A Comparison of Few-Shot Learning Methods for Underwater Optical and Sonar Image Classification

1st Mateusz Ochal Heriot-Watt University Edinburgh, UK m.ochal@hw.ac.uk 2nd Jose Vazquez

SeeByte
Edinburgh, UK
jose.vazquez@seebyte.com

3rd Yvan Petillot Heriot-Watt University Edinburgh, UK y.r.petillot@hw.ac.uk 4th Sen Wang Heriot-Watt University Edinburgh, UK s.wang@hw.ac.uk

Abstract—Deep convolutional neural networks generally perform well in underwater object recognition tasks on both optical and sonar images. Many such methods require hundreds, if not thousands, of images per class to generalize well to unseen examples. However, obtaining and labeling sufficiently large volumes of data can be relatively costly and time-consuming, especially when observing rare objects or performing real-time operations. Few-Shot Learning (FSL) efforts have produced many promising methods to deal with low data availability. However, little attention has been given in the underwater domain, where the style of images poses additional challenges for object recognition algorithms. To the best of our knowledge, this is the first paper to evaluate and compare several supervised and semi-supervised Few-Shot Learning (FSL) methods using underwater optical and side-scan sonar imagery. Our results show that FSL methods offer a significant advantage over the traditional transfer learning methods that fine-tune pre-trained models. We hope that our work will help apply FSL to autonomous underwater systems and expand their learning capabilities.

I. INTRODUCTION

Underwater object recognition is generally more challenging than in the usual indoor/outdoor environments due to the unique interaction of light in the water that distorts optical images. Water molecules, dust, and other floating particles can cause substantial attenuation of light, limit sensing range, affect the color, and introduce haze into pictures. In addition to optical cameras, acoustic sensors equip many underwater systems. Unaffected by lighting conditions, acoustic sensors have an extended sensing range, offering a significant advantage. However, sonar still greatly suffers from noisy sensor input and lower resolution. These characteristics can negatively affect the performance of deep convolutional neural networks (DCNN) [1], [2].

A key component for achieving good performance is training on large datasets [3]. However, obtaining larger datasets can be expensive and impractical in the marine setting due to the high operational costs and time constraints associated with underwater missions. A low abundance of some types of objects can further limit extensive gathering of data. Moreover, in real-time operations, it can be infeasible to perform rigorous labeling of data. Finding an algorithm capable of learning from only a handful of samples would be beneficial not only in the underwater domain but also to the general robotics and computer vision communities.

A variety of regularisation techniques address the problem of learning with limited data. One of the popular methods is transfer learning (TL), which has seen overall success in the underwater setting [1]. In TL, a network is typically trained on a significantly larger but readily available dataset, and later the model is fine-tuned on a smaller domain-specific dataset. However, TL alone may still require thousands of images in the smaller dataset to generalize reliably.

Over the past several years, there has been a renewed effort in developing more efficient algorithms to perform *Few-Shot Learning (FSL)*. FSL methods are commonly trained through meta-learning (e.g., MAML [4]) that aims to teach models how to learn from a few samples. Recent efforts have created a range of robust methods and proved to be promising for alleviating the problem of learning with limited data.

While FSL methods have been extensively tested on generic classification datasets, little attention has been given to practical underwater scenarios. To this end, we compare various FSL methods on a range of challenging optical and sonar datasets. We identify the state-of-the-art and highlight some challenges still faced by FSL methods. Our main contributions can be summarised as follows.

- To the best of our knowledge, our work is the first to compare the performance of several FSL methods for underwater sonar and optical image classification.
- We show that FSL methods offer a significant advantage over the traditional methods of fine-tunning.
- We show that pre-meta-training FSL methods on generalpurpose datasets can further improve performance, even when the image types differ significantly.
- We discuss the practicality of using FSL methods in realistic underwater robotics scenarios, highlighting their limitations, and proposing directions for future research.

This paper is structured as follows. We begin with an overview of the related literature in section II, describing the current efforts of training deep learning models with limited data, few-shot learning, and on underwater images. In section III we explain the datasets used for our experiments, before describing the examined methods in section IV and the experimental setup in section V. We report results in section VI, and discuss the limitations of few-shot learning and our experiments in section VII.

II. RELATED WORK

A. Learning with Limited Data

DCNN models can contain well into tens of millions of trainable parameters. As an example, EfficientNet-B7 [5], which achieves state-of-the-art performance on ImageNet [6], contains about 66M trainable parameters. Generally, the more parameters a model has, the greater its capacity to learn intricate patterns present in the data and achieve higher accuracy performance [5].

However, large models tend to overfit on small training datasets because they cannot learn a correct distribution of data due to the low variance of the training set, leading to high bias. The problem of overfitting has been addressed by numerous regularisation techniques, such as weight-decay [7], [8], dropout [9], [10], data augmentation [11], transfer learning [12] and many others [13]. A regularisation method can be "any supplementary technique that aims at making the model generalize better, i.e., produce better results on the test set" [13].

B. Few-Shot Learning

Few-Shot Learning (FSL) models aim to classify between classes from only a handful of sample representatives. Specifically, in a k-shot n-way FSL classification task, a model is given a small training set (called a *support set*) consisting of n never-seen-before classes with k image-label pairs per class. The goal is to use the support set to correctly classify a small evaluation set (called a *target set*) containing a different set of image-labels pairs sampled the same n classes. One-shot learning is an extreme case of FSL, which utilizes only a single support sample from each class (k = 1).

The approaches to FSL algorithms can be broadly categorized into five categories [14]: metric-learning, optimizationbased, hallucination, probabilistic, and domain adaptation. Metric-learning approaches (such as Prototypical Networks [15]–[17]) learn a feature extractor function capable of uniquely describing images from novel classes. Optimizationbased approaches (such as MAML [4] and Meta-Learner LSTM [18]) aim to achieve efficient learning through a guided optimization process on the support set. Hallucination or data augmentation techniques perform affine and color transformations on the support set to create additional data points, for example, [19] exploits an imperfect Generative Adversarial Network to generate additional negative examples that refine the class boundaries in feature space. Probabilistic methods use Bayesian inference to learn and classify samples (eg., GPShot [20]). Domain adaptation or transfer-learning that are pre-trained using classical supervised learning; these include fine-tuning baselines as well as varients, like Baseline++ [14].

FSL methods often learn through meta-learning, which employs three phases: meta-training, meta-validation, meta-testing¹. During meta-training, models can learn general features and hyperparameters that can be used later in the FSL

task. *Episodic training* [21] is a popular way of meta-training where a learner model is repeatedly exposed to batches of FSL classification tasks sampled from a more extensive but different set of classes. This process allows methods to exploit readily available datasets such as ImageNet [6].

C. Underwater Object Classification

Underwater object classification faces many unique challenges. Optical images' quality is strongly affected by the interactions of light with water molecules and other floating particles. These interactions introduce haze, noise (blur and 'marine snow' [22]), discoloring, and non-uniform illumination of objects. These factors can combine with various levels of strengths, and make object classification much more challenging to perform. Standard computer vision datasets (e.g., ImageNet) contain only up to a few underwater classes and do not generalize well to underwater optical datasets that contain higher levels of noise and color distortion [1], [23].

Additionally, there is a lower abundance of publically-available labeled underwater datasets. Many authors [1], [23]–[26] choose to train neural networks using transfer learning, by pre-training on nonspecialist datasets such as ImageNet [6], and then fine-tuning last of few layers of the pre-trained model on the smaller underwater dataset. Some authors such as [23] apply rigorous data augmentation (including rotation, random cropping, flipping, and color-shifting), which further boosts performance. Some authors, such as [27]–[30], parse datasets using image enhancement methods to improve the quality of images by restoring the actual color of objects, remove haze, and denoise. Image enhancement techniques can aid human visibility, and some authors such as [30] show them improving object tracking performance.

Due to the limitation of optical vision, it is common to equip underwater vehicles with supplementary sonar cameras [31]. Imaging sonar has become a widely adopted solution for providing measurements in many practical underwater operations [2], [32]–[34]. It offers significant advantages over optical cameras due to its robustness to water turbidity and variable lighting conditions. Side-scan sonar (SSS) is particularity popular for surveying and mapping due to its wide coverage and bathymetric capabilities [2]. It can have a range of over a hundred meters. However, the acoustic signal is not perfect and has its limitations. For example, it does not provide any color information and has a lower resolution than optimal images taken with modern cameras. The resolution varies with the distance of detected objects, and there is a trade-off between accuracy and range. The random sensor noise, viewing angle dependency, and sonar reflection of materials further contribute to the difficulty of working with sonar. As a result, it is common to equip vehicles and take advantage of both sensors. Although fusing input signals from both sensory modalities is complex and uncommon, some successful attempts have been made [31].

Despite the challenging nature of sonar data, [2] has successfully applied a pre-trained ResNet-50 [35] (on ImageNet [6]) for a reliable shipwreck recognition system. [32] has ap-

¹Due to clashing terminology of two communities, we make it explicit when referring to the meta-learning training/evaluation (by adding a prefix 'meta-') as opposed to support-set learning and target set evaluation.

plied a Faster-RCNN [36] with rigorous data augmentation for underwater object detection on both real and simulated sonar images. Transformantions on the training set included color inversion, horizontal and vertical flipping, scaling, rotation, and translation.

In the context of few-shot learning, and to the best of our knowledge, only one research paper has applied FSL on underwater sonar images [37]. However, the authors evaluate only a single method, called Siamese Networks [38], with no comparisons between alternative methods. Moreover, no quantifiable measure (such as accuracy) is reported offering limited insight into the underwater FSL problem.

III. DATASETS

FSL models are typically meta-trained using three disjoint dataset splits, one for each meta-learning phase: meta-training, meta-validation, and meta-testing. Unlike in classical supervised learning, the classes for each phase are strictly non-overlapping. Mini-ImageNet [18] is a popular benchmarking dataset for FSL models. It is a downscaled subset of ImageNet-2012 [6] containing only 100 of the original classes and only a few underwater classes and no sonar images. It is split into 64/16/20 classes for meta-training/meta-validation/meta-testing phases, respectively.

In our experiments, we evaluated methods on two color and two simulated-sonar datasets - offering an easier and a more difficult setting for each modality. When selecting datasets, we had to meet specific criteria, namely:

- 1) the datasets had to be of underwater images to fit the scope of this research,
- 2) contain at least 15 distinct classes to perform 5-way classification during each of the meta-learning phases,
- 3) contain at least 40 images per class to fit the minimum support/target set setup.

For these reasons, we chose the publically-available Fish Recognition dataset [39] and a privately-held Pipeline Feature dataset containing higher levels of blur and discoloration. From the original 23 classes of the fish dataset, we filtered classes with less than 40 samples. The remaining 19 classes were divided into 9/5/5 classes for meta-training/meta-validation/meta-testing phases, respectively. Similarly, in the pipeline dataset, containing 16 classes in total, we used 6/5/5 classes. All images were scaled to 84 by 84 pixels. In contrast to the fish dataset, Pipeline Features is significantly color shifted towards the green end of the color spectrum, offering a more challenging but realistic underwater scenario.

Sonar is integral in many underwater robotics systems. Due to the scarce availability of public sonar datasets, a specialized side-scan sonar (SSS) simulator was used to generate two datasets, as described in [40]. The simulator works by ray tracing a 3D Computer-Aided Design (CAD) model to emulate the signal received by a sonar sensor, producing realistic shadows and highlights of synthetic contacts (objects). We note that the sonar simulator's quality was validated in experiments with human participants and DCNN networks, which were unable to distinguish between real and simulated imagery [40].



(a) Mini-ImageNet [21], showing a wolf, dog, lipstick, ant, and some fish.



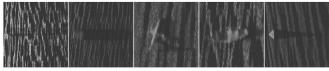
(b) Fish Recognition [39], showing five different fish species.



(c) Pipeline Features include an anode, grout bag, shell, fish and sea urchin.



(d) SSS (flat), showing an anchor, cube, plane, boat, and pyramid.



(e) SSS (rippled), showing an anchor, cube, plane, boat, and pyramid.

Fig. 1. Image examples from the datasets used in this work, representing only a small subset of available classes.

We inserted 18 different synthetic contacts into two types of simulated seabeds at various orientations and depth levels. We refer to the two seabed types as *flat* and *rippled*, with the latter offering a more challenging scenario. For each type of seabed, we used 8/5/5 classes for the three meta-learning phases. To generate the images, we cropped an area centered around each object with a large margin around it to include the shadows. Each image was then scaled to 84 by 84 pixels. Figure 1 shows image examples.

In our experiments, we also evaluated a couple of semisupervised FSL methods. To allow these methods to utilize unlabeled samples, we further partitioned each meta-learning dataset split, into a 40%/60% labeled/unlabeled partitions.

IV. METHODOLOGY

In this section, we give a low-level description of FSL methods evaluated in this work: Prototypical Network (PN) [15] and its variants, Relation Networks [41], Soft K-Means ProtoNets [16], and Consistent Prototypical Networks [17]. We begin by formally introducing the task of supervised FSL classification and the appropriate methods. Later, we introduce the semi-supervised FSL setting.

Consider the problem of a k-shot n-way classification task sampled from a dataset \mathcal{D} . A model is given a *support set*, $\mathcal{S} = \{(x_1, y_1), ..., (x_s, y_s)\} \sim \mathcal{D}$, containing n unique classes with k images per class $(|\mathcal{S}| = k \times n)$. The goal of the model is to correctly classify a *target set*, $\mathcal{T} = \{(x_1, y_1), ..., (x_t, y_t)\} \sim \mathcal{D}$, containing different samples from the same n classes (i.e. $X(\mathcal{T}) \cap X(\mathcal{S}) = \emptyset$ and $Y(\mathcal{T}) \equiv Y(\mathcal{S})$). *Episodic training* [21] is a popular way to meta-train FSL models, where models are exposed to mini-batches of k-shot n-way classification tasks sampled from a similar but disjoint dataset \mathcal{D}_{train} , where $\mathcal{D}_{train} \cap \mathcal{D} = \emptyset$.

1) **Prototypical Network**: A Prototypical Network [15] computes a representation of the support images for each class and assigns a class of a target image based on its similarity in embedding space. Specifically, support and target images are mapped into a feature space, through a non-linear mapping function $f_{\phi}: \mathbb{R} \to \mathbb{R}^{M}$, parameterized by the trainable parameters ϕ . A class's prototype, $\mathbf{p}_{c} \in \mathbb{R}^{M}$, is the mean of the mapped support samples belonging to a single class:

$$\mathbf{p}_c = \frac{\sum_i f_\phi(x_i) z_{i,c}}{\sum_i z_{i,c}} \tag{1}$$

where $z_{i,c} = \mathbf{1}$ when $y_i = c$ and $z_{i,c} = \mathbf{0}$ when $y_i \neq c$. Given a target point $(x_j, y_j) \in \mathcal{T}$ and a distance function, $d: \mathbb{R}^M \times \mathbb{R}^M \to [0, +\infty)$, the model computes a similarity between the mapped target point and each of the prototypes. A softmax over the distances produces a probability distribution p over the classes seen in the support set:

$$p_{\phi}(y = c|x_j) = \frac{\exp(-d(f_{\phi}(x_j), \mathbf{p}_c))}{\sum_{k'} \exp(-d(f_{\phi}(x_j), \mathbf{p}_{c'}))}$$
(2)

The model is meta-trained by minimizing the average negative log-probability:

$$J(\phi) = -\log p_{\phi}(y = y_j | x_j) \tag{3}$$

where y_j is the true class of x_j . Figure 2 shows an intuition of this method.

2) **Relation Network:** A Relation Network [41] augments the original Prototypical Network [15] and replaces the distance measure, d, with a relation module g_{φ} , parametized by trainable parameters φ . Specifically, first mapped target points and the prototypes are combined with an operator $h(\mathbf{p}_c, f_{\varphi}(x_j))$ that concatenates each target point with each prototype. Secondly, each of the concatenated vectors are passed through the relation module to produce relation scores, $r_{k,j}$, between a class's prototype, \mathbf{p}_c , and the target image x_j :

$$r_{k,j} = \sum x_i g_{\varphi} \left(h \left(\mathbf{p}_c, f_{\phi}(x_j) \right) \right) \tag{4}$$

The embedding function f_{ϕ} and the relation module g_{φ} are meta-trained end-to-end using the mean squared error (MSE).

B. Semi-supervised few-shot learning definition

In a semi-supervised few-shot classification task, in addition to the labeled support-set, $S \sim \mathcal{D}$, a model is also given an unlabeled set of images, $\tilde{S} = \{x_1, ..., x_{\tilde{s}}\}$, sampled from an unlabeled dataset $\tilde{\mathcal{D}}$. As before, the goal is to correctly classify the target set $\mathcal{T} \sim \mathcal{D}$. Episodic training [21] replaces datasets \mathcal{D} and $\tilde{\mathcal{D}}$ with \mathcal{D}_{train} and $\tilde{\mathcal{D}}_{train}$, respectively. Dataset $\tilde{\mathcal{D}}_{train}$ can be the same as \mathcal{D}_{train} , however, without losing generality we keep them seperate in notation.

1) Prototypical Network with K-Means Refinement [16]: This method also augments the original Prototypical Network [15] and refines the prototypes using the unlabeled data \tilde{S} . This method is almost identical to the original with the exception that the prototypes, \mathbf{p}_c , are replaced by the refined prototype, $\tilde{\mathbf{p}}_c$, for each class, k. The refinement process use an iteration of the Soft K-Means algorithm (where K=k) on mapped images from S and \tilde{S} .

The prototypes \mathbf{p}_c (defined in Eq. 1) act as the initial positions of the cluster centroids (i.e. $\tilde{\mathbf{p}}_c \leftarrow \mathbf{p}_c$). Each labeled example $x_i \in S^{(X)}$ is given a hard centroid assignment $(z_{i,c} = \mathbf{1} \ [y_i = c])$ since their label is considered known and therefore fixed. In contrast, each unlabeled sample \tilde{x}_r is given a partitial ('soft') assignment $\tilde{z}_{r,c}$ to each cluster (of each class k) based on their Euclidean distance to the centroid locations. At each iterative step of the K-Means algorithm, the centroids are refined by integrating the adjusted assignments:

$$\begin{split} \tilde{\mathbf{p}}_{c} &= \frac{\sum_{i} f_{\phi}(x_{i}) z_{i,c} + \sum_{r} f_{\phi}(\tilde{x}_{r}) \tilde{z}_{r,c}}{\sum_{i} z_{i,c} + \sum_{r} \tilde{z}_{r,c}}, \\ \text{where} \quad \tilde{z}_{r,c} &= \frac{\exp(-d(f_{\phi}(\tilde{x}), \tilde{\mathbf{c}}_{c}))}{\sum_{c'} \exp(-d(f_{\phi}(\tilde{x}), \tilde{\mathbf{c}}_{c'}))} \end{split} \tag{5}$$

Although it is possible to perform multiple iterations of the clustering algorithm, the authors found that the performance does not improve after a single iteration.

a) Soft K-Means PN + Cluster: The Soft K-Means approach described above assumes that $\tilde{\mathcal{S}}$ contains the same classes as \mathcal{S} , but this is unlikely to be true in a practical scenario. Classes that are not part of \mathcal{S} are called distractors since they are likely to interfere with the refinement process. To make the method more robust to distractors, the authors introduce an extra cluster (K=k+1) that acts as a 'catchall' cluster for anything that does not belong to the classes of interest, and thus, preventing any distractors from hindering with the refinement. The authors place the cluster at the origin $(\tilde{\mathbf{p}}_c = \mathbf{0} \text{ for } c > n)$ and introduce a learnable length-scale parameter, q_c , that reflects the amount of within-class variation. Thus, the partial assignment is defined as:

$$\tilde{z}_{r,k} = \frac{\exp\left(-\frac{1}{q_k^2}d\left(f_{\phi}\left(\tilde{x}\right),\tilde{\mathbf{c}}_k\right) - A(q_k)\right)}{\sum_{k'}\exp\left(-\frac{1}{q_k^2}d\left(f_{\phi}\left(\tilde{x}\right),\tilde{\mathbf{c}}_{k'}\right) - A(q_{k'})\right)}$$
where $A(q) = log(q) + \frac{1}{2}log(2\pi)$

For simplicity, the authors set $q_{1...C}$ to 1 in their experiments and only learn the length-scale of the distractor cluster q_{n+1} . Our experiments follow the same setup.

b) Soft K-Means PN + Mask: The authors consider an alternative method to deal with distractor classes. Intuitively, a single distractor cluster is unlikely to not work well with higher numbers of distractor classes. To address these problems, instead of using a high-variance 'catch-all' cluster, an image is labeled as a distractor if its embedding does not lie within legitimate proximity of any of the class' prototypes. Specifically, the Soft K-Means refinement process is altered as follows. Firstly, the normalized distances, \tilde{d} , are computed between examples $\tilde{x}_r \sim \tilde{\mathcal{S}}$ and prototypes \mathbf{p}_c :

$$\tilde{d}_{r,c} = \frac{d_{r,c}}{\frac{1}{\tilde{M}} \sum_{j} d_{r,c}} \tag{7}$$

where $d_{r,c}=d\left(f_{\phi}(x_r),\mathbf{p}_c\right)=||f(\tilde{x}_r)-\mathbf{p}_c||_2^2$. Secondly, a small neural network computes learnable parameters β_c and γ_c from various statistics of the normalised distances (i.e. using the min, max, variance, skewness and kurtosis of $\tilde{d}_{r,c}$). The parameters β_c and γ_c help to establish how aggressively the unlabeled samples should influence centroids during the refinement process. The final refinement process of the *Soft K-Means PN + Mask* method is:

$$\tilde{\mathbf{p}}_{c} = \frac{\sum_{i} f_{\phi}(x_{i}) z_{i,c} + \sum_{r} f_{\phi}(\tilde{x}_{r}) \tilde{z}_{r,c} m_{r,c}}{\sum_{i} z_{i,c} + \sum_{r} \tilde{z}_{r,c} m_{r,c}},$$
where $m_{r,c} = \sigma \left(-\gamma_{c} \left(\tilde{d}_{r,c} - \beta_{c} \right) \right)$ (8)

where $m_{r,c}$ are the soft-masks computed by comparing the normalised distances to the learned thresholds.

2) Consitent Prototypical Network: Consistent Prototypical Networks (CPNs) [17] are also a semi-supervised FSL method capable of working with the original PN [15] and the K-Mean refined PN [16]. The authors use virtual adversarial training (VAT) [42], and random walk (RW) loss [43], [44] to formulate a loss function that drives the meta-training process:

$$\mathcal{L}_{SSL} = \mathcal{L}_{VAT} + \mathcal{L}_{RW} \tag{9}$$

Virtual adversarial training loss [42] works on the assumption of local consistency, also known as smoothness, that two data points which are close together should get similar labels. In other words, if we add small perturbations to a point, it should not change its label by much. The local consistency loss of a point is calculated independently of the other points. Inspired by previous work [43], [44], the authors of CPN introduce a global-consistency loss that considers all data points and the overall structure of the embedding manifold. Let us consider points in the embedding space forming graph structures based on their similarity where the probability of going from a point to another varies based on the distance between the points. A loss can be calculated through a random-walk over these similarity graphs constructed between unlabeled examples and the prototypes. The idea is that a random walker starting from a prototype should rarely cross the natural class decision boundaries, thus, explicitly promoting clustering. This can be achieved by allowing the random walker to take some fixed number of steps jumping between points, and maximizing the probability that the random walker gets back to the initial prototype within those steps.

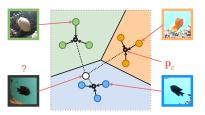


Fig. 2. Prototypical Network. Prototypes p_c are computed as the mean of the support samples belonging to a single class and mapped into an embedding space. A label for a target image is assigned based on the distances to the prototypes.

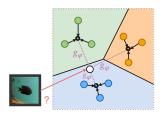


Fig. 3. Relation Network. Prototypes are computed in the same way as in Prototypical Networks. However, a label for a target image is assigned based on the score given by the relation module g_{φ} .

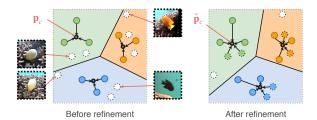


Fig. 4. Prototypical Network with K-Means refinement. Information from unlabeled samples (marked with dashed outlines) is incorporated into the prototypes by a single iteration of Soft K-Means. Some samples are omitted in the process due to their low proximity to any prototype.

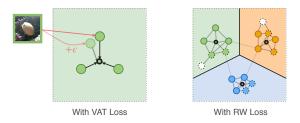


Fig. 5. Consistent Prototypical Networks (CPN). CPN works on top of Prototypical Networks with and without the K-Means refinement. The Cross-Entropy loss is replaced by Virtual Adversarial Training loss (VAT) and Random-Walk loss (RW). During meta-training, VAT adds a small perturbation ϵ to each support sample before mapping it into the embedding space and calculating the prototypes. The goal of RW is to construct a tight neighborhood of samples for each class. The idea is that a 'random walker' transverses similarity graphs between samples, and should rarely cross the natural class decision boundaries, thus, explicitly promoting clustering.

V. EXPERIMENTS

A. Meta-training

For each dataset, three meta-training scenarios were constructed: meta-training on Mini-ImageNet [18], meta-training on one of the underwater datasets, and meta-training using both datasets:

- 1) Meta-training on Mini-ImageNet. In the first set of experiments, we meta-trained models on the metatraining split of Mini-ImageNet, following the original papers' setup. Specifically, ordinary PN [15] was metatrained using 5-shot 15-way classification tasks, while all the other methods used 5-shot 5-way tasks. Semisupervised algorithms sampled additional 5 samples per class from the unlabeled partition of a relevant dataset split. All methods used 5 target images per class. The PN and Relation Networks were trained for 4×10^5 tasks but generally converged much sooner. Soft k-Means PNs were trained for 2×10^6 tasks but rarely improved beyond 5×10^5 tasks. CPNs were trained for 1.2×10^6 tasks. Evaluating the meta-testing split of Mini-ImageNet showed that our implementations achieved the within 3 accuracy points of the methods' claimed performances.
- 2) Meta-training on an underwater dataset. Similarly, we meta-trained the FSL models from random weight initialization on underwater datasets. We follow a similar setup as described above with a few notable changes to accommodate the smaller dataset sizes. Like other methods, ordinary PN [15] was trained using 5-shot 5-way classification to accommodate the lower number of classes in the meta-training split. The ordinary PN and Relation Networks were trained using 4×10^3 tasks, but we found that the algorithms generally converged much sooner. Soft k-Means PNs and CPNs were trained for 5×10^3 tasks.
- 3) **Meta-training on both datasets**. In this set of experiments, we pre-meta-trained the FSL models on the Mini-ImageNet dataset before meta-training on the underwater dataset. Specifically, we used the best meta-trained model on Mini-ImageNet (as described in point 1), and we further meta-trained it on the underwater dataset (as described in point 2 but without re-initialization).

B. Common evaluation and setup

All experiments follow the same evaluation setup. That is, throughout the meta-training process, the models were meta-validated after every few-thousand tasks, and the best model was saved based on the performance on the meta-validation dataset split. At the end of meta-training, the best model was meta-tested on 1000 FSL 5-shot 5-way tasks sampled from the meta-testing split. During a task, models used the support set and the previously acquired knowledge to classify target set samples. We repeated each experiment 10 times for each algorithm, dataset, and meta-training type. Our results show the average target set accuracy. Due to the low number of classes

in underwater datasets, we randomly picked resampled classes to be used for meta-training/meta-testing/meta-validation splits between each repeat. We reasoned that freezing the splits would create a bias towards specific FSL methods and create a skewed view of the methods' performance on the underwater dataset.

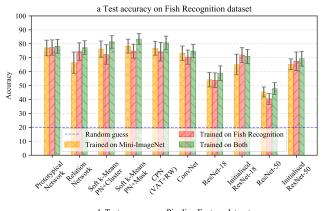
C. Network Architectures

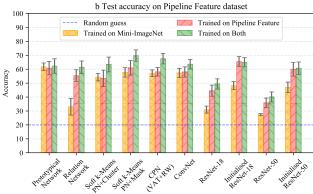
All FSL models used a vanilla convolutional neural network consisting of 4 convolutional blocks. Each block was composed of a convolutional layer (each with 3 by 3 receptive fields, 64 filters, stride 1, and padding 0) followed by batch normalization [45], ReLU activation functions, and maxpooling. ConvNet baseline followed the same setup. Relation Networks used a relation network consisting of two convolutional blocks followed by a linear layer with one output.

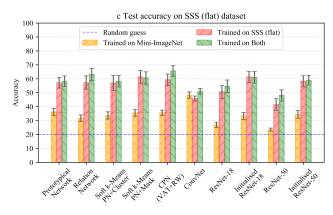
D. Fine-tuned Baselines

In addition to few-shot learning methods, we selected a few fine-tune baselines for comparison. These include a range of convolutional networks that replace the processes of meta-training (as described in subsection V-A) with pre-training. The pre-training is performed on the meta-training split of a dataset and then fine-tuning the last layers on the support sets during the meta-testing phases.

- a) ConvNet: We compared FSL methods with an equally powerful baseline model using the same 4 convolutional block architecture, and we called it ConvNet. The model contained an additional linear layer and a softmax over five output units (one for each class in the 5-way FSL task). The pre-training process was done over 4×10^5 mini-batches with batch size 64, and a learning rate of 0.001, slowly decaying at a rate of 0.9 after each 4×10^4 batches. After a few thousand mini-batches, the model was validated using FSL learning tasks, following the same meta-validation procedure as FSL models. Similarly, at the end of pre-training, the model was meta-tested on 1000 FSL 5-shot 5-way tasks sampled from the meta-testing dataset split. We performed fine-tuning by freezing all but the last linear layer of the baseline, which was randomly re-initialized, and fine-tuned on the support set (25 images for 5-way 5-shot task). The fine-tuning process performed 10 iterations with an initial learning rate of 0.01, and a rapid decay rate of 0.5 after each iteration. For each new evaluation task, we re-initialized the last layer with random weights.
- b) ResNets: Similarly to ConvNet, we pre-trained ResNet architectures. We used ResNet-18 and ResNet-50 trained from random weight initialisation. We also investigated versions of ResNets with a pre-trained set of weights that came with the PyTorch library, obtained from training on full-resolution ImageNet. We refer to these variants as Initialised ResNet-18 and Initialised ResNet-50. To accommodate the smaller images size 84 by 84 pixels in the ResNet architecture, we automatically turned off max-pooling layers. Like the ConvNet, we pre-trained the four ResNet baselines and then fine-tuned the models' last layers on the support set. We used a learning rate of 0.0001 to accommodate the higher number of trainable parameters.







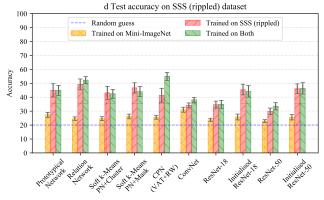


Fig. 6. Test accuracy of models on the meta-testing split of the underwater datasets after meta-/pre- training models on the meta-training split of Mini-ImageNet (yellow), the underwater dataset (red), and both datasets (green). The error bars show a 95% confidence interval.

TABLE I

ACCURACY ON TESTING SPLIT OF FISH RECOGNITION DATASET AFTER META-/PRE- TRAINING MODELS ON THE META-TRAINING SPLIT OF MINI-IMAGENET, FISH RECOGNITION, AND BOTH DATASETS.

Method	Mini-ImageNet	Fish Recognition	Both
Prototypical Network	77.0(±5.6)	77.3 (±5.5)	78.2(±5.0)
Relation Network	66.4(±7.7)	$74.4(\pm 6.3)$	77.3(±4.9)
Soft k-Means PN+Cluster	$76.2(\pm 5.8)$	$72.3(\pm 7.0)$	81.3(±4.5)
Soft k-Means PN+Mask	78.3 (±5.0)	$74.7(\pm 5.0)$	83.4 (±3.9)
CPN (VAT+RW)	$76.8(\pm 5.0)$	$74.3(\pm 6.7)$	80.7(±4.8)
ConvNet	$73.2(\pm 5.2)$	$70.4(\pm 5.1)$	74.8 (±4.8)
ResNet-18	54.2(±5.2)	$53.8(\pm 5.3)$	58.9(±5.2)
Initialised ResNet-18	65.1(±7.4)	$71.9(\pm 5.3)$	$70.9(\pm 5.1)$
ResNet-50	45.4(±3.6)	40.4 (±3.9)	47.8 (±4.3)
Initialised ResNet-50	65.3(±3.9)	$67.4(\pm 6.7)$	69.3 (±5.3)

TABLE II

ACCURACY ON TESTING SPLIT OF PIPELINE FEATURE DATASET AFTER META-/PRE- TRAINING MODELS ON THE META-TRAINING SPLIT OF MINI-IMAGENET, PIPELINE FEATURE, AND BOTH DATASETS.

Method	Mini-ImageNet	Pipeline Feature	Both
Prototypical Network	61.8 (±2.7)	60.8 (±4.8)	62.3(±5.2)
Relation Network	33.1(±6.0)	55.4(±4.2)	$61.4(\pm 4.5)$
Soft k-Means PN + Cluster	54.3 (±2.7)	53.4 (±5.9)	$63.4(\pm 5.3)$
Soft k-Means PN + Masking	57.7(±3.4)	$61.0(\pm 5.2)$	69.8 (±4.1)
CPN (VAT+RW)	57.1(±2.4)	$58.2(\pm 3.1)$	$67.6(\pm 3.8)$
ConvNet	57.3 (±3.3)	$58.0(\pm 3.7)$	$63.6(\pm 3.4)$
ResNet-18	$31.0(\pm 2.5)$	44.6 (±4.0)	49.4 (±3.7)
Initialised ResNet-18	48.2(±2.9)	65.5 (±3.6)	$65.0(\pm 3.3)$
ResNet-50	$27.4(\pm0.8)$	$35.9(\pm 3.4)$	40.1 (±3.6)
Initialised ResNet-50	47.0 (±3.8)	59.9 (±4.9)	$60.7(\pm 4.5)$

TABLE III

ACCURACY ON TESTING SPLIT OF SSS (FLAT) DATASET AFTER META-/PRE- TRAINING MODELS ON THE META-TRAINING SPLIT OF MINI-IMAGENET, SSS (FLAT), AND BOTH DATASETS.

Method	Mini-ImageNet	SSS (flat)	Both
Prototypical Network	36.2(±2.5)	57.2(±3.8)	58.3(±3.8)
Relation Network	$31.7(\pm 2.4)$	$57.3(\pm 4.8)$	$63.1(\pm 4.4)$
Soft k-Means PN + Cluster	$33.7(\pm 2.6)$	$57.0(\pm 5.5)$	$58.3(\pm 4.2)$
Soft k-Means PN + Masking	$35.5(\pm 2.4)$	61.4 (±4.6)	$60.6(\pm 4.4)$
CPN (VAT+RW)	$35.6(\pm 1.9)$	$59.3(\pm 4.1)$	65.6 (±3.8)
ConvNet	48.3 (±2.3)	$45.8(\pm 1.8)$	51.1 (±2.0)
ResNet-18	$26.9(\pm 1.9)$	$50.6(\pm 4.7)$	54.6 (±4.5)
Initialised ResNet-18	$33.3(\pm 2.5)$	$61.3(\pm 4.1)$	$61.0(\pm 4.2)$
ResNet-50	$23.5(\pm 1.1)$	$41.5(\pm 4.2)$	48.0 (±4.1)
Initialised ResNet-50	34.4 (±2.8)	$58.2 (\pm 4.0)$	59.0 (±3.5)

TABLE IV

ACCURACY ON TESTING SPLIT OF SSS (RIPPLED) DATASET AFTER META-/PRE- TRAINING MODELS ON THE META-TRAINING SPLIT OF MINI-IMAGENET, SSS (RIPPLED), AND BOTH DATASETS.

Method	Mini-ImageNet	SSS (rippled)	Both
Prototypical Network	27.3 (±1.8)	44.9 (±4.8)	44.9 (±3.6)
Relation Network	$24.6(\pm 1.3)$	49.2 (±3.9)	$52.3(\pm 2.5)$
Soft k-Means PN + Cluster	$24.6(\pm 1.4)$	43.1(±4.7)	$42.4(\pm 3.2)$
Soft k-Means PN + Masking	$26.2(\pm 1.7)$	$46.7(\pm 3.7)$	44.1(±3.6)
CPN (VAT+RW)	$25.6(\pm 1.4)$	41.4(±5.0)	54.9 (±2.9)
ConvNet	31.0 (±1.8)	34.1 (±1.8)	38.1 (±1.8)
ResNet-18	$23.8(\pm 1.1)$	$34.7(\pm 2.5)$	$34.8(\pm 2.9)$
Initialised ResNet-18	$25.8(\pm 2.1)$	$45.5(\pm 3.8)$	$44.2(\pm 4.2)$
ResNet-50	$22.9(\pm 1.0)$	$29.9(\pm 2.3)$	$33.5(\pm 2.6)$
Initialised ResNet-50	25.6(±1.9)	46.1(±3.5)	46.4 (±4.1)

VI. RESULTS

The results are presented in Tables I-IV, and Figure 6. Our experiments aim to answer the following questions:

- Does meta-training on a general-purpose dataset generalize to underwater datasets?
- Is there any advantage in pre-meta-training?
- Do FSL methods offer any advantage over traditional fine-tuning methods?
- What is state-of-the-art on underwater optical and sonar datasets?

For extra clarity in places, we will refer to the three metatraining scenarios by the numbers given in section V-A.

A. Generalization of Mini-ImageNet-trained models

FSL models meta-trained on Mini-ImageNet alone (scenario #1) achieved an average of 75.0% and 52.8% accuracy on Fish Recognition and Pipeline Features, respectively. On flat and rippled seabed sonar datasets, they achieved an average of 34.5% and 25.7% accuracy, respectively. These results confirm that the more the test dataset's style deviates from Mini-ImageNet, the worse the generalization of Mini-ImageNet trained models.

Compared to the other two meta-training scenarios (scenario #2: training on underwater datasets; and scenario #3: training on both dataset), we observe an overall difference of -11% and -16% between scenario #1 and #2, and scenario #1 and #3, respectively - the only situation where scenario #1 does better than scenario #2 is on the Fish Recognition dataset. The better performance can be attributed to the similarity of Mini-ImageNet to the Fish Recognition dataset as well as the increased number of samples and classes in the Mini-ImageNet, which allowed the models to learn more generalizable features and achieve an overall higher accuracy.

Furthermore, we observe that scenario #1 models generalize poorly to sonar, with an average difference of -21.2% accuracy points compared to the other two meta-training scenarios. In the more difficult sonar setting (with rippled seabed), the average model performance is only slightly better than random, reflecting the challenges caused by the significant style shift between optical and sonar images. It shows that general-purpose datasets alone are insufficient to meta-train few-shot learning models where the style of images differs significantly from the meta-testing split.

B. Advantages of pre-meta-training

For 17 out of 20 settings across all five FSL methods and four datasets, we observe that it is at least as good to metatrain models on both datasets (scenario #3) as training with either of the other two scenarios. Overall, we observe an average improvement of 3.9% accuracy points over the other two scenarios' best models - an average advantage of 5.2% for optical images and 2.7% for sonar images. Across the meta-training scenarios, we observe an average improvement of 16.5% and 4.5% over equivalent methods from scenario #1 and #2, respectively. This result demonstrates that there can be many gains of pre-meta-training on readily available datasets,

even if the style of images differs from that of the metaevaluation target set - a similar trend is observed in classical transfer learning approaches [26].

Interestingly, despite the significant differences in image style, the most substantial improvement of pre-meta-training (scenario #3) can be observed for CPN on SSS (rippled) with 13.5% improvement over the best of other two training scenarios. Although we observed poor generalization of Mini-ImageNet trained models in scenario #1, the results of scenario #3 show that some high-level features are still useful and can be utilized successfully during a later meta-training phase.

In this study, we explored one way of meta-training on both datasets; meta-training once on ImageNet, then meta-training again on a specialized dataset. However, mixing datasets into a single dataset could be another way of combining them. However, we leave this investigation for future work.

C. Advantages of Few-Shot Learning methods

Comparing FSL methods with fine-tuned baselines, across all datasets and meta-/pre- training scenarios, we observe that in 10 out of 12 settings, there is at least one few-shot learning model that achieves at least as good performance as the equally powerful ConvNet baseline. We observe an average improvement of 7% accuracy points using FSL methods over the ConvNet baseline, with up to 8.6% on optical datasets and up to 16.8% on sonar.

The FSL models can even achieve at least as good performance as the more powerful ResNet-18 and ResNet-50 baselines in 11 out of 12 settings. Interestingly, the pre-trained ResNet-50 (from random initialization) sometimes performed worse than the less powerful ResNet-18, which could be due to overfitting caused by the increased number of trainable parameters. The Initialized ResNets (pre-trained on full-scale ImageNet) performed overall the best out the baseline models.

On sonar datasets in scenario #1, the ConvNet baseline does the best out of all of the methods, outperforming the FSL methods and the more powerful ResNet baselines. We theorize that this superior performance could be due to the adaptation ability of the fine-tuning process. Adjusting the network's weights using the support set has some advantage over the nontunable meta-evaluation process of the Prototypical Networks and variants. It could be interesting to investigate optimizationbased FSL methods; however, we leave this for future work. Although the more powerful ResNet baselines also performed fine-tuning, their performance was inferior to ConvNet on the sonar datasets in scenario #1. It is likely that the ResNet networks, which contain many more convolutional layers, learn color-dependent features early in the network, which may impede the process of fine-tuning on sonar images. In contrast, ConvNet is a much shallower network, and the final network layer is more likely to contain high-level features that can easily be fine-tuned to the style of sonar images.

In some experiments, ResNet baselines do better than the FSL baselines. However, these models should not be directly compared since the underlying architecture of FSL methods is similar to the ConvNet that contains less trainable parameters

and has a shallower architecture. For example, we found that ConvNet contained 1.3×10^5 trainable parameters, whereas were 1.2×10^7 parameters in ResNet-18 and 2.6×10^7 in ResNet-50, which is at two orders of magnitude greater. It would be interesting to substitute FSL models with a more powerful architecture. Work by [46] shows that using a more powerful backbone model in Prototypical Network significantly improves its performance. However, we leave this investigation to future work.

D. State-of-the-art FSL on underwater datasets

Soft K-Means PN achieves the best performance on Fish Recognition and Pipeline Feature, with 83.4% and 69.8% accuracy in scenario #3, respectively - an improvement of 8.6% and 6.8% points over the ConvNet baseline model. On sonar datasets, CPN achieves the best performance with 65.6% and 54.9% accuracy on flat and rippled seabed in scenario #3 - offering 14.5% and 16.8% point improvement over the ConvNet baseline.

Generally, when meta-trained on both datasets, best semisupervised methods tend to do slightly better than the best fully supervised FSL methods on the same dataset, with an average improvement of 6.4% accuracy points on optical and 2.5% on sonar, across all three meta-training scenarios. Interestingly, semi-supervised methods achieved better performance to fully supervised methods even though they only used 40% of the labels. Their advantage could be due to at least two factors. On the one hand, the presence of 5 additional unlabeled samples per class exposes the algorithm to more information, which it could utilize when learning about new classes. On the other, the prototype refinement process could result in a more accurate representation of a classes' mean. Our supplementary experiments, on Soft K-Means PN models with no additional unlabeled samples, suggest that most of the performance gain is attributed to the post-processing of feature vectors, rather than the existence of additional data. In some experiments, additional data produced worse performance. However, more experiments would need to be collected to offer a more thorough insight, and we leave this investigation for future work.

VII. DISCUSSION

Throughout the previous sections, we have seen FSL methods performing well on underwater optical and sonar images. In this section, we would like to highlight some limitations of our results as well as FSL methods in general that require further consideration before applying these methods on real-world robotics applications.

Few-shot learning methods work under a strict set of assumptions that might make them challenging to apply to practical settings. Firstly, achieving strong performance significantly depends on the choice of the support set. In some of our experiments, we found that the support set's choice was essential for capturing the intraclass differences. Moreover, FSL benchmarks typically assume that the support set is sampled uniformly from a single distribution. However, in

real-world applications, the support set is likely to become available incrementally over time, contain a varying number of samples per class, and come from a highly correlated video frame stream.

Moreover, in this work, the FSL methods were examined on a constrained classification problem where objects were present in the center of images. In a practical situation, the FSL classification models are likely to function on top of automatic target recognition (ATR) systems. The ATR system is likely to output a range of regions with various scales and objects placed anywhere within. Further considerations are required to apply FSL to work with ATR systems.

Finally, the FSL methods examined assume the accessibility of all $k \times n$ images at once, with no future updates. In parallel work, we already investigate FSL methods in a general continual learning setting where the algorithms are exposed to new samples a small batch at a time [47]. However, more consideration is needed for learning with underwater images, as reflected in our experiments.

VIII. CONCLUSION

In this work, we investigated few-shot learning (FSL) methods on four underwater datasets. For each method, we compared three meta-training scenarios: meta-training on a general-purpose dataset (Mini-ImageNet), on an underwater dataset, and both datasets. In 10 out of 12 scenario-dataset combinations, FSL methods achieved at least as good performance as equally powerful baseline models, offering an average improvement of 7% accuracy points, with up to 9% and 17% on optical and sonar, respectively. Further, we found that meta-training on both datasets produced the best performance - an average improvement of 16.5% and 4.5% over meta-training on Mini-ImageNet alone and meta-training on underwater dataset alone, respectively. In our experiments, the semi-supervised FSL models performed slightly better than the supervised FSL models offering an average improvement of 4%. In future work, we plan to reduce some unrealistic assumptions made by FSL methods (e.g., introduce incremental updates) and investigate these methods working alongside an automatic target recognition system to develop a few-shot object detector.

ACKNOWLEDGEMENT

We want to give a special thanks to Antti Karjalainen for generating the simulated side-scan sonar data. This work was supported by the EPSRC Centre for Doctoral Training in Robotics and Autonomous Systems, funded by the UK Engineering and Physical Sciences Research Council and SeeByte Ltd (Grant No. EP/S515061/1).

REFERENCES

[1] M. Moniruzzaman, S. M. S. Islam, M. Bennamoun, and P. Lavery, "Deep learning on underwater marine object detection: A survey," in *In International Conference on Advanced Concepts for Intelligent Vision Systems, Springer, Cham*, 2017, vol. 2.

- [2] J. Rutledge, W. Yuan, J. Wu, S. Freed, A. Lewis, Z. Wood, T. Gambin, and C. Clark, "Intelligent shipwreck search using autonomous underwater vehicles," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018.
- [3] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, 2015.
- [4] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," *Proceedings of the 34th International Conference on Machine Learning (ICML)*, vol. 70, 2017.
- [5] M. Tan and Q. V. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," *Proceedings of the 36th International Conference on Machine Learning (ICML)*, 2019.
- [6] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision*, vol. 115, pp. 211–252, 2015.
- [7] D. C. Plaut, "Experiments on learning by back propagation." Carnegie-Mellon Univ., Pittsburgh, Pa. Dept. of Computer Science., 1986.
- [8] K. J. Lang and G. E. Hinton, "Dimensionality reduction and prior knowledge in e-set recognition," Advances in Neural Information Processing Systems 2 (NIPS), 1989.
- [9] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," arXiv preprint arXiv:1207.0580, 2012.
- [10] A. Labach, H. Salehinejad, and S. Valaee, "Survey of dropout methods for deep neural networks," arXiv preprint arXiv:1904.13310, 2019.
- [11] P. Y. Simard, D. Steinkraus, and J. C. Platt, "Best practices for convolutional neural networks." In Proceedings of the International Conference on Document Analysis and Recognition (ICDAR), 2003.
- [12] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, 2010.
- [13] J. Kukačka, V. Golkov, and D. Cremers, "Regularization for deep learning: A taxonomy," arXiv preprint arXiv:1710.10686, 2017.
- [14] W. Y. Chen, Y. C. F. Wang, Y. C. Liu, Z. Kira, and J. B. Huang, "A closer look at few-shot classification," 7th International Conference on Learning Representations, ICLR 2019, 2019.
- [15] J. Snell, K. Swersky, and R. S. Zemel, "Prototypical networks for fewshot learning," Advances in Neural Information Processing Systems 30 (NIPS), 2017.
- [16] M. Ren, E. Triantafillou, S. Ravi, J. Snell, K. Swersky, J. B. Tenenbaum, H. Larochelle, and R. S. Zemel, "Meta-learning for semi-supervised few-shot classification," 6th International Conference on Learning Representations (ICLR), 2018.
- [17] A. Ayyad, N. Navab, M. Elhoseiny, and S. Albarqouni, "Semi-supervised few-shot learning with local and global consistency," *International Journal of Computer Mathematics*, vol. 91, 2019.
- [18] S. Ravi and H. Larochelle, "Optimization as a model for few-shot learning," 5th International Conference on Learning Representations (ICLR), 2017.
- [19] R. Zhang, T. Che, Z. Ghahramani, Y. Bengio, and Y. Song, "Metagan : An adversarial approach to few-shot learning," *Advances in Neural Information Processing Systems 31 (NIPS)*, vol. 31, 2018.
- [20] M. Patacchiola, J. Turner, E. J. Crowley, M. O'Boyle, and A. Storkey, "Deep Kernel Transfer in Gaussian Processes for Few-shot Learning," arXiv preprint arXiv:1910.05199, 2019.
- [21] O. Vinyals, C. Blundell, T. Lillicrap, K. Kavukcuoglu, and D. Wierstra, "Matching networks for one shot learning," Advances in Neural Information Processing Systems 29 (NIPS), 2016.
- [22] I. Leonard, A. Arnold-Bos, and A. Alfalou, "Interest of correlation-based automatic target recognition in underwater optical images: theoretical justification and first results," in *Proceedings of SPIE - The International* Society for Optical Engineering, 2010.
- [23] W. Xu and S. Matzner, "Underwater Fish Detection using Deep Learning for Water Power Applications." 5th Annual Conference on Computational Science & Computational Intelligence (CSCI), 2018.
- [24] T. Rimavicius and A. Gelzinis, "A Comparison of the Deep Learning Methods for Solving Seafloor Image Classification Task," In International Conference on Information and Software Technologies, Springer, Cham, vol. 319, 2017.
- [25] D. Levy, Y. Belfer, E. Osherov, E. Bigal, A. P. Scheinin, H. Nativ, D. Tchernov, T. Treibitz, A. King, and S. M. Bhandarkar, "Automated Analysis of Marine Video With Limited Data," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops* (CVPR workshops), vol. 1, 2018.

- [26] A. B. Tamou, A. Benzinou, K. Nasreddine, and L. Ballihi, "Underwater Live Fish Recognition by Deep Learning." Springer International Publishing, 2018, vol. 1.
- [27] I. Yoon, S. Jeong, J. Jeong, D. Seo, and J. Paik, "Wavelength-adaptive dehazing using histogram merging-based classification for UAV images," *Sensors (Switzerland)*, vol. 15, 2015.
- [28] P. Sahu, N. Gupta, and N. Sharma, "A Survey on Underwater Image Enhancement Techniques," *International Journal of Computer Applica*tions, vol. 87, 2014.
- [29] D. Berman, D. Levy, S. Avidan, and T. Treibitz, "Underwater Single Image Color Restoration Using Haze-Lines and a New Quantitative Dataset," arXiv preprint arXiv:1811.01343, 2018.
- [30] J. Lu, N. Li, S. Zhang, Z. Yu, H. Zheng, and B. Zheng, "Multi-scale adversarial network for underwater image restoration," *Optics & Laser Technology*, vol. 110, 2019.
- [31] F. Ferreira, D. Machado, G. Ferri, S. Dugelay, and J. Potter, "Underwater optical and acoustic imaging: A time for fusion? a brief overview of the state-of-the-art," in *IEEE OCEANS*, 2016.
- [32] S. Lee, B. Park, and A. Kim, "Deep Learning from Shallow Dives: Sonar Image Generation and Training for Underwater Object Detection," arXiv preprint arXiv:1810.07990, 2018.
- [33] C. Barngrover, R. Kastner, and S. Belongie, "Semisynthetic Versus Real-World Sonar Training Data for the Classification of Mine-Like Objects," *IEEE Journal of Oceanic Engineering*, vol. 40, 2015.
- [34] L. Paull, S. Saeedi, M. Seto, and H. Li, "AUV Navigation and Localization: A Review," *IEEE Journal of Oceanic Engineering*, vol. 39, 2014.
- [35] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2016.
- [36] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," Advances in Neural Information Processing Systems 28 (NIPS), 2015.
- [37] Y. Chen, Q. Ma, J. Yu, and T. Chen, "Underwater acoustic object discrimination for few-shot learning," in 2019 4th International Conference on Mechanical, Control and Computer Engineering (ICMCCE), 2019.
- [38] G. Koch, R. Zemel, and R. Salakhutdinov, "Siamese neural networks for one-shot image recognition," in *ICML deep learning workshop*, vol. 2, 2015
- [39] R. B. Fisher, K.-T. Shao, and Y.-H. Chen-Burger, "Overview of the fish4knowledge project," in *Springer International Publishing*, 2016.
- [40] A. I. Karjalainen, R. Mitchell, and J. Vazquez, "Training and Validation of Automatic Target Recognition Systems using Generative Adversarial Networks," in *IEEE Sensor Signal Processing for Defence Conference* (SSPD), 2019.
- [41] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. S. Torr, and T. M. Hospedales, "Learning to compare: Relation network for few-shot learning," *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.
- [42] T. Miyato, S.-I. Maeda, S. Ishii, and M. Koyama, "Virtual adversarial training: A regularization method for supervised and semi-supervised learning," *IEEE Transactions on Pattern Analysis and Machine Intelli*gence, 2018.
- [43] K. Kamnitsas, D. C. Castro, L. L. Folgoc, I. Walker, R. Tanno, D. Rueckert, B. Glocker, A. Criminisi, and A. Nori, "Semi-supervised learning via compact latent space clustering," arXiv preprint arXiv:1806.02679, 2018.
- [44] P. Häusser, A. Mordvintsev, and D. Cremers, "Learning by association a versatile semi-supervised training method for neural networks," Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition (CVPR), vol. 30, 2017.
- [45] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," *International Conference on Machine Learning (ICML)*, 2015.
- [46] B. N. Oreshkin, P. Rodriguez, and A. Lacoste, "Tadam: Task dependent adaptive metric for improved few-shot learning," Advances in Neural Information Processing Systems 31 (NIPS), 2018.
- [47] A. Antoniou, M. Patacchiola, M. Ochal, and A. Storkey, "Defining Benchmarks for Continual Few-Shot Learning," arXiv preprint arXiv:2004.11967, 2020.