0.6861	0.0153
HSNet (ICCV2021) [23]	0.6281	0.5739	0.6138	0.6309	0.6117	0.6801	0.6262	0.6887	0.6985	0.6734	0.0617
ASNet (CVPR2022) [17]	0.6293	0.5825	0.6704	0.6442	0.6316	0.6837	0.6523	0.7191	0.7326	**0.6969**	0.0653
UWSNetV2	0.7048	0.6236	0.6750	0.7022	**0.6764**	0.7420	0.6291	0.7009	0.7077	0.6949	0.0185
UWSNetV6	0.7066	0.6175	0.6813	0.6981	0.6759	0.7421	0.6396	0.7008	0.7048	0.6968	0.0209

TABLE II: Comparison with other state-of-the-art methods for 1-shot and 5-shot segmentations.

where C is the total number of semantic categories. We benchmarked our dataset for the semantic segmentation task using two state-of-the-art methods: (i) Mask2Former [24], and (ii) HRNetV2 [25].

V. EXPERIMENTS ON FEW-SHOT SEGMENTATION

Dataset Split and Evaluation Metric. Following the earlier few-shot segmentation works for other environments [38], [15], we created a split based on the animal categories of our underwater dataset. We partitioned our dataset of 21 classes into 4 folds. The first three folds comprised 5 classes each, while the last fold contained 6 classes. The resulting splits are as follows: **Split 1**: {*Crab, Dolphin, Frog, Turtle, Whale*}, **Split 2:** {*Nettles, Octopus, Sea Anemone, Shrimp, Stingray*}, **Split 3:** {*Penguin, Sea Urchin, Seal, Shark, Nudibranch*}, **Split 4:** {*Crocodile, Otter, Polar Bear, Sea Horse, Star Fish, Squid*}. We employed 3 splits for training the model, reserving the remaining split for testing the trained model. It is crucial to emphasize that during the model training phase, the classes included in the test split were entirely unseen by the model. In other words, the test split contained classes that the model had never encountered during its training process. Denote the training data classes as *seen* classes and testing data classes as *unseen* classes. We randomly choose a class from the *seen* classes to create the training samples. After selecting the class, we proceed to randomly choose either 1 or 5 (for 1-shot and 5-shot, respectively) support images and 1 query image from the chosen class. Following the work of [15], we repeat this selection process 1000 times in each epoch to create our training samples. Following the same protocol just described, we create test samples on the *unseen* classes for evaluation. A test split is comprised of 1000 image tuples and we saved their file names for the reproducibility of our model evaluation. Using cross-validation, we trained our model on 3 splits and then evaluated the trained model on the remaining split. For each split, we computed the mean IoU (Intersection over Union) as the evaluation metric. This involved calculating the IoU score for each class and then taking the average over all classes.

Ablation Study. We conducted our experiments by varying the types of attention modules, their placements inside our network architecture, and loss functions (see in Section IV-A and Table I). By incorporating an instance of an attention module within the shared feature extractor (e.g., UWSNetV2), we obtain two initial feature maps – support feature map $F_{support}$ and query feature map F_{query} (as shown in Figure 4). Then, another instance of the attention module is used to construct attention maps $A_{support}$ and A_{query} from the support features and the query features, respectively (as shown in Figure 4). In the process, our UWSNetV2 model in Table I was obtained by using the *ECA Attention Module* within the shared feature extractor, support, and query branches. When we drop the *ECA Attention Module* from the shared feature extractor while retaining it in the support and query branches, we obtain our UWSNetV1 model in Table I. If we replace all the instances of the *ECA Attention Module* with the *Triplet Attention Module* in our models UWSNetV2 and UWSNetV1, we obtain models UWSNetV5 and UWSNetV3, respectively. We train these four models using cross-entropy loss. If we replace the cross-entropy losses with dice loss for training our models UWSNetV3 and UWSNetV5, we obtain models UWSNetV4 and UWSNetV6, respectively. Dice loss is known to handle class imbalance better, which could potentially improve the performance of the models. The results of various combinations of these experimental setups are reported in Table I. Among the models, UWSNetV2 and UWSNetV6 emerged as the top performers.

Implementation. We initialize our feature extractor with

Model	Classes														
	Crab	Crocodile	Dolphin	Frog	Nettles	Octopus	Otter	Penguin	Polar Bear	Sea Ane.	Sea Urchin	Sea Horse	Seal	Shark	Shrimp	mIoU
HRNetV2 [25]	0.6944	0.8527	0.6733	0.7041	0.5624	0.5177	0.7401	0.7165	0.7477	0.5420	0.7229	0.4807	0.4126	0.7772	0.4300	-
Mask2Former [24]	0.6489	0.8789	0.6800	0.7345	0.1819	0.4315	0.5450	0.7670	0.9184	0.2003	0.5668	0.2416	0.5394	0.6426	0.6790	-
	Star Fish	Stingray	Squid	Turtle	Whale	Nudibranch	Coral	Rock	Water	Sand	Plant	Human	Reef	Other		
HRNetV2 [25]	0.8791	0.8619	0.4797	0.5192	0.7362	0.6348	0.2574	0.2778	0.8005	0.5081	0.4578	0.5919	0.2000	0.0252	-	**0.5794**
Mask2Former [24]	0.9482	0.9266	0.6225	0.9260	0.7210	0.8019	0.4786	0.4369	0.8904	0.5660	0.1161	0.4541	0.1024	0.0970	-	0.5770

TABLE III: Semantic segmentation results on our dataset using state-of-the-art methods for benchmarking.

VGG-16 [33] pretrained weights. All models were trained using an SGD optimizer with a learning rate of 0.001, a momentum of 0.9, and a weight decay of 0.0005. We trained our model for 40 epochs with batch size 1, and at every 10000 iterations, the learning rate decayed by 0.1 times.

Time and Memory Complexities. Table II depicts the top-performing models, UWSNetV2 and UWSNetV6, both containing 14.7M parameters. Our models were trained and tested using NVIDIA Titan Xp GPU. On a single GPU, the inference times for processing an image in the 1-shot and 5-shot settings are approximately 200ms and 300ms (on average), respectively.

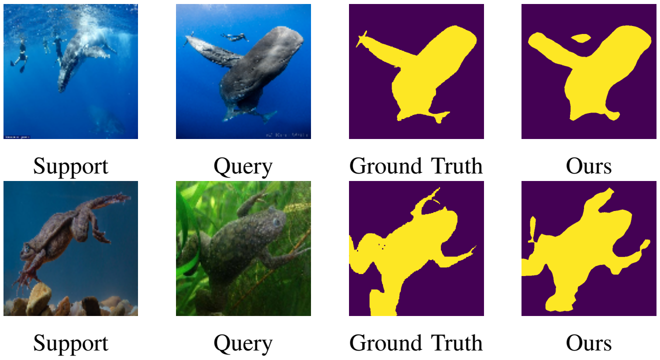

Fig. 5: Qualitative results for 1-shot segmentation using our best model – UWSNetV6 (separate test cases in each row).

Discussions and Comparison with Other Methods. We compare our method against four state-of-the-art few-shot segmentation methods: (1) PANet [15], (2) Prototype Mixture Model (PMMs) [16], (3) HSNet [23] and (4) ASNet [17]. All methods were trained using their publicly available implementations. The quantitative comparisons are reported in Table II. By incorporating attention modules, our model was able to learn improved feature maps and prototypes, resulting in better performance compared to PANet, which shares a similar architectural layout with our model but lacks attention modules. For our 1-shot segmentation setup, the best mean IoU score of **67.64%** was obtained by our model UWSNetV2, with an improvement of **2.12%**, **0.56%**, **6.47%** and **4.48%** over the PANet, PMMs, HSNet, and ASNet baseline models, respectively. In the case of the 5-shot segmentation setup, our method UWSNetV6 achieved a mean IoU score of **69.68%**, which is **1.59%**, **1.07%**, and **2.34%** higher than the scores of PANet, PMMs, and HSNet models, respectively, and comparable to the score achieved by ASNet. We also observed that the dice loss was more effective in improving the performance compared to models trained with cross-entropy loss. This can be verified by comparing the mean scores of UWSNetV5 (without dice loss) and UWSNetV6 (with dice loss) in both 1-shot and 5-shot segmentation. With dice loss, there is a gain of 0.57% and 0.53% for 1-shot and 5-shot segmentation respectively. Figure 5 shows some qualitative results, produced by our best model UWSNetV6 for 1-shot segmentation. Notably, all models demonstrate superior performance in 5-shot segmentation compared to 1-shot segmentation. The models benefit from learning better representations with the aid of multiple support images for each object. The relative improvements in performance between 1-shot and 5-shot segmentation are reported in the rightmost column of Table II.

VI. EXPERIMENTS ON SEMANTIC SEGMENTATION

Data Split and Evaluation Metric. For our semantic segmentation task, we utilized the dense pixel-wise annotations of all 29 categories present in our dataset (Figure 2). To create train and test partitions, a random split with an 80:20 ratio was employed, resulting in a training set of 461 images and a test set of 113 images. During training, data augmentation techniques such as random flip, shift, and rotation were applied. The evaluation of the method was performed using the standard mean IoU metric for semantic segmentation.

Implementation Details and Discussion. We employed the publicly available and official implementations of Mask2Former [24] and HRNetV2 [25], and trained them on our dataset. During training, we set the base learning rate to 10^{-3} and 10^{-4}, and the weight decay to 10^{-4} and 5×10^{-2} for HRNetV2 and Mask2Former, respectively. Both models were trained for 500 epochs. Table III presents the best class-wise IoU score and mean IoU over 29 categories after 100 epochs of training. The mean IoU scores achieved by the two networks were comparable, with the HRNetV2 model achieving a slightly higher score of **57.94%** mIoU compared to Mask2Former.

VII. CONCLUSION AND FUTURE WORK

In this study, we addressed the image segmentation challenges in underwater environments through the lens of few-shot learning. To tackle this problem, we presented a novel architecture specifically designed for few-shot segmentation tasks. The integration of attention maps in our proposed architecture yielded improved few-shot segmentation results when compared to methods lacking attention modules. To facilitate research in this domain and mitigate the scarcity of dense pixel-level annotations in underwater datasets, we introduced a novel underwater animal-centric dataset. Additionally, we evaluated the performance of our dataset for semantic segmentation using state-of-the-art methods in the field. Our future plans include expanding the dataset by incorporating a larger number of images and introducing additional animal categories.

References

[1] S. Gupta, P. Arbelaez, and J. Malik, "Perceptual organization and recognition of indoor scenes from RGB-D images," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2013.

[2] X. Ren, L. Bo, and D. Fox, "RGB-(D) scene labeling: Features and algorithms," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

[3] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, "Indoor segmentation and support inference from RGBD images," in *European Conference on Computer Vision*, 2012.

[4] J. Lambert, Z. Liu, O. Sener, J. Hays, and V. Koltun, "MSeg: A composite dataset for multi-domain semantic segmentation," in *Computer Vision and Pattern Recognition (CVPR)*, 2020.

[5] L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. Yuille, "DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018.

[6] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: a deep convolutional enclоder-decoder architecture for scene segmentation," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[7] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[8] S. R. Richter, V. Vineet, S. Roth, and V. Koltun, "Playing for data: Ground truth from computer games," in *European Conference on Computer Vision (ECCV)*, 2016.

[9] 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 *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[10] G. Neuhold, T. Ollmann, S. R. Bulò, and P. Kontschieder, "The mapillary vistas dataset for semantic understanding of street scenes," *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 5000–5009, 2017.

[11] A. Bearman, O.Russakovsky, V. Ferrari, and L. Fei-Fei, "What's the point: Semantic segmentation with point supervision," in *European Conference on Computer Vision*, 2016.

[12] A. Handa, V. Patraucean, V. Badrinarayanan, S. Stent, and R. Cipolla, "Understanding real world indoor scenes with synthetic data," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[13] S. Richter, V. Vineet, S. Roth, and V. Koltun, "Playing for data: Ground truth from computer games," in *European Conference on Computer Vision*, 2016.

[14] N. Dong and E. Xing, "Few-shot semantic segmentation with prototype learning," in *Britsh Machine Vision Conference (BMVC)*, 2018.

[15] K. Wang, J. H. Liew, Y. Zou, D. Zhou, and J. Feng, "Panet: Few-shot image semantic segmentation with prototype alignment," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 9197–9206.

[16] B. Yang, C. Liu, B. Li, J. Jiao, and Q. Ye, "Prototype mixture models for few-shot semantic segmentation," in *ECCV*, 2020.

[17] D. Kang and M. Cho, "Integrative few-shot learning for classification and segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[18] N. Dong and E. Xing, "Few-shot semantic segmentation with prototype learning," in *British Machine Vision Conference*, 2018.

[19] M. J. Islam, C. Edge, Y. Xiao, P. Luo, M. Mehtaz, C. Morse, S. S. Enan, and J. Sattar, "Semantic Segmentation of Underwater Imagery: Dataset and Benchmark," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020.

[20] T. Manderson, J. C. G. Higuera, R. Cheng, and G. Dudek, "Vision-based autonomous underwater swimming in dense coral for combined collision avoidance and target selection," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018.

[21] "Aquarium Dataset: The Henry Doorly Zoo in Omaha (October 16, 2020) and the National Aquarium in Baltimore (November 14, 2020)," https://public.roboflow.com/object-detection/aquarium.

[22] M. Ochal, J. Vazquez, Y. Petillot, and S. Wang, "A comparison of few-shot learning methods for underwater optical and sonar image classification," in *IEEE Global Oceans 2020: Singapore–US Gulf Coast*, 2020.

[23] J. Min, D. Kang, and M. Cho, "Hypercorrelation squeeze for few-shot segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

[24] B. Cheng, I. Misra, A. G. Schwing, A. Kirillov, and R. Girdhar, "Masked-attention mask transformer for universal image segmentation," 2022.

[25] J. Wang, K. Sun, T. Cheng, B. Jiang, C. Deng, Y. Zhao, D. Liu, Y. Mu, M. Tan, X. Wang, W. Liu, and B. Xiao, "Deep high-resolution representation learning for visual recognition," *CoRR*, vol. abs/1908.07919, 2019.

[26] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and Sabine, "SLIC superpixels compared to state-of-the-art superpixel methods," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2012.

[27] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," *International Journal on Computer Vision*, 2004.

[28] H. Abu Alhaija, S. K. Mustikovela, L. Mescheder, A. Geiger, and C. Rother, "Augmented reality meets computer vision: Efficient data generation for urban driving scenes," *International Journal of Computer Vision*, vol. 126, no. 9, pp. 961–972, 2018.

[29] O. Zendel, K. Honauer, M. Murschitz, D. Steininger, and G. F. Dominguez, "Wilddash - creating hazard-aware benchmarks," in *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

[30] P. Jiang, P. Osteen, M. Wigness, and S. Saripalli, "Rellis-3d dataset: Data, benchmarks and analysis," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021.

[31] H. Sun, X. Lu, H. Wang, Y. Yin, X. Zhen, C. G. Snoek, and L. Shao, "Attentional prototype inference for few-shot semantic segmentation," *arXiv preprint arXiv:2105.06668*, 2021.

[32] "LabelMe Annotation Tool," https://github.com/CSAILVision/LabelMeAnnotationTool.

[33] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *International Conference on Learning Representation (ICLR)*, 2015.

[34] Z. Deng, S. Todorovic, and L. Latecki, "Semantic segmentation of RGBD images with mutex constraints," in *International Conference on Computer Vision (ICCV)*, 2015.

[35] Q. Wang, B. Wu, P. Zhu, P. Li, W. Zuo, and Q. Hu, "ECA-Net: Efficient channel attention for deep convolutional neural networks," in *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[36] D. Misra, T. Nalamada, A. U. Arasanipalai, and Q. Hou, "Rotate to attend: Convolutional triplet attention module," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2021, pp. 3139–3148.

[37] F. Milletari, N. Navab, and S.-A. Ahmadi, "V-net: Fully convolutional neural networks for volumetric medical image segmentation," 2016.

[38] A. Shaban, S. Bansal, Z. Liu, I. Essa, and B. Boots, "One-shot learning for semantic segmentation," *arXiv preprint arXiv:1709.03410*, 2017.