scientific reports

Check for updates

OPEN

# Weakly supervised underwater fish segmentation using affinity LCFCN

Issam H. Laradji[1,3]✉, Alzayat Saleh[2], Pau Rodriguez[1], Derek Nowrouzezahrai[3], Mostafa Rahimi Azghadi[2] & David Vazquez[1]

**Estimating fish body measurements like length, width, and mass has received considerable research due to its potential in boosting productivity in marine and aquaculture applications. Some methods are based on manual collection of these measurements using tools like a ruler which is time consuming and labour intensive. Others rely on fully-supervised segmentation models to automatically acquire these measurements but require collecting per-pixel labels which are also time consuming. It can take up to 2 minutes per fish to acquire accurate segmentation labels. To address this problem, we propose a segmentation model that can efficiently train on images labeled with point-level supervision, where each fish is annotated with a single click. This labeling scheme takes an average of only 1 second per fish. Our model uses a fully convolutional neural network with one branch that outputs per-pixel scores and another that outputs an affinity matrix. These two outputs are aggregated using a random walk to get the final, refined per-pixel output. The whole model is trained end-to-end using the localization-based counting fully convolutional neural network (LCFCN) loss and thus we call our method Affinity-LCFCN (A-LCFCN). We conduct experiments on the DeepFish dataset, which contains several fish habitats from north-eastern Australia. The results show that A-LCFCN outperforms a fully-supervised segmentation model when the annotation budget is fixed. They also show that A-LCFCN achieves better segmentation results than LCFCN and a standard baseline.**

Fish habitat monitoring is an important step for sustainable fisheries, as we acquire important fish measurements such as size, shape and weight. These measurements can be used to judge the growth of the fish and act as reference for feeding, fishing and conservation[1]. Thus, it helps us identify which areas require preservation in order to maintain healthy fish stocks.

The UN Food and Agriculture Organization found that 33 percent of commercially important marine fish stocks worldwide are over-fished[2]. This finding is attributed to the fact that fishing equipments often catch unwanted fish that are not of the right size[3]. Catching unwanted fish can lead to more time needed to sort them. It can also lead to more fuel consumption as these fish are extra weight on the boat, and cause long-term negative impact on the fisheries[4]. Thus, acquiring fish size information has many important applications.

Many methods for measuring fish size are based on manual labor. Some experienced fishers are able to estimate length by eye. Other fishers use a ruler to measure the length[5]. More recently, fishermen use echosounders to get the fish size but these tools are still on trail[6,7]. Unfortunately these methods are time consuming, labour intensive and can cause significant stress to the fish[8,9]. Garcia et al.[4] proposed an "underwater studio" with stereo cameras and illumination that is incorporated in trawls for automatic fish segmentation. However, their setup causes disruption to the fish which could reduce the reliability of the results.

Therefore, image segmentation systems for fish analysis[10–12] have gained lots of traction within the research community due to their potential efficiency. They can be used to segment fishes in an image in order to acquire morphological measurements such as size and shape. These systems can be installed in a trawl or underwater to cluster fish based on their sizes[4]. Promising methods for image segmentation are based on deep learning, such as fully Convolutional Neural Networks (CNN) which now dominate many computer vision related fields. FCN8[13] and ResNet38D[14] have shown to achieve promising performance in several segmentation tasks. In this work, we use a segmentation network based on FCN8 with an ImageNet[15] pretrained VGG16[16] backbone.

Most segmentation algorithms are fully supervised[13,17,18], as they require per-pixel annotations in order to train. These annotations are prohibitively expensive to gather due to the requirement of field expert annotators, a specialized tool, and intensive labor. In order to reduce these annotation costs, weakly supervised methods were proposed to leverage annotations that are cheaper to acquire. The most common labeling scheme is image-level annotation[19,20], which only requires a global label per image. Other forms of weak supervision are scribbles[21]

[1]Element AI, Montreal, Canada. [2]James Cook University, Brisbane, Australia. [3]McGill University, Montreal, Canada. ✉email: issam.laradji@gmail.com

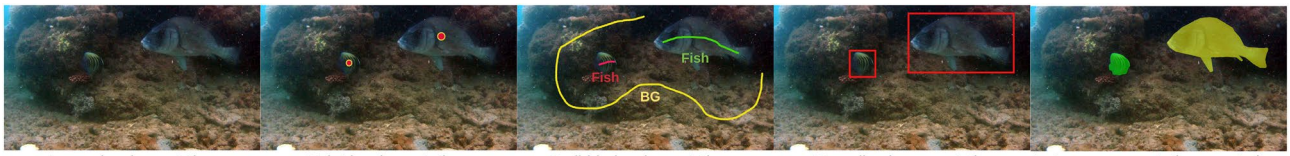



**Figure 1.** Different labeling schemes. Point-level supervision places a single point on each fish body, whereas other non-precise labelling methods such as scribble-level and bounding box annotations provide more labelling details. The full supervision labelling method on the far right provides full label masks.

**Figure 2.** DeepFish dataset. Images from different habitats with point annotations on the fish (shown as red dots). These images are from the open-source DeepFish dataset available at https://alzayats.github.io/DeepFish/.

and bounding boxes[22] which were shown to improve the ratio of labeling effort to segmentation performance. In this work, we use point-level annotations since they require a similar acquisition time as image-level annotations, while significantly boosting the segmentation performance[23]. Unfortunately, methods that use point-level supervision either need training a proposal network[24] or tend to output large blobs that do not conform to the segmentation boundaries[23]. Thus, these methods are not well suited to images with objects of specific boundaries like fish. A promising weakly supervised method is localization-based counting fully convolutional neural network[25] (LCFCN), which is better at localizing multiple objects but does not segment the objects correctly. In this work we build on LCFCN to improve its segmentation capabilities (Fig. 1).

Ahn and Kwak[26] showed that it is possible to train a segmentation network with image-level annotations by learning to predict a pixel-wise affinity matrix. This matrix is a weighted graph where each edge represents the similarity between each pair of pixels[27,28]. However, in Ahn and Kwak[26] the process to obtain this affinity matrix is costly and depends heavily on proxy methods such as Class Activation Map (CAM)[26] to approximate it. Given the advantages of affinity networks for image segmentation as shown in Ahn and Kwak[26] and Tang et al.[29], we propose a novel affinity module that automatically infers affinity weights. This module can be integrated on any standard segmentation network and it eliminates the need for explicit supervision such as acquiring pairs between pixels of CRF-refined CAMs[26].

Therefore, we extend LCFCN with an affinity-based module in order to improve the output segmentation of the fish boundaries. Our model follows three main steps. First, features are extracted using a pre-trained backbone like ResNet38. Then, an activation branch uses these features to produce pixel-wise class scores. From the same backbone features, the affinity branch infers pairwise affinity scores between the pixels. Finally, the affinity matrix is combined with the pixel-wise class scores using random walk[30] to produce a segmentation mask. The random walk encourages neighboring pixels to have similar probabilities based on their semantic similarities. As a result, the predicted segmentations are encouraged to take the shape of the fish. During training, these segmentations are compared against the point-level annotations using the LCFCN loss[25]. This loss ensures that only one blob is output per object which is important when there are multiple fish in an image. Unlike AffinityNet[26] which requires expensive pre-processing and stage-wise learning, the whole model can be trained end-to-end efficiently. Finally, the segmentation output by our model can be used to generate pseudo ground-truth labels for the training images. Thus, we can train a fully supervised network on these pseudo ground-truth masks achieving better results. The reason behind the improvement can be attributed to the fact that these networks can be robust against noisy labels[31].

We benchmark A-LCFCN on the segmentation subset of the *DeepFish*[32] dataset. This dataset contains images from several habitats from north-eastern Australia (see Fig. 2 for examples). These habitats represent nearly the entire range of coastal and nearshore benthic habitats frequently accessible to fish species in that area. Each image in the dataset has a corresponding segmentation label, where pixels are labelled to differentiate between fish pixels and background pixels (see Fig. 4). Our method achieved an mIoU of 0.879 on *DeepFish*[32], which is significantly higher than standard point-level supervision methods, and fully-supervised methods when the annotation budget is fixed. That is, when the total dataset annotation time is capped at a certain amount of seconds. We have also evaluated our method on the SUIM dataset[33] and observed consistent results, indicating that our method can also be applied in controlled environments like those that have stereo cameras and conveyor belts.

For our contributions, (1) we propose a framework that can leverage point-level annotations and perform accurate segmentation of fish present in the wild. (2) We propose an affinity module that can be easily added to any segmentation method to make the predictions more aware of the segmentation boundaries. (3) We present results that demonstrate that our methods achieve significant improvement in segmentation over baselines and fully supervised methods when the annotation budget is fixed.

## Related work

In this section, we first review methods applied to general semantic segmentation, followed by semantic segmentation for fish analysis. Then we discuss affinity methods that use pair-wise relationships between the pixels for improved segmentation. Finally, we discuss weakly supervised methods for segmentation and object localization.

*Semantic segmentation* is an important computer vision task that can be applied to many real-life applications[13,17,18]. This task consists of classifying every object pixel into corresponding categories. Most methods are based on fully convolutional networks which can take an image of arbitrary size and produce a segmentation map of the same size. Methods based on Deeplab[17] consistently achieve state-of-the-art results as they take advantage of dilated convolutions, skip connections, and Atrous Spatial Pyramid Pooling (ASPP) for capturing objects and image context at multiple scales. However, these methods require per-pixel labels in order to train, which can result in expensive human annotation cost when acquiring a training set for a semantic segmentation task.

*Semantic segmentation methods for fish analysis* have been used for efficient, automatic extraction of fish body measurements[34], and prediction of their body weight[34–36] and shape for the purposes of preserving marine life. Garcia et al.[4] used fully-supervised segmentation methods and the Mask R-CNN[37] architecture to localize and segment each individual fish in underwater images to obtain an estimate of the boundary of every fish in the image for estimating fish sizes to prevent catches of undersized fish. French et al.[38] presented a fully-supervised computer vision system for segmenting the scenes and counting the fish from CCTV videos installed on fishing trawlers to monitor abandoned fish catch. While we also address the task of segmentation for fish analysis, to the best of our knowledge, we are the first to consider the problem setup of using point-level supervision, which can considerably lower the annotation cost.

*Affinity-based methods for semantic segmentation* have been proposed to leverage the inherent structure of images to improve segmentation outputs[39–41]. They consider the relationship between pixels which naturally have strong correlations. Many segmentation methods use conditional random fields (CRF)[17,39] to post-process the final output results. The idea is to encourage pixels that have strong spatial and feature relationships to have the same label. CRF were also incorporated to a neural network as a differentiable module to train jointly with the segmentation task[40]. Others leverage image cues based on grouping affinity and contour to model the image structure[42,43]. Most related to our work is Ahn and Kwak[26] which proposes an affinity network that learns from pairwise samples of pixels labeled with a segmentation network and a CRF. The network is then used to output an affinity matrix which is used to refine the final segmentation output. Unfortunately, these methods require expensive iterative inference procedures, and require to learn the segmentation task in stages. In addition, it does not use point-level annotations for segmentation and it is used for images with clearly salient objects in the image like in PASCAL. This is incompatible with DeepFish where there could be many objects in a single image. In our work, we use part of the affinity network as a module that can be incorporated to any segmentation network, adding minimal computational overhead while increasing the model's sensitivity to object boundaries and segmentation accuracy.

*Weakly supervised semantic segmentation* methods have risen in popularity due to their potential in decreasing the human cost in acquiring a training set. Bearman et al.[23] is one of the first methods that use point-supervision to perform semantic segmentation. They showed that manually collecting image-level and point-level labels for the PASCAL VOC dataset[44] takes only 20.0 and 22.1 seconds per image, respectively. This scheme is an order of magnitude faster than acquiring full segmentation labels, which is 239.0 seconds. The most common weak supervision setup is using image-level labels to perform segmentation[19,20]. They use a wide range of techniques that include affinity learning, self-supervision, and co-segmentation. However, these methods were applied to the PASCAL VOC[44] dataset that often has large objects. Other lines of weakly supervised methods address the problem of object localization and segment annotation[45,46]. In our work we consider underwater fish segmentation with point-level supervision which has its own unique challenges. For instance, compared to datasets like PASCAL and COCO, the DeepFish dataset has images of fish that are highly occluded. Most fishes are indistinguishable from elements in the background like debris and rocks and their shapes and sizes are difficult to capture by the model as the contrast between the body of the fish and the region surrounding it is small as observed in Fig. 2.

*Weakly supervised object localization* methods can be an important step for segmentation as they allow us to identify the locations of the objects before grouping the pixels for segmentation. Redmon and Farhadi[48] and Ren et al.[47] are current state-of-the-art methods for object localization, but they require bounding boxes. However, several methods exist that use weaker supervision to identify object locations[31,49–55]. Close to our work is LCFCN[25] which uses point-level annotations in order to obtain the locations and counts of the objects of interest. While this method produces accurate counts and identifies a partial mask for each instance, it does not produce accurate segmentation of the instances. Thus, we extend this method by using an affinity-based module that takes pairwise pixel relationships into context in order to output blobs that are more sensitive to the object boundaries.

## Methodology

We propose A-LCFCN, which extends a fully convolutional neural network with an affinity-based module that is trained using the LCFCN loss. We consider the following problem setup. We are given $X$ as a set of $n$ training images with their corresponding set of ground-truth labels $Y$. $Y_i$ is a binary matrix of the same height $H$ and width $W$ as $X$ with non-zero entries that indicate the locations of the object instances. As shown in Fig. 1, there is a single non-zero entry per fish which is represented as a dot on top of the fish.

Shown in Fig. 3, our model consists of a backbone $F_\theta^{bb}()$, an activation branch $F_\theta^{act}()$ and an affinity branch $F_\theta^{aff}()$. The backbone is a fully-convolutional neural network that takes as input an image of size $W \times H$ and extracts a downsampled feature map $f$ for the image. The activation branch takes the feature map as input and applies a set of convolutional and upsampling layers to obtain a per-pixel output $f^{act}$ as a heatmap that represents

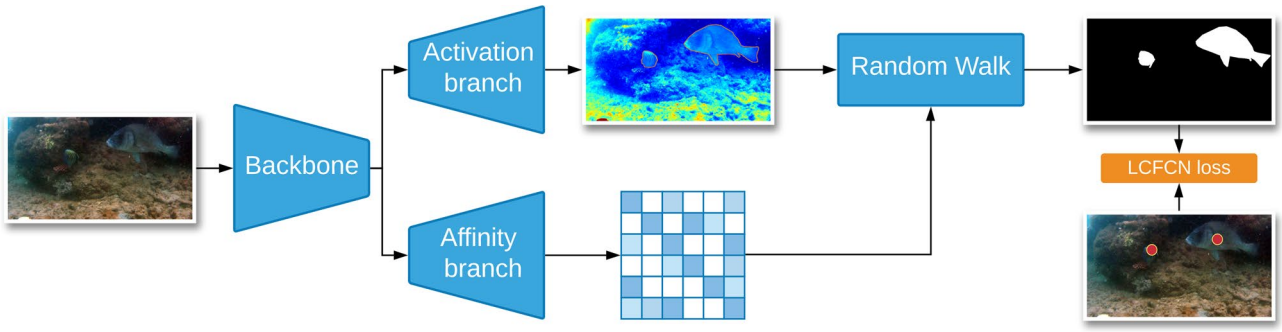



**Figure 3.** Affinity-based architecture. The first component is the ResNet-38 backbone which is used to extract features from the input image. The second component is the activation branch, which receives features from the backbone and outputs per-pixel scores with a $1 \times 1$ convolution. The third component is the affinity branch, which outputs an affinity matrix by upsampling backbone features at three different depths and merging them with a $1 \times 1$ convolution. These two outputs are aggregated using a random walk to get the final, refined per-pixel output. The images here were obtained from the open-source DeepFish dataset available at https://alzayats.github.io/DeepFish/.

the spatial likelihood of the objects of interest. The affinity branch takes the same feature map as input and outputs a class-agnostic affinity matrix $f^{aff}$ that represents the pairwise relationships between the pixels. The affinity map and the activation map are then combined using random walk to refine the per-pixel output $f^{ref}$. This refinement adapts the output to be aware of the semantic boundaries of the objects, leading to better segmentation. These components are trained collectively, end-to-end, using the LCFCN loss $\mathscr{L}_L$, which encourages each object to have a single blob. To further improve the performance, the trained model is used to output pseudo ground truth masks for the training images. These masks are then used as ground truth for training a fully-supervised network that is then validated on the test set. The details of this pipeline are laid out below.

**Obtaining the activation map and the affinity matrix.** The activation branch $F_\theta^{act}$ transforms the features $f$ obtained from the backbone to per-pixel class scores, and upsamples them to the size of the input image.

The affinity branch is based on the AffinityNet structure described in Ahn and Kwak[26], and the goal is to predict class-agnostic semantic affinity between adjacent coordinate pairs on a given image. These affinities are used to propagate the per-pixel scores from the activation branch to nearby areas of the same semantic object to improve the segmentation quality.

The affinity branch outputs a convolutional feature map $f^{aff}$ where the semantic affinity between a pair of feature vectors is defined in terms of their L1 distance as follows,

$$W_{ij} = \exp\{-||f^{\text{aff}}(x_i, y_i) - f^{\text{aff}}(x_j, y_j)||_1\}, \tag{1}$$

where $(x_i, y_i)$ indicates the coordinate of the *ith* feature on feature map $f^{aff}$.

In contrast to AffinityNet[26], we do not require affinity labels for feature pairs to train our affinity layers. These layers are directly trained using the LCFCN loss on the point-level annotations as described in "Training the weakly supervised model" section.

**Refining the activation map with affinity.** The affinity matrix is used to refine the activation map to diffuse the per-pixel scores within the object boundaries. As explained in Ahn and Kwak[26], the affinity matrix is first converted to a transition probability matrix by first applying the Hadamard power on $W$ with value $\beta$ to get $W^\beta$ and normalizing it with row-wise sum on $W^\beta$. This operation results in the following transition matrix:

$$T = D^{-1}W^\beta, \text{ where } D_{ii} = \sum_j W_{ij}^\beta. \tag{2}$$

higher $\beta$ makes the affinity propagation more conservative as it becomes more robust against small changes in the pairwise distances in the feature space. Using the random walk described in Ahn and Kwak[26] we perform matrix multiplication of $T$ on the activation map $f^{act}$ for $t$ iterations to get the refined activations $f^{ref}$.

**Training the weakly supervised model.** The goal of our training strategy is to learn to output a single blob per fish in the image using point-level annotations (Fig. 1). Thus we use the LCFCN loss described in Laradji et al.[25] as it only requires point-level supervision. While this was originally designed for counting, it is able to locate objects and segment them. On the refined activation output $f^{ref}$, we obtain per-pixel probabilities by applying the softmax operation to get $S$ which contains the likelihood that a pixel either belongs to the background or fish. The LCFCN loss $\mathscr{L}_L$ is then defined as follows:

$$\mathscr{L}_L = \underbrace{\mathscr{L}_I(S,Y)}_{\text{Image-level loss}} + \underbrace{\mathscr{L}_P(S,Y)}_{\text{Point-level loss}} + \underbrace{\mathscr{L}_S(S,Y)}_{\text{Split-level loss}} + \underbrace{\mathscr{L}_F(S,Y)}_{\text{False positive loss}}, \tag{3}$$

where $Y$ is a binary matrix with non-zero entries to indicate the point annotation ground-truth. It consists of an image-level loss ($\mathscr{L}_I$) that trains the model to predict whether there is an object in the image; a point-level loss ($\mathscr{L}_P$) that encourages the model to predict an object class for each pixel; a split-level ($\mathscr{L}_S$) and a false-positive ($\mathscr{L}_F$) loss that enforce the model to predict a single blob per object instance (see[25] for details for each of the loss components).

Applying the LCFCN loss on the original activation map usually leads to small blobs around the center of the objects which form poor segmentation masks[56]. However, with the activation map refined using the affinity matrix, the predicted blobs make better segmentation of the located objects. We call our method A-LCFCN as an LCFCN model that uses an affinity-based module.

**Training on pseudo ground-truth masks.** A trained A-LCFCN can be used to output a refined activation map for each training image. These maps are used to generate pseudo ground-truth segmentation labels for the training images. The outputs are first upsampled to the resolution of the image by bilinear interpolation. For each pixel, the class label associated with the largest activation score is selected, which could be either background or foreground. This procedure gives us segmentation labels for the training images which can be used for training a fully-supervised segmentation network, which could be any model such as DeepLabV3[57]. At test time, the trained fully-supervised segmentation network is used to get the final segmentation predictions.

**Network architecture.** While our framework can use any fully convolutional architecture, we chose a ResNet38 model based on the version defined in Ahn and Kwak[26] due to its ability to recover fine shapes of objects. However, instead of having two networks, one for the affinity output and one for the activation output, we used a shared ResNet38 as the backbone which we found to improve the results by up to 0.23 mIoU and speed up training by around 0.3 seconds per iteration using 1 NVIDIA Tesla P100.

The affinity branch consists of three layers of 1×1 convolution with 64, 128, 256 channels, respectively, to be applied on 3 levels of feature maps from the backbone. The results are bilinearly upsampled to the same size and concatenated as a single feature map. This feature map then goes through a 1×1 convolution with 448 channels to obtain affinity features.

The activation branch consists of one 1x1 convolution with 2 channels. It is applied on the last feature map of the backbone to obtain the background and the foreground activation map. These activation maps are refined using random walk with the affinity branch to get improved segmentations.

For the fully supervised segmentation model that is trained on the pseudo ground-truth masks, we use a model that consists of a backbone that extracts the image features and an upsampling path that aggregates and upscales feature maps to output a score for each pixel. The backbone is an ImageNet pretrained network such as ResNet38[26] and the upsampling layers are based on FCN8[13]. The output is a score for each pixel $i$ indicating the probability that it belongs to background or foreground. The final output is an argmax between the scores to get the final segmentation labels.

## Experiments

We evaluate our models on two splits of the DeepFish dataset[32], *FishSeg* and *FishLoc* to compare segmentation performance. We show that our method A-LCFCN outperforms the fully supervised segmentation method if the labeling effort between acquiring per-pixel labels and point annotations is fixed. Further, we show that our method outperforms other methods that do not use affinity. We further show that training on pseudo ground-truth masks generated by A-LCFCN using a fully segmentation model boosts segmentation performance even further.

**DeepFish[32].** The *DeepFish* dataset (found here: https://github.com/alzayats/DeepFish) consists of around 40 thousand images obtained from 20 different marine habitats in tropical Australia (Fig. 2). For each habitat, a fixed camera has been deployed underwater to capture a stream of images over a long period of time. The purpose is to understand fish dynamics, monitor their count, and estimate their sizes and shapes.

The dataset is divided into 3 groups: *FishClf* that contains classification labels about whether an image has fish or not, *FishLoc* that contains point-level annotatons indicating the fish location, and *FishSeg* that contains segmentation labels of the fish. Since our models require at least point-level supervision, we use *FishLoc* and *FishSeg* for our benchmarks.

*FishLoc dataset* It consists of 3200 images where each image is labeled with point-level annotations indicating the locations of the fish. It is divided into a training set (n = 1600), a validation set (n = 640), and a test set (n = 960). The point-level annotations are binary masks, in which the non-zero entries represent the (x, y) coordinates around the centroid of each fish within the images (Fig. 2).

*FishSeg dataset* It consists of 620 images with corresponding segmentation masks (see Fig. 4), separated into a training set (n = 310), validation set (n = 124), and a test set (n = 186). The images are resized into a fixed dimension 256 × 455 pixels and normalized using ImageNet statistics[15]. According to Saleh et al.[32], it takes around 2 minutes to acquire the segmentation mask of a single fish. From the segmentation masks, we acquire point-level annotations by taking the pixel with the largest distance transform of the masks as the centroid (Fig. 1). These annotations allow us to train weakly supervised segmentation models.

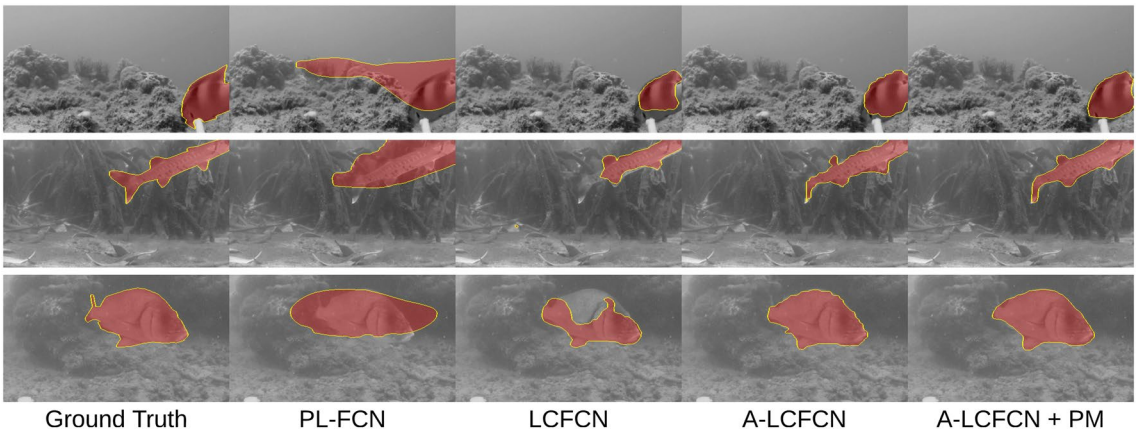



**Figure 4.** Qualitative results. Predictions obtained from training point-level FCN (PL-FCN), LCFCN, affinity LCFCN (A-LCFCN) and A-LCFCN with pseudo-masks (A-LCFCN + PM). Each row corresponds to a different sample. With the affinity branch the predictions are much closer to the ground-truth labels.

Our models were trained either on FishLoc's or FishSeg's training set. For both cases we use FishSeg's test set to evaluate the segmentation performance. We have removed training images from FishLoc that overlap with FishSeg's test set for reliable results.

**SUIM dataset[33].** The SUIM dataset consists of 1525 pixel-level annotated images for training/validation and 110 samples for testing. Annotations contain human divers, aquatic plants, wrecks/ruins, robots/instruments, reefs/invertebrates, fish and vertebrates, and sea-floor/rocks. For this work, we only use the fish labels, and we have used 20% of the training set as validation.

**Evaluation procedure.** We evaluate our models against Intersection over Union (IoU), which is a standard metric for semantic segmentation that measures the overlap between the prediction and the ground truth: $IoU = \frac{TP}{TP+FP+FN}$ where TP, FP, and FN is the number of true positive, false positive and false negative pixels across all images in the test set.

We also measure the model's efficacy in predicting the fish count using mean absolute error which 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[51,58] and it measures the number of miscounts the model is making on average across the test images.

We also measure localization performance using Grid Average Mean Absolute Error (GAME)[58] which is defined as, $\text{GAME}(L) = \frac{1}{N}\sum_{i=1}^{N}\left(\sum_{l=1}^{4^L}\left|\hat{C}_i^l - c_i^l\right|\right)$, where, $\hat{C}_i^l$ is the estimated count in a region $l$ of image $n$, and $c_i^l$ is the ground truth for the same region in the same image. The higher $L$, the more restrictive the *GAME* metric will be. We present results for $GAME(L = 4)$ which divides the image using a grid of 256 non-overlapping regions where we compute the sum of the MAE across these sub-regions.

**Methods and baselines.** We compare our method against two other weakly supervised image segmentation methods and a fully-supervised method. All these methods use the same feature extracting backbone of ResNet38, which we describe below.

*Fully supervised fully convolutional neural network (FS-FCN)* This method is based on the FCN8 architecture described by Long et al.[13]. It is trained with the true per-pixel class labels (full supervision). It combines a weighted cross-entropy loss and weighted IoU loss as defined in Eq. (3) and (5) from Wei et al.[59], respectively. It is an efficient method that can learn from ground truth segmentation masks that are imbalanced between different classes. In our case the number of pixels corresponding to fish is much lower than those to the background.

*Point-level loss (PL-FCN)* This method uses the loss function described in Bearman et al.[23] which minimizes the cross-entropy against the provided point-level annotations. It also encourages all pixel predictions to be background for background images.

*LCFCN* This method is trained using the loss function proposed by Laradji et al.[25] against point level annotations to produce a single blob per object and locate objects effectively. LCFCN is based on a semantic segmentation architecture that is similar to FCN[13]. Since it was originally designed for counting and localization, LCFCN optimizes a loss function that ensures that only a single small blob is predicted around the centre of each object. This prevents the model from predicting large blobs that merge several object instances.

*A-LCFCN (ours)* This method extends LCFCN by adding an affinity branch as described in "Methodology" section. Inspired by AffinityNet[26], this branch predicts class-agnostic semantic similarity between pairs of neighbouring coordinates. The predicted similarities are used in a random walk[30] as transition probabilities to refine the activation scores obtained from the activation branch.

| | Trained on FishLoc—tested on FishSeg | | | Trained and Tested on FishSeg | | |
|---|---|---|---|---|---|---|
| | Background | Foreground | mIoU | Background | Foreground | mIoU |
| FS-FCN | – | – | – | 0.992 | 0.663 | 0.827 |
| PL-FCN | 0.931 | 0.214 | 0.573 | 0.910 | 0.173 | 0.542 |
| LCFCN | 0.989 | 0.559 | 0.774 | 0.992 | 0.684 | 0.838 |
| A-LCFCN | 0.993 | 0.727 | 0.860 | 0.993 | 0.713 | 0.853 |
| A-LCFCN+PM | 0.994 | 0.764 | 0.879 | 0.993 | 0.730 | 0.862 |

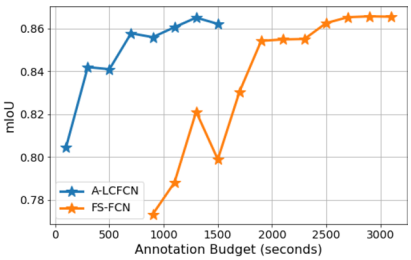
**Table 1.** Comparison between methods evaluated on the FishSeg test set, trained on either the FishLoc train set or the FishSeg train set. *Foreground* is the IoU between the predicted fish segmentation and their ground-truth, and *Background* is the IoU between the predicted background segmentation and its ground-truth. The annotation budget for FS-FCN and the methods trained on FishLoc was around 1500 seconds.

| | mIoU |
|---|---|
| LCFCN | 0.598 |
| PL-FCN | 0.683 |
| A-LCFCN | **0.749** |

**(a)** Results on the SUIM Dataset

| | MAE | GAME |
|---|---|---|
| always-median | 0.575 | - |
| LCFCN | 0.032 | 0.066 |
| A-LCFCN | 0.038 | 0.066 |
| A-LCFCN+PM | 0.031 | 0.063 |

**(b)** Counting and Localization Results.




**(c)** Annotation budget and mIoU.

**Figure 5.** Additional results on (**a**) the SUIM Dataset, (**b**) counting and localization, (**c**) Annotation budget.

*A-LCFCN + PM (ours)* This method first uses the output of a trained A-LCFCN on the training set to obtain pseudo mask labels. Then an FS-FCN is trained on these pseudo masks and is used to output the final segmentation results.

*Implementation details* Our methods use an Imagenet[15] pre-trained ResNet38[14]. The models are trained with a batch size of 1 for 1000 epochs with ADAM[60] and learning rates of $10^{-4}$, $10^{-5}$ and $10^{-6}$. We report the scores on the test set of *FishSeg* using the model with the learning rate that achieved the best validation score. We used early stopping with patience of 10 epochs. We used the default coefficients for the LCFCN loss from Laradji et al.[25], since we have not observed a difference in the final result when these coefficients are changed.

**Comparison against weak supervision.** We train the proposed method and baselines on the FishSeg and FishLoc training sets and report the results on the FishSeg test set (which is a held-out set) in Table 1. Our results include 3 statistics, the Intersection-over-Union (IoU) between the predicted foreground mask and the fish true mask, the predicted background mask and the true background mask, and their average (mIoU).

Training on the FishLoc train set, A-LCFCN obtains a significantly higher IoU than LCFCN and PL-FCN methods, we observe a similar trend on the SUIM dataset (Fig. 5a). As shown in the qualitative results (Fig. 4), we see that LCFCN produces small blobs around the center of the objects while PL-FCN outputs large blobs. For both cases, they do not consider the shape of the object as much as A-LCFCN, suggesting that the affinity branch helps in focusing on the segmentation boundaries.

Training on the FishSeg train set which contains less images than FishLoc, the margin improvement between A-LCFCN and LCFCN is smaller. Further, LCFCN performed better when trained on the FishSeg training set than with FishLoc (see Table 1). We observed that the reason behind this result is that LCFCN starts outputting smaller blobs around the object centers the more images it trains on. Thus, it learns to perform better localization at the expense of worse segmentation. On the other hand, A-LCFCN achieved improved segmentation results when trained on the larger training set FishLoc than FishSeg. This result suggests that, with enough images, the affinity branch helps the model focus on achieving better segmentation.

We also report the results of A-LCFCN + PM which shows a consistent improvement over A-LCFCN for both FishLoc and FishSeg benchmarks. This result shows that a fully supervised method can use noisy labels generated from A-LCFCN to further improve the predicted segmentation labels. In Fig. 4 we see that this procedure significantly improves the segmentation boundaries over A-LCFCN's output.

**Comparison against full supervision.** In Table 1 we report the results of our methods when fixing the annotation budget. The annotation budget was fixed at around 1500 seconds, which is the estimated time it took to annotate the FishLoc dataset. The average time of annotating a single fish and images without fish was one second[32]. For FS-FCN which was trained on segmentation annotations, the training set consisted of 161 background images and 11 foreground images as it required around 2 minutes to segment a single fish. We see that A-LCFCN + PM outperforms FS-FCN in this setup by a significant margin, which suggests that with A-LCFCN

point-level annotations are more cost-efficient in terms of labeling effort and segmentation performance. In Fig. 5c we compare FS-FCN with A-LCFCN for multiple annotation budgets. We observe that A-LCFCN outperforms supervised learning by a significant margin.

**Counting and localization results.** To further evaluate the quality of the representations learned by A-LCFCN, we also test it on the FishLoc dataset for the counting and localization tasks. These tasks are essential for marine biologists, which have to assess and track changes in large fish populations[61,62]. Thus, having a model that automates the localization of these fishes can greatly reduce the cost of tracking large populations, thus helping marine scientist to do efficient monitoring. For our models, the counts are the number of predicted blobs in the image using the connected components algorithms described in Laradji et al.[25].

As a reference, we added the MAE result of 'always-median' in Fig. 5b which is a model that outputs a count of 1 for every test image as it is the median fish count in the training set. We see that although A-LCFCN+PM has improved segmentation over A-LCFCN and LCFCN, the counting and localization counts are similar. These results suggest that we can solely use A-LCFCN+PM for the tasks of segmentation, localization and counting to have a comprehensive analysis of a fish habitat. Note that all blobs count for the MAE metric even if they do not intersect with the fish. Thus, MAE measures the counting score but not the localization score. GAME (described in "SUIM Dataset" section), on the other hand, measures localization by computing the MAE within small regions. So if one fish is in a region and the blob is in another region in the image, then the localization score is low.

**Model's limitations.** Point-level annotations are not as easy to acquire as image-level annotations. If there are plenty of fish, it would be easier to simply specify that the image has at least one fish and let the model learn to localize all fish in the image. This approach is not currently possible with our method. Another limitation is in the possible lack of generalization. It is not clear that the model can localize fish at habitats of completely different background and constraints. These limitations are opportunities for future work that could make significant contributions to this area.

## Conclusion

In this paper, we presented a novel affinity-based segmentation method that only requires point-level supervision for efficient monitoring of fisheries. Our approach, A-LCFCN, is trained end-to-end with the LCFCN loss and eliminates the need of explicit supervision for obtaining the pair-wise affinities between pixels. The proposed method combines the output of any standard segmentation architecture with the predicted affinity matrix to improve the segmentation masks with a random walk. Thus, the proposed method is agnostic to the architecture and can be used to improve the segmentation results of any standard backbone. Experimental results demonstrate that A-LCFCN produces significantly better segmentation masks than previous point-level segmentation methods. We also demonstrate that A-LCFCN gets closer to full supervision when used to generate pseudo-masks to train fully-supervised segmentation network. These results are particularly encouraging for reducing the costs of fish monitoring and achieving sustainable fisheries.

## Data availability

The code is publicly available at https://github.com/IssamLaradji/affinity_lcfcn.

Received: 18 February 2021; Accepted: 11 August 2021
Published online: 30 August 2021

## References

1. Ying, Y. *et al.* Application of machine vision technique to automatic quality identification of agricultural products (i). *Trans. Chin. Soc. Agric. Eng.* **16**(1), 103–108 (2000).
2. Delgado, C. L.. *Fish to 2020: Supply and Demand in Changing Global Markets*, Vol. 62 (WorldFish, 2003).
3. Roda, M. A. P., Gilman, E., Huntington, T., Kennelly, S. J., Suuronen, P., Chaloupka, M., & Medley, P. A.. *A Third Assessment of Global Marine Fisheries Discards* (Food and Agriculture Organization of the United Nations, 2019).
4. Garcia, R. *et al.* Automatic segmentation of fish using deep learning with application to fish size measurement. *ICES J. Mar. Sci.* **6**, 66 (2019).
5. Strachan, N. Length measurement of fish by computer vision. *Comput. Electron. Agric.* **8**(2), 93–104 (1993).
6. Pobitzer, A., Ona, E., Macaulay, G., Korneliussen, R., Totland, A., Heggelund, Y., & Eliassen, I.. Pre-catch sizing of herring and mackerel using broadband acoustics. In *ICES Symposium on Marine Ecosystem Acoustics (Some Acoustics)* (2015).
7. Berges, B., Sakinan, S., & van Helmond, E. *Practical Implementation of Real-Time Fish Classification from Acoustic Broadband Echo Sounder Data*. Technical report (Wageningen Marine Research, 2017).
8. Beddow, T. A., Ross, L. G. & Marchant, J. A. Predicting salmon biomass remotely using a digital stereo-imaging technique. *Aquaculture* **146**(3–4), 189–203 (1996).
9. Booman, A., Parin, M. & Zugarramurdi, A. Efficiency of size sorting of fish. *Int. J. Prod. Econ.* **48**(3), 259–265 (1997).
10. Yu, C. *et al.* Segmentation and measurement scheme for fish morphological features based on mask r-CNN. *Inf. Process. Agric.* **6**, 66 (2020).
11. Garcia, R. *et al.* Automatic segmentation of fish using deep learning with application to fish size measurement. *ICES JMS* **77**(4), 1354–1366 (2020).
12. Hao, M., Yu, H., & Li, D. The measurement of fish size by machine vision-a review. In *ICCCTA*, 15–32 (2015).
13. Long, J., Shelhamer, J., & Darrell, T. Fully convolutional networks for semantic segmentation. In *CVPR* (2015).
14. Wu, Z., Shen, C. & Van Den Hengel, A. Wider or deeper: Revisiting the resnet model for visual recognition. *PR* **90**, 119–133 (2019).
15. Russakovsky, O. *et al.* ImageNet large scale visual recognition challenge. *IJCV* **115**(3), 66 (2015).
16. Simonyan, K., & Zisserman, A. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014).

17. Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K. & Yuille, A. L. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE T-PAMI* **40**(4), 834–848 (2017a).
18. Jégou, S., Drozdzal, M., Vazquez, D., Romero, D., & Bengio, Y. The one hundred layers tiramisu: Fully convolutional densenets for semantic segmentation. In *CVPR* (2017).
19. Briq, R., Moeller, M., & Gall, J. Convolutional simplex projection network (CSPN) for weakly supervised semantic segmentation. In *BMVC* (2018).
20. Ahn, J., Cho, S., & Kwak, S. Weakly supervised learning of instance segmentation with inter-pixel relations. In *CVPR* (2019).
21. Vernaza, P., & Chandraker, M. Learning random walk label propagation for weakly-supervised semantic segmentation. In *CVPR* (2017).
22. Hu, R., Dollar, P., He, P., Darrell, T., & Girshick, R. Learning to segment every thing. In *CVPR* (2018).
23. Bearman, R., Russakovsky, O., Ferrari, V., & Fei-Fei, L. What's the point: Semantic segmentation with point supervision. In *ECCV* (2016).
24. Laradji, I. H., Rostamzadeh, N., Pinheiro, P. O., Vázquez, P. O., & Schmidt, M. Instance segmentation with point supervision. In *ICIP* (2019).
25. Laradji, I. H., Rostamzadeh, N., Pinheiro, P. O., Vazquez, D., & Schmidt, M. Where are the blobs: Counting by localization with point supervision. In *ECCV* (2018).
26. Ahn, J., & Kwak, S. Learning pixel-level semantic affinity with image-level supervision for weakly supervised semantic segmentation. In *CVPR* (2018).
27. Shi, J. & Malik, J. Normalized cuts and image segmentation. *T-PAMI* **22**(8), 888–905 (2000).
28. Levin, A., Lischinski, D. & Weiss, Y. A closed-form solution to natural image matting. *T-PAMI* **30**(2), 228–242 (2007).
29. Tang, M., Perazzi, F., Djelouah, A., Ben Ayed, I., Schroers, C., & Boykov, Y. On regularized losses for weakly-supervised CNN segmentation. In *ECCV* (2018).
30. Lovász, L. Random walks on graphs: A survey. *Combin. Paul Erdos Eighty* **2**(1), 1–46 (1993).
31. Laradji, I. H., Vazquez, D., & Schmidt, M. Where are the masks: Instance segmentation with image-level supervision. In *BMVC* (2019).
32. Saleh, A. *et al.* A realistic fish-habitat dataset to evaluate algorithms for underwater visual analysis. *Sci. Rep.* **10**(1), 1–10 (2020).
33. Islam, M. J., Edge, C., Xiao, Y., Luo, Y., Mehtaz, M., Morse, C., Enan, S. S., & Sattar, J. Semantic segmentation of underwater imagery: Dataset and benchmark. In *IROS* (2020).
34. Fernandes, A. F. *et al.* Deep Learning image segmentation for extraction of fish body measurements and prediction of body weight and carcass traits in Nile tilapia. *Comput. Electron. Agric.* **170**, 105–274 (2020).
35. 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).
36. 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)* (2019).
37. He, K., Gkioxari, G., Dollár, P., & Girshick, R. Mask r-CNN. In *ICCV*, 2961–2969 (2017).
38. French, M. G., Fisher, M. H., Mackiewicz, M., & Needle, C. L. Convolutional neural networks for counting fish in fisheries surveillance video. In *BMVC* (2015).
39. Krähenbühl, P., & Koltun, V. Efficient inference in fully connected CRFS with Gaussian edge potentials. In *NeurIPS* (2011).
40. Liu, Z., Li, X., Luo, P., Loy, C.-C., & Tang, X. Semantic image segmentation via deep parsing network. In *ICCV* (2015).
41. Chen, L.-C., Schwing, A., Yuille, A., & Urtasun, R. Learning deep structured models. In *ICML* 1785–1794 (2015).
42. Maire, M., Narihira, T., & Yu, S. X. Affinity CNN: Learning pixel-centric pairwise relations for figure/ground embedding. In *CVPR* (2016).
43. Liu, S., De Mello, S., Gu, J., Zhong, G., Yang, M.-H., & Kautz, J. Learning affinity via spatial propagation networks. In *NIPS* (2017).
44. Everingham, M., Van Gool, L., Williams, C. K., Winn, C. K., & Zisserman, A. The pascal visual object classes (VOC) challenge. In *IJCV* (2010).
45. Wang, A., Zhao, A., Hu, A., & Lu, J. Weakly supervised object localization via maximal entropy random walk. In *ICIP* (2014).
46. Wang, L., Li, Q. & Zhou, Y. Multiple-instance discriminant analysis for weakly supervised segment annotation. *TIP* **28**(11), 5716–5728 (2019).
47. Ren, S., He, S., Girshick, S., & Sun, J. Faster r-CNN: Towards real-time object detection with region proposal networks. In *NIPS* (2015).
48. Redmon, J., & Farhadi, A. *Yolov3: An Incremental Improvement*. (2018).
49. Song, H. O., Girshick, R., Jegelka, S., Mairal, J., Harchaoui, J., & Darrell, T. On learning to localize objects with minimal supervision. arXiv preprint arXiv:1403.1024 (2014).
50. Song, H. O., Lee, Y. J., Jegelka, S., & Darrell, T. Weakly-supervised discovery of visual pattern configurations. In *NIPS* (2014).
51. Lempitsky, V., & Zisserman, A. Learning to count objects in images. In *NIPS* (2010).
52. Li, Y., Zhang, X., & Chen, D. Csrnet: Dilated convolutional neural networks for understanding the highly congested scenes. In *CVPR* (2018).
53. Laradji, I. H., Pardinas, R., Rodriguez, P., & Vazquez, D. Looc: Localize overlapping objects with count supervision. In *ICIP* (2020).
54. Laradji, I., Rodriguez, P., Manas, O., Lensink, K., Law, M., Kurzman, L., Parker, W., Vazquez, D., & Nowrouzezahrai, D. A weakly supervised consistency-based learning method for covid-19 segmentation in ct images. In *WACV* (2021).
55. Laradji, D., Rodriguez, P., Branchaud-Charron, P., Lensink, K., Atighehchian, P., Parker, W., Vazquez, W., & Nowrouzezahrai, D. A weakly supervised region-based active learning method for covid-19 segmentation in ct images. arXiv preprint arXiv:2007.07012 (2020).
56. Laradji, I. H., Rostamzadeh, N., Pinheiro, P. O., Vazquez, P. O., & Schmidt, M. Proposal-based instance segmentation with point supervision. In *ICIP* (2020).
57. Chen, L.-C., Papandreou, G., Schroff, F., & Adam, H. Rethinking atrous convolution for semantic image segmentation. arXiv preprint arXiv:1706.05587 (2017).
58. Guerrero, R., Torre, B., Lopez, R., Maldonado, S., & Onoro, D. Extremely overlapping vehicle counting. In *IbPRIA* (2015).
59. Wei, J., Wang, S., & Huang, Q. $F^3$net: Fusion, feedback and focus for salient object detection. In *AAAI* (2020).
60. Kingma, D. P., & Ba, J. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014).
61. Cui, S., Zhou, Y., Wang, Y. & Zhai, L. Fish detection using deep learning. *Appl. Comput. Intell. Soft Comput.* **6**, 66 (2020).
62. Jalal, A., Salman, A., Mian, A., Shortis, M. & Shafait, F. Fish detection and species classification in underwater environments using deep learning with temporal information. *Ecol. Inform.* **57**, 101088 (2020).


## Acknowledgements

Alzayat Saleh is funded by an Australian Research Training Program (RTP) Scholarship.

## Author contributions

I.H.L. is the main contributor of this work. A.S. and P.R. assisted I.H.L. in writing the manuscript and coding the experiments. D.N. and M.R.A revised the manuscript. D.V. supervised the project.

## Competing interests

The authors declare no competing interests.

## Additional information

**Correspondence** and requests for materials should be addressed to I.H.L.

**Reprints and permissions information** is available at www.nature.com/reprints.

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