




OPEN

# A realistic fish-habitat dataset to evaluate algorithms for underwater visual analysis

Alzayat Saleh<sup>1,4</sup>, Issam H. Laradji<sup>2,3,4</sup>, Dmitry A. Konovalov<sup>1</sup>, Michael Bradley<sup>1</sup>, David Vazquez<sup>3</sup> & Marcus Sheaves<sup>1</sup>

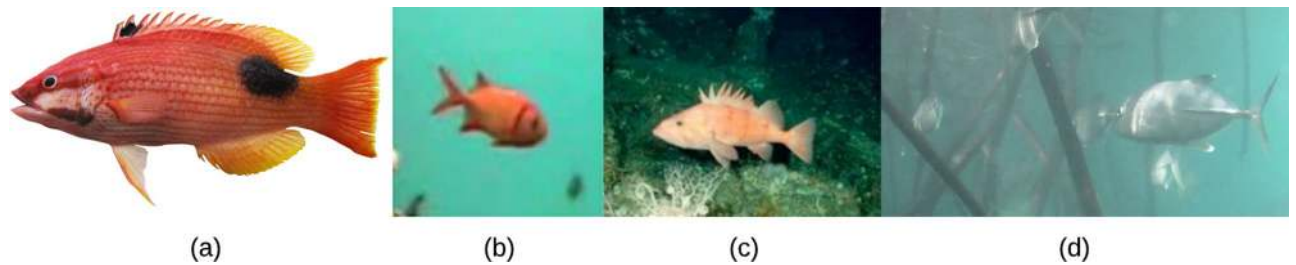
Visual analysis of complex fish habitats is an important step towards sustainable fisheries for human consumption and environmental protection. Deep Learning methods have shown great promise for scene analysis when trained on large-scale datasets. However, current datasets for fish analysis tend to focus on the classification task within constrained, plain environments which do not capture the complexity of underwater fish habitats. To address this limitation, we present DeepFish as a benchmark suite with a large-scale dataset to train and test methods for several computer vision tasks. The dataset consists of approximately 40 thousand images collected underwater from 20 habitats in the marine-environments of tropical Australia. The dataset originally contained only classification labels. Thus, we collected point-level and segmentation labels to have a more comprehensive fish analysis benchmark. These labels enable models to learn to automatically monitor fish count, identify their locations, and estimate their sizes. Our experiments provide an in-depth analysis of the dataset characteristics, and the performance evaluation of several state-of-the-art approaches based on our benchmark. Although models pre-trained on ImageNet have successfully performed on this benchmark, there is still room for improvement. Therefore, this benchmark serves as a testbed to motivate further development in this challenging domain of underwater computer vision.

Monitoring fish in their natural habitat is an important step towards sustainable fisheries. In the New South Wales state of Australia, for example, fisheries is valued at more than 100 million Australian dollars in 2012–2013<sup>14</sup>. Effective monitoring can provide information about which areas require protection and restoration to maintain healthy fish populations for both human consumption and environmental protection. Having a system that can automatically perform comprehensive monitoring can significantly reduce labour costs and increase efficiency. The system can lead to a large positive sustainability impact and improve our ability to maintain a healthy ecosystem.

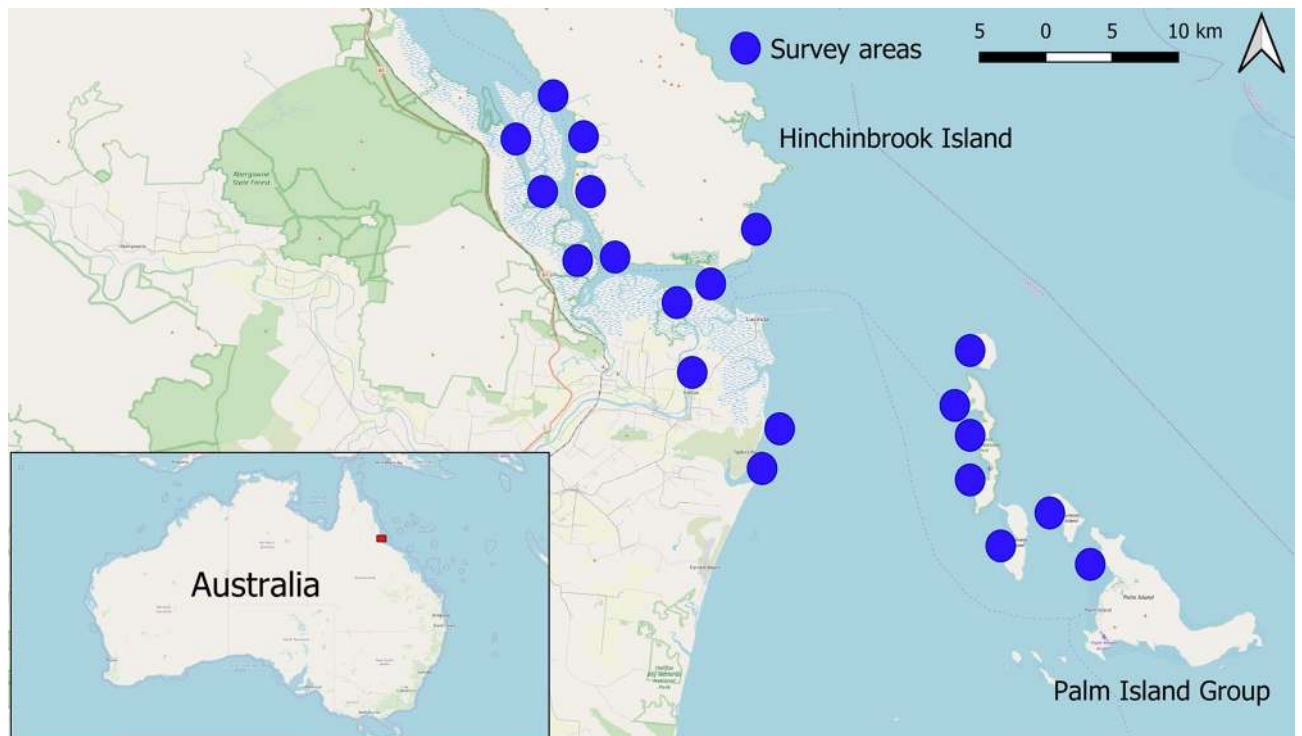
Deep learning methods have consistently achieved state-of-the-art results in image analysis. Many methods based on deep neural networks achieved top performance for a variety of applications, including, ecological monitoring with camera trap data. One reason behind this success is that these methods can leverage large-scale, publicly available datasets such as ImageNet<sup>6</sup> and COCO<sup>24</sup> for training before being fine-tuned for a new application.

A particularly challenging application involves automatic analysis of underwater fish habitats which demands a comprehensive, accurate computer vision system. Thus, considerable research efforts have been put towards developing systems for the task of understanding complex marine environments and distinguishing between a diverse set of fish species, which are based on publicly available fish datasets<sup>1,3,8,15,35</sup>. However, these fish datasets are small and do not fully capture the variability and complexity of real-world underwater habitats which often have adversarial water conditions, high similarity of the appearance between fish and some elements in the background such as rocks, and occlusions between fish. For example, the QUT fish dataset<sup>1</sup> contains only 3,960 labelled images of 468 species. Many of these fish images are taken in controlled environments where the background is plain white and the illumination is carefully adjusted (Fig. 1a). Similarly, underwater images

<sup>1</sup>James Cook University, Townsville, Australia. <sup>2</sup>University of British Columbia, Vancouver, Canada. <sup>3</sup>Element AI, Montreal, Canada. <sup>4</sup>These authors contributed equally: Alzayat Saleh and Issam H. Laradji. ✉email: alzayat.saleh@my.jcu.edu.au



**Figure 1.** A comparison of fish datasets. (a) QUT<sup>1</sup>, (b) Fish4Knowledge<sup>8</sup>, (c) Rockfish<sup>35</sup>, and (d) our proposed dataset DeepFish. (a–c) Datasets are acquired from constrained environments, whereas *DeepFish* has more realistic and challenging environments. (Figures a–c were obtained from the open-source datasets<sup>1,8,35</sup>).



**Figure 2.** Locations where the DeepFish images were acquired. Most of DeepFish has been acquired from the Hinchinbrook/Palm Islands region in North Eastern Australia, the rest from Western Australia (not shown in the map). (The map was created using QGIS version 3.8, which is available at <https://qgis.org>).

collected for the Fish4Knowledge<sup>8</sup> and Rockfish<sup>35</sup> datasets are cropped to have a single fish shown at the center (Fig. 1b,c), which requires costly human labor to produce and does not help models learn to recognise fish in the wild. Thus, the limitations of these datasets can inhibit further progress in building systems for comprehensive visual understanding of underwater environments.

To this end, we propose DeepFish as a benchmark that includes a dataset based on in-situ field recordings of fish habitats<sup>4</sup> and we tailor it towards analyzing fish in underwater marine environments. The dataset consists of approximately 40 thousand high-resolution ( $1,920 \times 1,080$ ) images collected underwater from 20 different marine habitats in tropical Australia (see Fig. 1d for an example image). These represent the breadth of different coastal and nearshore benthic habitats commonly available to fish species in tropical Australia<sup>32</sup> (Fig. 2).

Further, we go beyond the original classification labels by also acquiring point-level and semantic segmentation labels for additional computer vision tasks. These labels allow models to learn to analyze fish habitats from several perspectives, including, understanding fish dynamics, monitoring their count, and estimating their sizes and shapes. We evaluate state-of-the-art methods on these labels to analyze the dataset characteristics and establish initial results for this benchmark.

Overall, we can summarize our contributions as follows; (1) we present a benchmark that includes a dataset that captures the complexity and diversity of underwater fish habitats compared to previous fish datasets, (2) we incorporate additional labels to allow for a more comprehensive analysis of fish in underwater environments, (3) we show the importance of having pretrained models for achieving good performance on the benchmark,



**Figure 3.** DeepFish image samples across 20 different habitats.

and (4) we provide results that can serve as reference for evaluating new methods. The dataset and the code have been made public to help spark progress in developing systems for analysing fish habitats.

### Dataset

Our goal is to design a benchmark that can enable significant progress in fish habitat understanding. Thus we carefully look into the quality of data acquisition, preparation, and annotation protocol.

Accordingly, we start with the dataset based on the work of Bradley and colleagues<sup>4</sup> as it consists of a large number of images (around 40 thousand) that capture high variability of underwater fish habitats. The dataset's diversity and size makes it suitable for training and evaluating deep learning methods. However, the dataset's original purpose was not to evaluate machine learning methods. It was to examine the interactive structuring effects of local habitat characteristics and environmental context on assemblage composition of juvenile fish.

Yet the characteristics of the dataset makes it suitable as a machine learning benchmark. We tailor it to make the dataset a more comprehensive testbed to spark new, specialized algorithms in this problem setup and name the dataset as *DeepFish*. In the following sections we discuss how the data was collected, the additional annotations acquired for the dataset, how it was split between training, validation and testing, and how the dataset compares with current fish datasets.

**Data collection.** Videos for DeepFish were collected for 20 habitats from remote coastal marine environments of tropical Australia (Fig. 2). These videos were acquired using cameras mounted on metal frames, deployed over the side of a vessel to acquire video footage underwater. The cameras were lowered to the seabed and left to record the natural fish community, while the vessel maintained a distance of 100 m. The depth and the map coordinates of the cameras were collected using an acoustic depth sounder and a GPS, respectively. Video recording were carried out during daylight hours, and in relatively low turbidity periods. The video clips were captured in full HD resolution ( $1,920 \times 1,080$  pixels) from a digital camera. In total, the number of video frames taken is 39,766 and their distribution across the habitats are shown in Table 2. Examples of these video frames are shown in Fig. 3 which illustrate the diversity between the habitats.

This method of acquiring images is a low-disturbance technique that allows us to accurately assess fish-habitats associations in challenging, even inaccessible environments<sup>4</sup>. In contrast, many existing monitoring techniques used to understand fish habitats suffer from the problem of fish flight response, especially for habitats with limited visibility<sup>34</sup>. For example, a common surveying technique requires divers to conduct visual census<sup>2</sup>, which can cause disturbance to the fish, leading to inaccurate assessment of the fish community. Furthermore, divers cannot access areas with predators such as crocodiles with this technique. Other surveying techniques involve netting<sup>33</sup> and trawling<sup>30</sup> for catching and counting fish. However, these methods are invasive and interfere with the behaviour of the fish which can lead to inaccurate estimates. Further, they are limited to estimating fish count only. On the other hand, the data collection procedure for *DeepFish* is one of the most efficient methods for capturing a realistic, unaltered view of fish habitats associations<sup>4</sup>.

Habitats	FishClf	FishLoc	FishSeg
Low complexity reef	4,977	357	77
Sandy mangrove prop roots	4,162	322	42
Complex reef	4,018	190	16
Seagrass bed	3,255	328	16
Low algal bed	2,795	282	17
Reef trench	2,653	187	48
Boulders	2,314	227	16
Mixed substratum mangrove	2,139	177	28
Rocky Mangrove—prop roots	2,119	211	27
Upper Mangrove	2,101	129	21
Rock shelf	1,848	186	19
Mangrove	1,542	157	33
Sparse algal bed	1,467	0	0
Muddy mangrove	1,117	113	79
Large boulder and pneumatophores	900	91	37
Rocky mangrove—large boulder	560	57	28
Bare substratum	526	55	32
Upper mangrove	475	49	28
Large boulder	434	45	27
Muddy mangrove	364	37	29
Total	39,766	3,200	620

**Table 1.** *DeepFish* dataset statistics. Number of images annotated for each sub-dataset: FishClf for classification, FishLoc for counting/localization, and FishSeg for semantic segmentation.

**Additional annotations.** The original labels of the dataset are only suitable for the task of classification. These labels were acquired for each video frame, and they indicate whether an image has fish or not (regardless of the count of fish). These labels can be useful to train models for analyzing a fish utilization estimate between different habitats<sup>19</sup>. For example, classifying images between those that contain and do not contain fish allows biologists and ecologists to focus their efforts by analyzing only those images with the fish. However, they do not allow for a more detailed analysis of the habitats.

To address this limitation, we acquired point-level and semantic segmentation labels to enable models to learn to perform the computer vision tasks such as object counting, localization and segmentation. Point-level annotations are provided as a single click on each fish as shown in Fig. 5b, and segmentation labels as boundaries over the fish instances Fig. 5d. We describe them in detail in the following sections.

*Point-level annotations.* The goal of these annotations is to enable models to learn to perform fish counting. A useful application of this task is to automatically monitor fish population in order to avoid the risk of overfishing. These annotations also enable the task of localizing fish within each image which can be used for fish tracking and fish dynamics analysis.

We annotated 3,200 images with point-level annotations which we acquired across different habitats as shown in Table 1. These annotations represent the  $(x, y)$  coordinates of each fish within the images and they are placed around the centroid of the corresponding fish (Fig. 5b). These annotations were acquired using Labelme<sup>31</sup>, which is an open-source image annotation tool. It took approximately 1 second per fish with a labeler who is familiar with fish habitats. Since there is an average of 7 fish in each image, the annotation time is estimated at 7 s per image. Thus, this labeling scheme makes it easy to acquire additional annotations for new images and fish classes.

*Per-pixel annotations.* The goal of these annotations is to train and evaluate models to segment fish across images. As a result, the segmentation output can be used to estimate fish sizes, shapes, and their weight as shown in<sup>18,20</sup>. These are important statistics that can be useful in applications like commercial trawling<sup>10</sup>.

We collected per-pixel labels for 620 images. We labeled the fish using layered polygons in order to distinguish between pixels that belong to fish and those to the background (Fig. 5c). The pixel labels represent the size and shape of the fishes in the image. We used Lear<sup>17</sup> to extract these segmentation masks, an open-source image annotation tool commonly used for obtaining segmentation labels. Acquiring per-pixel labels is vastly more time-consuming than point-level annotations. It took around 2 min to label a single fish, to ensure quality masks we multiplied the manually generated masks with the original images to visually check the quality of the segmentation. In total, it took around 25 h to acquire segmentation labels for 310 valid images out of 620 images which is around 5 min per image. We acquired labels for a variety of habitats as shown in Table 1. We see that no point-level nor per-pixel labels were collected for “Sparse algal bed”. The reason is that the videos taken for



Dataset	# images	Tasks	# fish/image	Resolution	# habitats	Environment type	Has background images
DeepFish (ours)	39,766	Clf, Cnt, Loc, Seg	~ 7	1,920 × 1,080	20	In-situ	Yes
QUT <sup>1</sup>	3,960	Clf	1	480 × 360	N/A	30% in-situ, 70% controlled	No
Fish4Knowledge <sup>8</sup>	27,370	Detection	~ 1	352 × 240	N/A	In-situ (cropped at center)	No
Rockfish <sup>35</sup>	4,307	Detection	~ 1	1,280 × 720	N/A	In-situ (cropped at center)	Yes

**Table 2.** Comparison between dataset characteristics. Clf, Cnt, Loc, Seg refer to the tasks of classification, counting, localization and segmentation. “in-situ” datasets consist of unaltered images that capture the fish underwater in their natural habitat, whereas “controlled” consist of post-processed fish images where background and illumination have been altered. “in-situ (cropped at center)” datasets have images cropped at the center where the fish is.

the habitat shows hundreds of tiny fish in each frame where many of them are occluded and are indistinguishable from debris and tiny rocks. As a consequence, it is difficult to annotate a single image for localization and segmentation.

**Dataset splits.** We define a sub-dataset for each computer vision task: *FishClf* for classification, *FishLoc* for counting and localization, and *FishSeg* for the segmentation task. For each sub-dataset, we split the annotated images into training, validation, and test sets. Instead of splitting the data completely at random, we consider each split to represent the variability of different fish habitats and to have similar fish population size. Concretely, we first divide each habitat into images with no fish (background) and images with at least one fish (foreground). We randomly select 50% training, 20% for validation and 30% for testing for each habitat while ensuring that the number of background and foreground images are equal between them. Finally, we aggregate the selected training images from each habitat into one training split for the dataset. We do the same for the validation and testing splits.

As a result, we get a unique split consisting of 19,883, 7,953, 11,930 (training, validation and test) for *FishClf*, 1,600, 640, 960 for *FishLoc*, and 310, 124, 186 for *FishSeg*. While all the annotations, including for the test images, are made available, the expected evaluation setup is to select the best model on the validation set and perform a single evaluation on the test set. The reported results on the test set are then presented in a leaderboard to compare between the algorithms.

**Comparison to other datasets.** We compare *DeepFish* to other datasets in terms of (i) dataset size (ii) visual complexity and (iii) vision tasks. Many datasets exist for fish analysis<sup>3,8,9,15</sup>. But we chose those that are most similar to ours, namely, QUT<sup>1</sup>, Rockfish<sup>35</sup>, and Fish4Knowledge<sup>8</sup>.

Table 2 shows that *DeepFish* is the largest dataset with images of highest resolution. Unlike other datasets, *DeepFish* images capture a wide view of the underwater fish habitats. The images also represent a diverse set of numerous habitats, and different underwater conditions. Further, *DeepFish* images are in-situ as they are extracted directly unaltered from the underwater camera. These images can also contain several fish that are potentially occluded and overlapping. In contrast, QUT images are post-processed. Most of the images in the QUT dataset are captured in “controlled” conditions, that is, the image collector spread the fish fins and captured the fish image against a constant background with controlled illumination then annotated all the images by drawing a tight red bounding box around the fish body. Fish4Knowledge and Rockfish images are taken in the fish natural habitat but they are also post-processed as they are cropped to ensure fish are at the center of the image (see Fig. 1 for a comparison between the images from each dataset). Thus, *DeepFish* is more suitable for training models for the purpose of analyzing fish in the wild, and it requires less effort for collecting additional images and annotations.

The task that the other datasets address is limited to classification where the goal is to distinguish between fish species. Fish4Knowledge and Rockfish also address the task of detection where the goal is to draw a bounding box around the fish. On the other hand, *DeepFish* addresses 4 tasks, which are classification, counting, localization, and segmentation, which means algorithms that score well on this benchmark should be able to provide a comprehensive analysis for the fish community. Overall, the *DeepFish* dataset exceeds previous fish datasets in terms of size, annotation richness, and scene complexity and variability.

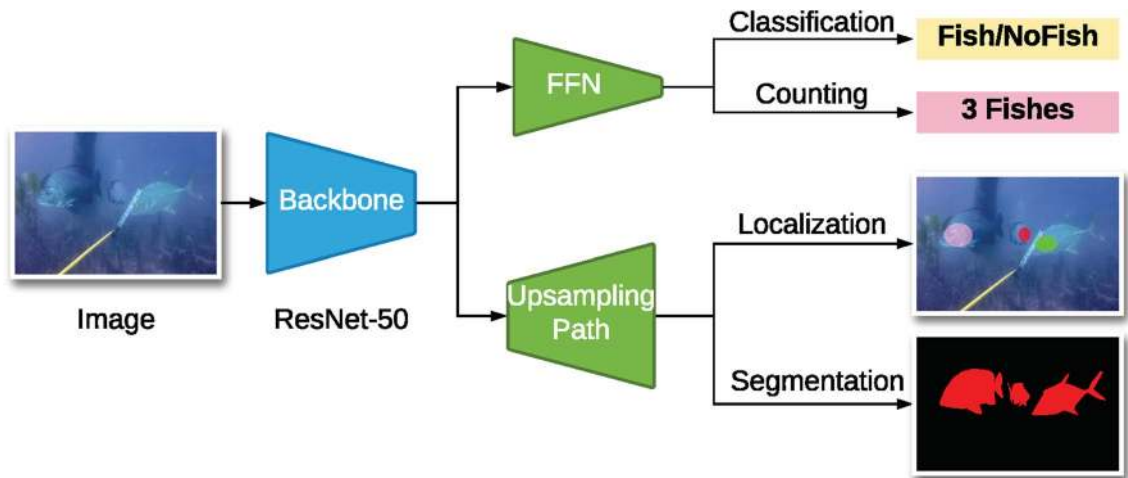
## Methods and experiments

Based on the labels of *DeepFish*, we consider these four computer vision tasks: classification, counting, localization, and segmentation. Deep learning have consistently achieved state-of-the-art results on these tasks as they can leverage the enormous size of the datasets they are trained on. These datasets include ImageNet<sup>6</sup>, Pascal<sup>7</sup>, CityScapes<sup>5</sup> and COCO<sup>24</sup>. *DeepFish* aims to be part of these large scale datasets with the unique goal of understanding complex fish habitats for the purpose of inspiring further research in this area.

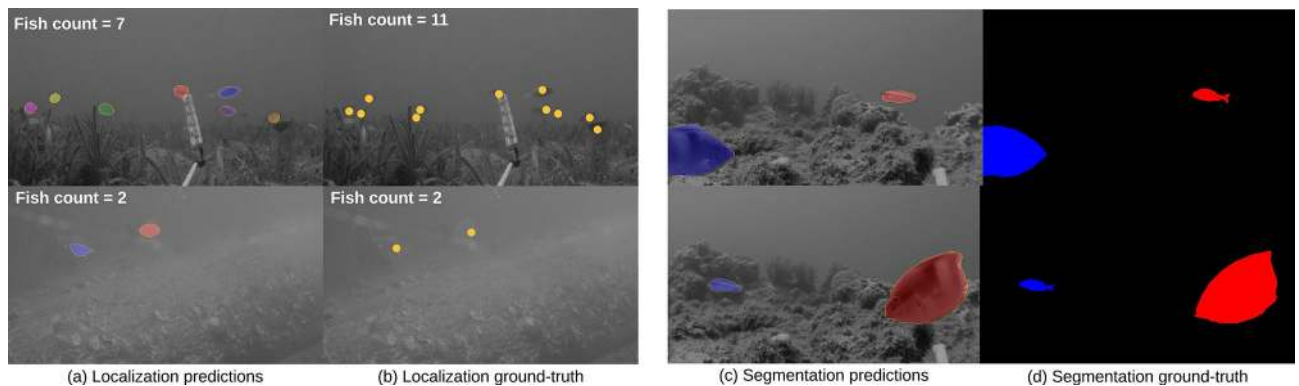
We present standard deep learning methods for each of these tasks. Shown as the blue module in Figure 4, these methods have the ResNet-50<sup>13</sup> backbone which is one of the most popular feature extractors for image

	Classification	Counting	Localization		Segmentation
	Accuracy	MAE	MAE	GAME	mIoU
Random weights	0.65	1.30	1.22	1.30	0.49
Pretrained weights	0.99	0.38	0.21	1.22	0.93

**Table 3.** Comparison between randomly initialized and ImageNet pretrained models. Classification results were evaluated on the *FishClf* dataset, counting and localization on the *FishLoc* dataset, and segmentation on the *FishSeg* dataset.



**Figure 4.** Deep learning methods. The architecture used for the four computer vision tasks of classification, counting, localization, and segmentation consists of two components. The first component is the ResNet-50 backbone which is used to extract features from the input image. The second component is either a feed-forward network that outputs a scalar value for the input image or an upsampling path that outputs a value for each pixel in the image.



**Figure 5.** Qualitative results on counting, localization, and segmentation. (a) Prediction results of the model trained with the LCFCN loss<sup>21</sup>. (b) Annotations that represent the  $(x, y)$  coordinates of each fish within the images. (c) Prediction results of the model trained with the focal loss<sup>25</sup>. (d) Annotations that represent the full segmentation masks of the corresponding fish.

understanding and visual recognition. They enable models to learn from large datasets and transfer the acquired knowledge to train efficiently on another dataset. This process is known as transfer learning and has been consistently used in most current deep learning methods<sup>22</sup>. Such pretrained models can even recognize object classes that they have never been trained on<sup>29</sup>. This property illustrates how powerful the extracted features are from a pretrained ResNet-50.

Therefore, we initialize the weights of our ResNet-50 backbones by pre-training it on ImageNet following the procedure discussed in<sup>6</sup>. ImageNet consists of over 14 million images categorized over 1,000 classes. As a result, the backbone learns to extract strong, general features for unseen images by training on such dataset.

These features are then used by a designated module to perform their respective computer vision task such as classification and segmentation. We describe these modules in the sections below.

To put the results into perspective, we also include baseline results by training the same methods without ImageNet pretraining (Table 3). In this case, we randomly initialize the weights of the ResNet-50 backbone with Xavier's method<sup>11</sup>. These results also illustrate the efficacy of having pretrained models over randomly initialized models.

**Classification results.** The goal of the classification task is to identify whether images are foreground (contains fish) or background (contains no fish). We use accuracy to evaluate the models on this task which is a standard metric for binary classification problems<sup>3,8,9,15,27</sup>. The metric is computed as

$$ACC = (TP + TN)/N,$$

where  $TP$  and  $TN$  are the true positives and true negatives, respectively, and  $N$  is the total number of images. A true positive represents an image with at least one fish that is predicted as foreground, whereas a true negative represents an image with no fish that is predicted as background. For this task we used the FishClf dataset for this task where the number of images labeled is 39,766.

The classification architecture consists of a ResNet-50 backbone and a feed-forward network (FFN) (classification branch of Fig. 4). FFN takes as input features extracted by ResNet-50 and outputs a probability for the image corresponding to how likely it contains a fish. If the probability is higher than 0.5 the predicted classification label is foreground. For the FFN, we use the network presented in ImageNet which consists of 3 layers. However, instead of the original 1,000-class output layer, we use a 2-class output layer to represent the foreground or background class.

During training, the classifier learns to minimize the binary cross-entropy objective function<sup>28</sup> using the Adam<sup>16</sup> optimizer. The learning rate was set as  $10^{-3}$  and the batch size was set to be 16. Since FFN require a fixed resolution of the extracted features, the input images are resized to  $224 \times 224$ . At test time, the model outputs a score for each of the two classes for a given unseen image. The predicted class for that image is the class with the higher score.

In Table 3 we compare between a classifier with the backbone pretrained on ImageNet and with the randomly initialized backbone. Note that both classifiers have their FFN network initialized at random. We see that the pretrained model achieved near-perfect classification results outperforming the baseline significantly. This result suggests that transfer learning is important and that deep learning has strong potential for analyzing fish habitats.

**Counting results.** The goal of the counting task is to predict the number of fish present in an image. We evaluate the models on the FishLoc dataset, which consists of 3,200 images labeled with point-level annotations. We measure the model's efficacy in predicting the fish count by using the mean absolute error. It is defined as,

$$MAE = \frac{1}{N} \sum_{i=1}^N |\hat{C}_i - C_i|,$$

where  $C_i$  is the true fish count for image  $i$  and  $\hat{C}_i$  is the model's predicted fish count for image  $i$ . This metric is standard for object counting<sup>12,23</sup> and it measures the number of miscounts the model is making on average across the test images.

The counting branch in Fig. 4 shows the architecture used for the counting task, which, similar to the classifier, consists of a ResNet-50 backbone and a feed-forward network (FFN). Given the extracted features from the backbone for an input image, the FFN outputs a number that correspond to the count of the fish in the image. Thus, instead of a 2-class output layer like with the classifier, the counting model has a single node output layer.

We train the models by minimizing the squared error loss<sup>28</sup>, which is a common objective function for the counting task. At test time, the predicted value for an image is the predicted object count.

The counting model with the backbone pretrained on ImageNet achieved an MAE of 0.38 (Table 3). This result corresponds to making an average of 0.38 fish miscounts per image which is satisfactory as the average number of fish per image is 7. In comparison, the counting model initialized randomly achieved an MAE of 1.30. This result further confirms that transfer learning and deep learning can successfully address the counting task despite the fact that the dataset for counting (FishLoc) is much smaller than classification (FishClf).

**Localization results.** Localization considers the task of identifying the locations of the fish in the image. It is a more difficult task than classification and counting as the fish can extensively overlap. Like with the counting task, we evaluate the models on the FishLoc dataset. However, MAE scores do not provide how well the model performs at localization as the model can count the wrong objects and still achieve perfect score. To address this limitation, we use a more accurate evaluation for localization by following<sup>12</sup>, which considers both the object count and the location estimated for the objects. This metric is called Grid Average Mean absolute Error (GAME). It is computed as

$$GAME = \sum_{i=1}^4 GAME(L), \quad GAME(L) = \frac{1}{N} \sum_{i=1}^N \left( \sum_{l=1}^{4^L} |D_i^l - \hat{D}_i^l| \right),$$

where  $D_i^l$  is the number of point-level annotations in region  $l$ , and  $\hat{D}_i^l$  is the model's predicted count for region  $l$ .  $GAME(L)$  first divides the image into a grid of  $4^L$  non-overlapping regions, and then computes the sum of

the MAE scores across these regions. The higher  $L$ , the more restrictive the GAME metric will be. Note that  $\text{GAME}(0)$  is equivalent to MAE.

The localization branch in Fig. 4 shows the architecture used for the localization task, which consists of a ResNet-50 backbone and an upsampling path. The upsampling path is based on the network described in FCN8<sup>26</sup> which is a standard fully convolutional neural network meant for localization and segmentation, which consists of three upsampling layers.

FCN8 processes images as follows. The features extracted with the backbone are of a smaller resolution than the input image. These features are then upsampled with the upsampling path to match the resolution of the input image. The final output is a per-pixel probability map where each pixel represents the likelihood that it belongs to the *fish* class.

The model is trained using a state-of-the-art localization-based loss function called LCFCN<sup>21</sup>. LCFCN is trained using four objective functions: image-level loss, point-level loss, split-level loss, and false positive loss. The image-level loss encourages the model to predict all pixels as background for background images. The point-level loss encourages the model to predict the centroids of the fish. Unfortunately, these two loss terms alone do not prevent the model from predicting every pixel as fish for foreground images. Thus, LCFCN also minimizes the split loss and false-positive loss. The split loss splits the predicted regions so that no region has more than one point annotation. This results in one blob per point annotation. The false-positive loss prevents the model from predicting blobs for regions where there are no point annotations. Note that training LCFCN only requires point-level annotations which are spatial locations of where the objects are in the image.

At test time, the predicted probability map are thresholded to become 1 if they are larger than 0.5 and 0 otherwise. This results in a binary mask, where each blob is a single connected component and they can be collectively obtained using the standard connected components algorithm. The number of connected components is the object count and each blob represents the location of an object instance (see Fig. 5 for example predictions with FCN8 trained with LCFCN).

Models trained on this dataset are optimized using Adam<sup>16</sup> with a learning rate of  $10^{-3}$  and weight decay of 0.0005, and have been ran for 1,000 epochs on the training set. In all cases the batch size is 1, which makes it applicable for machines with limited memory.

Table 3 shows the MAE and GAME results of training an FCN8 with and without a pretrained ResNet-50 backbone using the LCFCN loss function. We see that pretraining leads to significant improvement on MAE and a slight improvement for GAME. The efficacy of the pretrained model is further confirmed by the qualitative results shown in Fig. 5a where the predicted blobs are well-placed on top of the fish in the images.

**Segmentation results.** The task of segmentation is to label every pixel in the image as either fish or not fish (Fig. 5c,d). When combined with depth information, a segmented image allows us to measure the size and the weight of the fish in a location, which can vastly improve our understanding of fish communities. We evaluate the model on the FishSeg dataset for which we acquired per-pixel labels for 620 images. We evaluate the models on this dataset using the standard Jaccard index<sup>5,7</sup> which is defined as the number of correctly labelled pixels of a class, divided by the number of pixels labelled with that class in either the ground truth mask or the predicted mask. It is commonly known as the intersection-over-union metric IoU, computed as  $\frac{TP}{TP+FP+FN}$ , where TP, FP, and FN are the numbers of true positive, false positive, and false negative pixels, respectively, which is determined over the whole test set. In segmentation tasks, the IoU is preferred over accuracy as it is not as affected by the class imbalances that are inherent in foreground and background segmentation masks like in *DeepFish*.

During training, instead of minimizing the standard per-pixel cross-entropy loss<sup>26</sup>, we use the focal loss function<sup>25</sup> which is more suitable when the number of background pixels is much higher than the foreground pixels like in our dataset. The rest of the training procedure is the same as with the methods trained for localization.

At test time, the model outputs a probability for each pixel in the image. If the probability is higher than 0.5 for the foreground class, then the pixel is labeled as fish, resulting in a segmentation mask for the input image.

The results in Table 3 show a comparison between the pretrained and randomly initialized segmentation model. Like with the other tasks, the pretrained model achieves superior results both quantitatively and qualitatively (Fig. 5).

**Ethical approval.** This work was conducted with the approval of the JCU Animal Ethics Committee (protocol A2258), and conducted in accordance with DAFF general fisheries permit #168652 and GBRMP permit #CMES63.

## Conclusions and perspectives

We have introduced *DeepFish* as a benchmark suite consisting of a large-scale dataset for the purpose of developing new models that can efficiently analyze remote underwater fish habitats. Compared to current fish datasets, *DeepFish* consists of a diverse set of images that capture complex scenes from a large set of fish habitats that span coastal marine-environments of tropical Australia. We acquired point-level and per-pixel annotations and designed experimental setups that enable models to be evaluated for the tasks of classification, counting, localization and segmentation. We also present results demonstrating the efficacy of standard deep learning methods that were pretrained on ImageNet. These results can be used as baseline to help evaluate new models for this problem setup.

For future work, we plan to adapt *DeepFish* by adding new benchmarks and annotations in order to inspire fish analysis models for other useful use cases. Thus, we will consider challenges that fall under weak supervision, active learning, or few-shot learning where the goal is to train on datasets whose labels were collected with minimal human effort.



## Data availability

The *DeepFish* dataset and the code is publicly available at <https://alzayats.github.io/DeepFish/> and <https://github.com/alzayats/DeepFish>, respectively.

Received: 18 November 2019; Accepted: 14 August 2020

Published online: 04 September 2020

## References

- Anantharajah, K., Ge, Z., McCool, C., Denman, S., Fookes, C., Corke, P. I., Tjondronegoro, D. & Sridharan, S. Local inter-session variability modelling for object classification. In *IEEE Winter Conference on Applications of Computer Vision*, 309–316 (2014).
- Barnes, L. M., Bellwood, D. R., Sheaves, M. & Tanner, J. K. The use of clear-water non-estuarine mangroves by reef fishes on the great barrier reef. *Mar. Biol.* **159**, 211–220 (2012).
- Boom, B. *et al.* A research tool for long-term and continuous analysis of fish assemblage in coral-reefs using underwater camera footage. *Ecol. Inform.* **23**, 83–97 (2014).
- Bradley, M., Baker, R. W., Nagelkerken, I. & Sheaves, M. Context is more important than habitat type in determining use by juvenile fish. *Landsc. Ecol.* **34**, 427–442 (2019).
- Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S. & Schiele, B. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3213–3223 (2016).
- Deng, J., Dong, W., Socher, R., Li, L.J., Li, K. & Fei-Fei, L. Imagenet: a large-scale hierarchical image database. In *Conference on Computer Vision and Pattern Recognition (CVPR)* (2009).
- Everingham, M. *et al.* The pascal visual object classes challenge: a retrospective. *Int. J. Comput. Vis.* **111**(1), 98–136 (2015).
- Fisher, R. B. *et al.* Fish4knowledge deliverable d7.6, final report to ec-technical. <https://bit.ly/2Ex7dnZ> (2013).
- French, G., Fisher, M., & Mackiewicz, M. Needle, C. Convolutional neural networks for counting fish in fisheries surveillance video. In *Machine Vision of Animals and their Behaviour (MVAB)* (2015).
- Garcia, R. *et al.* Automatic segmentation of fish using deep learning with application to fish size measurement. *ICES J. Mar. Sci.* <https://doi.org/10.1093/icesjms/fsz186> (2019).
- Glorot, X. & Bengio, Y. Understanding the difficulty of training deep feedforward neural networks. In *AISTATS* (2010).
- Guerrero-Gomez-Olmedo, R., Torre-Jimenez, B., Lopez-Sastre, R., Maldonado-Bascon, S. & Onoro-Rubio, D. Extremely overlapping vehicle counting. In *Iberian Conference on Pattern Recognition and Image Analysis (IbPRIA)* (2015).
- He, K., Zhang, X., & Ren, S. Sun, J. Deep residual learning for image recognition. In *Conference on Computer Vision and Pattern Recognition (CVPR)* (2016).
- Hussain, M. A., Saputra, T., Szabo, E. A. & Nelan, B. An overview of seafood supply, food safety and regulation in New South Wales, Australia. *Foods* **6**(7), 52. <https://doi.org/10.3390/foods6070052> (2017).
- Joly, A., Goeau, H., Glotin, H., Spampinato, C., Bonnet, P., Vellinga, W.-P., Planque, R., Rauber, A., Fisher, R. B. & Muller, H. LifeCLEF 2014: multimedia life species identification challenges. In *CLEF* (2014).
- Kingma, D. P. & Ba, J. Adam: a method for stochastic optimization. In *CoRR*. [arXiv:abs/1412.6980](https://arxiv.org/abs/1412.6980) (2014).
- Klaser, A. Image annotation tool with image masks. [https://lear.inrialpes.fr/people/klaeser/software\\_image\\_annotation](https://lear.inrialpes.fr/people/klaeser/software_image_annotation) (2010).
- Konovalov, D. A., Saleh, A., Domingos, J. A., White, R. D. & Jerry, D. R. Estimating mass of harvested Asian seabass *Lates calcarifer* from images. *World J. Eng. Technol.* **6**(03), 15 (2018).
- Konovalov, D. A., Saleh, A., Bradley, M., Sankupellay, M., Marini, S. & Sheaves, M. Underwater fish detection with weak multi-domain supervision. In *International Joint Conference on Neural Networks (IJCNN)*, 1–8 (2019).
- Konovalov, D. A., Saleh, A., Efremova, D. B., Domingos, J. A. & Jerry, D. R. Automatic weight estimation of harvested fish from images. In *2019 Digital Image Computing: Techniques and Applications (DICTA)*, 1–7 (2019).
- Laradji, I. H., Rostamzadeh, N., Pinheiro, P. H. O., Vazquez, D. & Schmidt, M. W. Where are the blobs: counting by localization with point supervision. In *European Conference on Computer Vision (ECCV)* (2018).
- Laradji, I. H., Pardinias, R., & Rodriguez, P. Vazquez, D. Looc: localize overlapping objects with count supervision. [arXiv:2007.01837](https://arxiv.org/abs/2007.01837) (2020).
- Lempitsky, V. & Zisserman, A. Learning to count objects in images. In *Advances in Neural Information Processing Systems*, 1324–1332 (2010).
- Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P. & Zitnick, C. L. Microsoft coco: common objects in context. In *European Conference on Computer Vision (ECCV)* (2014).
- Lin, T.-Y., Goyal, P., Girshick, R., He, K. & Dollár, P. Focal loss for dense object detection. In *International Conference on Computer Vision (ICCV)* (2017).
- Long, J., Shelhamer, E., & Darrell, T. Fully convolutional networks for semantic segmentation. In *Computer Vision and Pattern Recognition (CVPR)* (2015).
- Mounsaveng, S., Laradji, I., Ayed, I. B., Vazquez, D. & Pedersoli, M. Learning data augmentation with online bilevel optimization for image classification. [arXiv:2006.14699](https://arxiv.org/abs/2006.14699) (2020).
- Murphy, K. P. Machine learning—a probabilistic perspective. In *Adaptive Computation and Machine Learning Series* (2012).
- Rodriguez, P., Laradji, I., & Drouin, A. Lacoste, A. Embedding propagation: smoother manifold for few-shot classification. [arXiv:2003.04151](https://arxiv.org/abs/2003.04151) (2020).
- Rozas, L. P. & Minello, T. J. Estimating densities of small fishes and decapod crustaceans in shallow estuarine habitats: a review of sampling design with focus on gear selection. *Estuaries* **20**, 199–213 (1997).
- Russell, B. C., Torralba, A., Murphy, K. P. & Freeman, W. T. Labelme: a database and web-based tool for image annotation. *Int. J. Comput. Vis.* **77**, 157–173 (2005).
- Sheaves, M. Consequences of ecological connectivity: the coastal ecosystem mosaic. *Mar. Ecol. Prog. Ser.* **391**, 107–115 (2009).
- Sheaves, M., Johnston, R., Connolly, R. M. & Baker, R. Importance of estuarine mangroves to juvenile banana prawns. *Estuar. Coast. Shelf Sci.* **114**, 208–219. <https://doi.org/10.1016/j.ecss.2012.09.018> (2012).
- Sheaves, M., Johnston, R. & Baker, R. Use of mangroves by fish: new insights from in-forest videos. *Mar. Ecol. Prog. Ser.* **549**, 167–182 (2016).
- Stierhoff, K. & Cutter, G. Rockfish training and validation image dataset. In *NOAA Southwest Fisheries Science Center Remotely Operated Vehicle (ROV) Digital Still Images* (2013).

## Acknowledgements

This research is supported by an Australian Research Training Program (RTP) Scholarship. We gratefully acknowledge Strategic Research Initiative Funding (SRIF-2018) of James Cook University. Issam Laradji is funded by the UBC Four-Year Doctoral Fellowships (4YF).

### Author contributions

A. S. and I. H. L. contributed equally to writing the manuscript, coding and annotating the dataset. D. V. reviewed, revised the manuscript and conceived the project. D. A. K. revised the manuscript. M. B. and M. S. collected the dataset and reviewed the manuscript.

### Competing interests

The authors declare no competing interests.

### Additional information

**Correspondence** and requests for materials should be addressed to A.S.

**Reprints and permissions information** is available at [www.nature.com/reprints](http://www.nature.com/reprints).

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

© The Author(s) 2020