Underwater Fish Detection with Weak Multi-Domain Supervision

Dmitry A. Konovalov
College of Science and Engineering
James Cook University
Townsville, Australia
dmitry.konovalov@jcu.edu.au

Alzayat Saleh
College of Science and Engineering
James Cook University
Townsville, Australia
alzayat.saleh@my.jcu.edu.au

Michael Bradley
College of Science and Engineering
James Cook University
Townsville, Australia
michael.bradley@my.jcu.edu.au

Mangalam Sankupellay
College of Science and Engineering
James Cook University
Townsville, Australia
mangalam.sankupellay@jcu.edu.au

Simone Marini
Institute of Marine Sciences
National Research Council of Italy
Forte Santa Teresa, 19032, La Spezia, Italy
simone.marini@sp.ismar.cnr.it

Marcus Sheaves
College of Science and Engineering
James Cook University
Townsville, Australia
marcus.sheaves@jcu.edu.au

***Abstract*—Given a sufficiently large training dataset, it is relatively easy to train a modern convolution neural network (CNN) as a required image classifier. However, for the task of fish classification and/or fish detection, if a CNN was trained to detect or classify particular fish species in particular background habitats, the same CNN exhibits much lower accuracy when applied to new/unseen fish species and/or fish habitats. Therefore, in practice, the CNN needs to be continuously fine-tuned to improve its classification accuracy to handle new project-specific fish species or habitats. In this work we present a labelling-efficient method of training a CNN-based fish-detector (the Xception CNN was used as the base) on relatively small numbers (4,000) of project-domain underwater fish/no-fish images from 20 different habitats. Additionally, 17,000 of known negative (that is, missing fish) general-domain (VOC2012) above-water images were used. Two publicly available fish-domain datasets supplied additional 27,000 of above-water and underwater positive/fish images. By using this multi-domain collection of images, the trained Xception-based binary (fish/not-fish) classifier achieved 0.17% false-positives and 0.61% false-negatives on the project's 20,000 negative and 16,000 positive holdout test images, respectively. The area under the ROC curve (AUC) was 99.94%.**

***Index Terms*—fish, detection, convolution neural network, image, video**

I. Introduction

For the purpose of fish monitoring, remote underwater video (RUV) recording is a promising tool for fisheries, ecosystem management and conservation programs [1], [2]. RUV applications are primarily divided into *baited* [2] or *unbaited*. The focus of this study was the unbaited RUV processing because it uniquely offers the following benefits: information about early life-history stages, and the spatial distribution and temporal dynamics of juveniles. Such information is critical to fisheries and conservation management because it provides: (a) knowledge of juvenile habitats that need to be protected; (b) an understanding of the extent and direction of change of populations; (c) the ability to predict the size of future harvestable stocks; and (d) an understanding of the impact of habitat/environmental change on recruitment and survival through early life-history stages.

With the advent of consumer-grade action cameras, it is financially viable to deploy a large number of RUVs especially within the recreational scuba diving 30-meter depth limit. However, the amount of data that need to be processed from the deployed RUVs can quickly overwhelm the resources of human video viewers, often rendering video analysis prohibitively costly.

Conservation management requires unbaited RUVs to be placed in visually complex underwater habitats (Figs. 1 and 2), where the traditional fish detection methods become unreliable (see Section I-A). Modern Deep Learning [3] convolutional neural networks (CNNs) are currently achieving state of the art object-detection results in a wide variety of application domains; for example, automatic cattle detection from drones [4]. Since the majority of unbaited videos do not contain any fish, the maximum positive impact could be achieved by using a CNN to automatically detect and discard the *empty* video clips/frames.

In this study we developed a labelling-efficient procedure for training a CNN-based [5] binary image classifier (fish/non-fish) for fish-detection (see XFishHmMp in Section II-B) and fish-localization (see XFishHm in Section II-E). The structure of this paper is as follows. Section I-A reviews recent development in underwater fish detection and classification. Section II-A describes the labelling-efficient training and testing data preparation protocol. Section II-C presents the training pipeline. Section II-D introduces the main novel aspect of this work: weakly supervised training of the CNN fish-detector using external-to-project image domains. Section III presents the results from the project test images, which were not used in the training of the fish-detector.

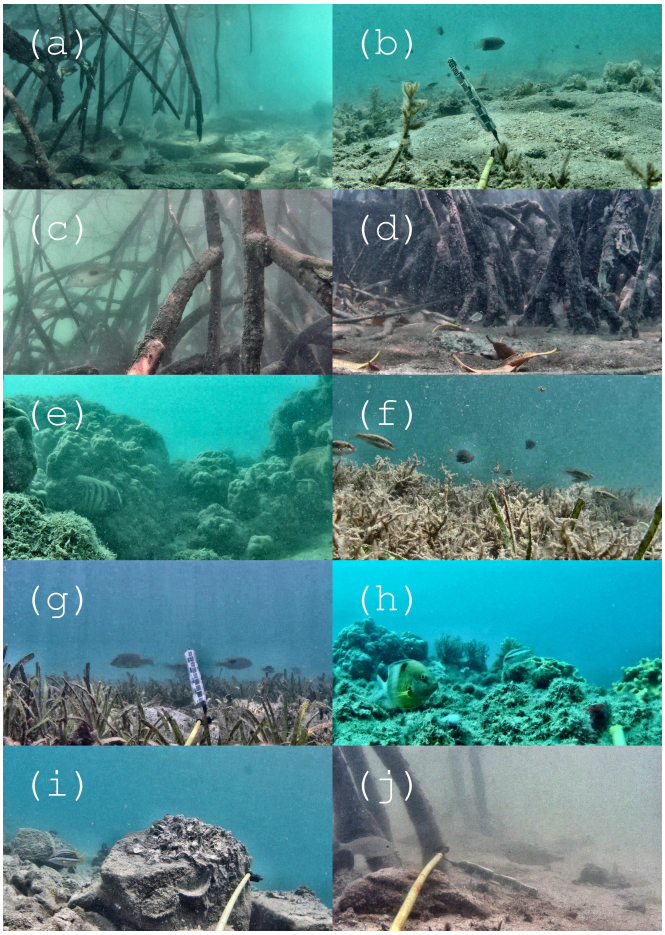

Fig. 1. Typical CLAHE processed fish-containing video frames from the first ten considered habitats.

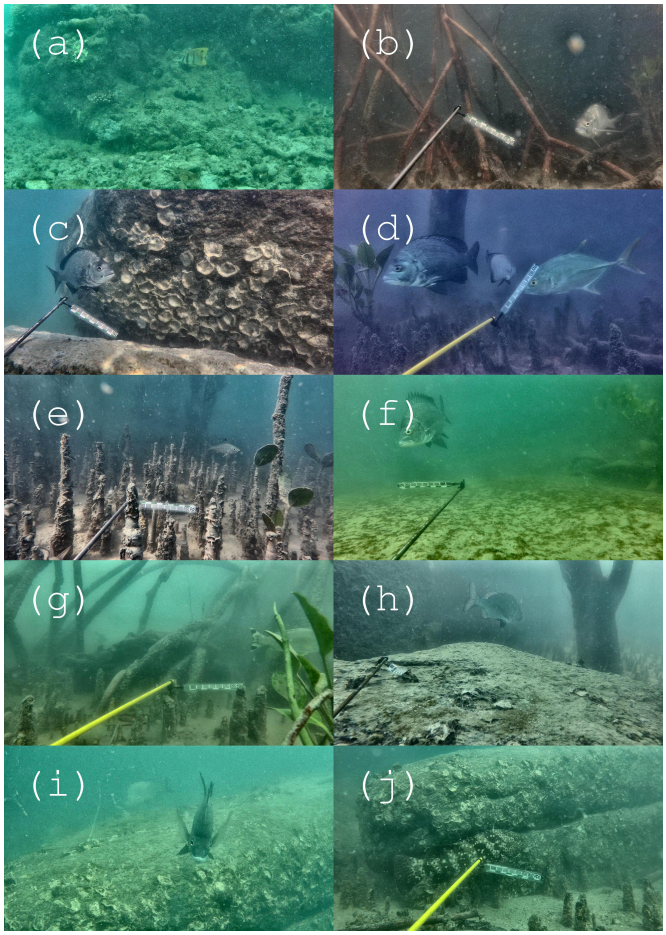

Fig. 2. Typical CLAHE processed fish-containing video frames from the 11th-20th habitats.

A. Related Work

The first large-scale automatic image and video-based fish detection and species classification study was the Fish4Knowledge (F4K) [6]–[9] project, which was run over five years during 2010-2015. F4K accumulated thousands of hours of underwater video clips of coral reefs in Taiwan. Similar (to our work) studies since the F4K project are reviewed next.

The manually-annotated LCF-14 dataset of 30,000 fish images and 1,000 video clips containing ten fish species was reported by [10]. The images and videos were used as the challenge dataset for the fish task of the LifeCLEF2014 [6] contest, and were derived from the F4K [7]–[9] project. The VLfeat-BoW [11], [12] classification method was used as the baseline for the task of recognizing fish in still images achieving 97% average precision (AP) and 91% average recall (AR), defined as [13]

$$AP = \frac{1}{c}\sum_{j=1}^{c} TP_j/(TP_j + FP_j), \tag{1}$$

$$AR = \frac{1}{c}\sum_{j=1}^{c} TP_j/(TP_j + FN_j), \tag{2}$$

where TP_j, FP_j and FN_j are the numbers of true-positive, false-positive and false-negative classified results for the jth species, respectively, and where c is the number of species. For the videos, the ViBe [14] background subtraction algorithm was applied first, then followed by the VLfeat-BoW [11], [12] achieving only $AP = AR = 54\%$. On the same LCF-14 test videos, very similar $AP \approx AR \approx 50\%$ was also reported by [15] using a Support Vector Machine (SVM) classifier. From (1), the significantly lower recall value (91%) compared to the average precision (97%) was due to the larger number of false-negatives FN_j (compared to false-positives FP_j). Furthermore, the dramatically worse results ($AP \approx AR = 50 - 54\%$) on videos highlighted the need for more accurate fish *detection* methods [10], [15], which is the main focus of this study.

The AlexNet CNN [16] together with the Fast-R-CNN [17] method were applied in [18] to classify 12 different fish species in 24,272 images, which were a manually curated subset of the train and test images from the fish task competition of LifeCLEF2014 [6], [10]. The LifeCLEF2014 fish

task images were derived from the F4K [7] collection. Fast-R-CNN is a Fast Region-based Convolutional Neural Network [17] method of object detection and classification in images. A mean average precision (mAP) of 81.4% was achieved [18] across the 12 considered species, where the mAP was defined as the total area under the precision-recall curve (see [18] for the exact definition). The most relevant aspect of [18] was that every train and test image was manually selected to contain one of the 12 species. Therefore, the method's ability to *detect* each species in the unconstrained underwater videos remained unknown.

The LCF-14 [10] dataset was used in [1] to create 32×32 gray training and test images. Then, the face recognition algorithm of [19] was applied to classify test images by finding the most similar species-specific images. Average classification accuracy of 94.6% was reported, which was a significant improvement over the conceptually similar method of sparse image representation [20]. However, the face-recognition [1] approach, and hence its accuracy, relied on an external method of [21] for extracting and cropping fish sub-images from a given video.

An earlier version of the mixture of Gaussians (MoG) [22] algorithm was used in [21] to segment moving fish from the stationary underwater background. The background-subtraction MoG [22] algorithm works extremely well when the variations in the *background* pixel intensities are on average less than the variations due to the moving *foreground* object. For example, the clear and debris-free water at the top-right corner of Fig 1(a) or the top of Fig 1(b). The MoG is readily available in many common software packages such as Matlab and OpenCV [23]. Unfortunately, the standard motion-based fish detection methods (for example, [21]), by design, could not distinguish between floating debris and juvenile fish of comparable size, or when the fish is stationary. For example, the fish in the center of Fig 1(j) is indistinguishable from the ground debris, and the fish in the left-middle (same sub-figure) remained stationary for many seconds. Furthermore, the MoG-type [22] methods fail when the background pixel variations are comparable with the slow-moving fish; for example, bottom-left *Lutjanus argentimaculatus* in Fig 1(a) or middle-left fish in Fig 1(c).

The LifeCLEF2015 Fish classification challenge dataset LCF-15 [24], [25] contained 20,000 labeled images, 93 videos and 15 fish species. Both LCF-14 [10] and LCF-15 datasets were used in [13], where the videos were processed by extracting frames as separate images, then all available images were resized to 32×32 shape and converted to grayscale. The use of only a three-layer CNN [13] achieved $AP = 97.18\%$ when the CNN was trained on LCF-15 but tested on LCF-14. However, when trained on LCF-14 but tested on the noisy and poorer quality LCF-15, the CNN's performance degraded to $AP = 65.36\%$. Furthermore, the same research group later reported [2] that the classification accuracy of the [13]'s CNN degraded from 87.46% on the LCF-15 dataset to 53.5% on a completely different dataset [2]. This performance drop illustrates the technology challenge faced by any fish-monitoring fishery or ecology project: it is unknown how to pre-train a generic fish detection CNN and use it with confidence in different environmental locations and/or to detect unknown (for the CNN) species. Hence, the financial and human cost of setting up and training a project-specific CNN becomes a critical factor, which is a key issue our work is trying to address.

Baited remote underwater cameras were used to collect videos from kelp, seagrass, sand and coral reef habitats in Western Australia [26]. The videos were processed in [2] to extract and label 2,209 images containing 16 fish species including *other* species as a separate 17th label. The images were re-sized and cropped to $224 \times 224 \times 3$ shape, where the three color channels were retained (hence the extra $\times 3$). Note the significant increase in image training complexity compared to the $32 \times 32 \times 1$ gray images of [1], [13]. The following three CNN architectures were used in [2]: AlexNet [16], VGGNet [27], and ResNet [28]. The CNNs' original layers were initialized by loading the weights pre-trained on the ImageNet's [29] vast collection of images, which is commonly referred to as the *transfer learning* or *knowledge transfer* setup or technique [30], [31]. An ImageNet-trained CNN often exhibits superior performance compared to the same but randomly initialized CNN, when the CNN is re-trained and/or re-purposed for different classes of images [31]; for example, the 16 fish species in [2]. The three considered ImageNet-trained CNNs were applied without further training to extract image features and then [2] used the features as input into a standard SVM classifier. Out of the considered three CNNs, the ImageNet pre-trained ResNet [28] together with the SVM classifier achieved the best accuracy of 89% on the testing subset of 663 images. Furthermore, the ResNet+SVM combination achieved even better accuracy of 96.73% on the LCF-15 [24], [25] dataset. Note that similar to the preceding Fast-R-CNN [18] and face-recognition-type [1] studies, the ResNet+SVM [2] focused on the classification of externally (and manually) detected and appropriately cropped images. Therefore, the reported ResNet+SVM's high classification accuracy could only be achieved if it is accompanied by an automatic fish detection and bounding-box segmentation method of comparably high accuracy, which at present are estimated as only 50% accurate [10], [15].

Focusing only on the fish detection task, [32] reported a method of automatic fish counting in real-world videos, which is referred to as the OBSEA method hereafter. A binary (fish/no-fish) classifier was trained on 11,920 images collected at the OBSEA testing-site [33] in 2012, and then tested on 10,961 imaged acquired at the same site in 2013. The OBSEA method consisted of two distinct steps. Within the first step, Regions of Interest (RoI) were automatically extracted from all training and test images. The RoI step used consecutive images sorted by the acquisition time and then essentially extracted the image differences as RoIs. Conceptually the RoI step is identical to the MoG-type [22] methods and therefore arguably would exhibit similar limitations: a large percentage of false-negatives when the fish is stationary, slow moving or below the

adopted detection threshold. The figures and supplementary video of [32] clearly demonstrated this issue, where many fish instances were not segmented by the RoI step. The second step of the OBSEA method applied a genetic programming method from [34] to deliver a binary fish/not-fish classification for each of the segmented RoIs from the first step. Within a 10-fold cross-validation framework, the classifier achieved 92% validation accuracy on manually labelled RoIs. Note that similar to the ResNet+SVM [2] results, the reported accuracy can only be achieved on the per-fish/per-image basis if the preceding RoI or bounding-box segmentation step delivers appropriately low false-negative and false-positive rates, which was not reported in [32].

Based on this review of recent studies, the following working hypotheses were adopted for our study:

- Given a RoI or a bounding box in an image, there have been a number of methods achieving 85%-95% accuracy of correctly classifying fish species or fish/not-fish detection.
- All reviewed classifiers required human-intensive annotation/labelling of RoI/bounding-boxes for each training image.
- Trained classifiers are highly specialized to the training fish species and/or the training environmental habitat, and should not be assumed to work equally well on different species and/or different backgrounds.
- The accuracy of automatic fish-related RoI/segmentation methods is highly dependent on the image/video background/habitat.
- In complex reef-type habitats, the RoI extraction methods are only 50%-80% accurate, which is significantly less accurate than the classifiers from the corresponding studies.

II. Materials and Methods

A. Labelling-Efficient Dataset Preparation Protocol

The essential goal of this study was to design and test a practical and labelling-efficient data preparation protocol, which could be used in future fish-survey studies. The following protocol was utilized with realistic estimations of the required human labor for project planning and costing.

Video clips from 20 diverse habitats were selected (see typical examples in Figs. 1 and 2). Video clips were recorded in a range of different environmental conditions present across a near-shore island chain of the Great Barrier Reef (Palm Islands, Queensland, Australia). These clips represent the range of different conditions encountered during a tropical marine fish survey, and form part of the field data presented in an assessment of juvenile fish habitat [36]. Recording sites varied in three-dimensional habitat architecture, levels of natural shading, current and wave energy levels, levels of suspended sediments, organic flocculation (marine snow), turbidity and salinity.

The video clips were visually examined to determine if they contained at least one fish. Then all clips containing

Fig. 3. Examples of frames from mangroves habitat with: one *Lutjanus argentimaculatus* adult (top row); one *Chaetodon vagabundus* (bottom row); and multiple *Caranx sexfasciatus* juveniles (middle row). Top-left, middle-right and bottom-row sub-images were histogram equalized via CLAHE [35].

fish were placed into sub-folder named *valid*, while all the clips without any fish species were placed into the sub-folder named *empty*. All but one habitat (Fig. 1b) had at least one valid and one empty clip, where the collected valid and empty clips could be reused in the future projects to gradually build a more comprehensive fish-detection training dataset. This sorting took approximately two days (10 hours) for an experienced marine biologist already familiar with the content of the videos. All clips were then converted to individual frame images where the first, 11th, 21st, etc. frames (intervals of 10) were saved for training (denoted as FD10) and the remaining frames were saved separately for testing (denoted as FD10-Test). In total the clips yielded 40,000 frames, where the FD10 dataset contained 1764 positive (fish) and 2253 negative (no-fish) images. The FD10-Test dataset contained 16,000 positive and 20,000 negative images. This clip-level labelling is highly human-labor efficient in generating thousands of project-specific image-level annotations. However, the proposed labelling procedure is valid only if both fish and no-fish clips are available from the same location. A CNN model needs to learn the fish features (only present in positive clips) and learn to ignore the underwater habitat features (available in both negative and positive clips).

The FD10 and FD10-Test collections of the unprocessed original frames were further processed by the CLAHE [35] algorithm from the OpenCV [23] library to create the corresponding FD10c and FD10c-Test datasets. The CLAHE default clip limit value was retained at 2, and the CLAHE

tile grid sizes were set at 16 column tiles and 8 row tiles. Each image was first converted from the RGB color space to the CIELAB [37] space via the relevant OpenCV function, where the luminosity (L) channel was then processed by the CLAHE algorithm. Visually, the CLAHE processed frames were significantly better than the unprocessed raw video frames (see the effect of CLAHE in Fig. 3).

The original 1080×1920 ($rows \times columns$) resolution project videos (not the training clips) were approximately 1TB in combined disk storage size and 200 hours in total duration time (600 videos, each 20 min long), which required at least 200 hours of paid (or effectively paid via the lost opportunity-cost) processing by marine biologists. More than half of the video frames did not contain any fish species, which meant at least 100 human hours were wasted analyzing the original videos. The unproductive human effort was further compounded by searching within the videos for the relevant fish-containing sections of interest. Since such fish surveys are repeated regularly, the presented fish-detection method and procedures could be further refined in future studies.

B. Project-Domain

For the supervised training, the most human-efficient labelling is achieved by the image-level class labels [38]. If annotated by a class label, one or more instances of the class are curated to be present somewhere in the image scene, directly visible or directly inferrable.

Out of the common ImageNet-trained and available in Keras [39] CNNs, Xception [5] was selected as the base CNN. The required binary fish/no-fish classifier (denoted as XFishMp) was constructed by replacing the Xception's 1,000-class top with one spatial/global maximum pooling layer (hence the "Mp" abbreviation in XFishMp) followed by a 0.5-probability dropout layer, and then by a one-class dense layer with the *sigmoid* activation function. The Xception-based XFishMp contained the smallest number of trainable parameters (20.8 million) compared to 23.5 million in the ResNet50-based and 21.7 million in the InceptionV3-based XFish's equivalents. Another CNN configuration was also considered and denoted XFishHmMp, where the XFishMp's global max-pool was moved to be the last layer and the one-class dense layer was converted to a convolution layer. The "Hm" naming mnemonic was due to the one-class convolution layer yielding a two-dimensional *heatmap* of $[0, 1]$-ranged values.

In this study, the FD10 dataset consisted of 4017 color images (1764 with fish and 2253 without fish), where each image had 1080 pixel rows and 1920 columns (1080×1920 shape). The fish sizes were mostly within the $[30, 300]$-pixel range.

In comparison with the ImageNet's more than one million images (used to train Xception), the FD10 dataset is very small (4017 images). Hence, additional measures were required to prevent over-fitting of the XFish CNNs. The first such measure was to use the training color images only via their grayscale versions. As the color variation in reef fish species is generally greater than the variation in fish shapes, by removing the color features, the XFish CNNs were potentially more likely to learn (*generalize*) the fish shapes, rather than to memorize (*fit*) pixel colors. Furthermore, the underwater background colors vary dramatically (Figs. 1 and 2) and therefore the considered FD10's 4017 images could be classified more easily if the color channels were included. However, such fitting of the colors would have little or no generalization value beyond the considered training dataset.

Since the ImageNet-trained Xception required the three color channels in its input, a trainable gray-to-RGB convolution conversion layer was added to the front of the XFish CNNs to accept the one-channel grayscale training images.

In order to achieve practical training times and to fit the training onto common GPUs (Nvidia GTX 1070 and GTX 1080 Ti were used), the training image dimensions were limited to the 512×512 shape. Then, in addition to the grayscale input images, the following augmentations were performed to reduce over-fitting (that is, additional regularization). Each original 1080×1920 image was converted to the grayscale and then zero padded by a 5% border yielding 1188×2112 shaped images. The padded images were downsized (or zero-padded if required in Section II-D) to the 512×512 shape and then:

- randomly rotated within $[-20, 20]$ degree range;
- randomly flipped horizontally with the 0.5 probability;
- their rows and columns were resized independently by random scales from $[0.9, 1]$ range;
- after zero-padding to the 512×512-shape, random perspective transformation was applied;
- normal Gaussian noise was added and the final image values were clipped to $[0, 255]$ range, where the noise mean was zero, and the noise standard deviation was randomly selected from $[0, 8]$ range;
- the grayscale $[0, 255]$-range pixel values were normalized to zero minimum and maximum of one with each image.

C. Training Pipeline

All considered models were trained in Keras [39] with the Tensorflow [40] back-end, where the Adam [41] algorithm was used as the training optimizer. The Adam's initial learning-rate (lr) was set to $lr = 1 \times 10^{-5}$ for training XFishMp and to 1×10^{-4} for XFishHmMp, where the rate was halved every time the epoch *validation* accuracy did not increase after 10 epochs. The training was done in batches of four images and was restarted twice from the highest-accuracy model if the validation accuracy did not increase after 32 epochs, where at each restart the initial lr was multiplied by 0.9. The validation subset of images was not augmented but only pre-processed: 5% zero-padded, resized to the 512×512 input shape, and normalized to the $[0, 1]$ range.

D. Weak Supervision by External Domains

Arguably, the only reason to use the project-domain datasets is the absence of public fish-domain image/video datasets of the required fish-species and of required image quality and quantity. However, there are many general-domain image

datasets where fish instances are labelled (for example, ImageNet [29]) or known to be missing (for example, VOC2012 [42]).

The Xception CNN utilized here was trained on more than one million ImageNet images (including some fish images). The project's FD10 collection of 4,000 training images was still very small for the modern high capacity CNNs, such as Xception. Therefore, in this study, we regularized XFish CNNs by using negative (not-fish) general-domain images, where the 17,000 VOC2012 [42] images were used in this study to achieve weak negative supervision. All of the original videos (used as the base for this project's training clips) contained the above-water sections at the beginning of each video, when the camera was manually turned on before being lowered to its underwater destination. The negative every-day type VOC2012 images assisted in more robust rejection of the above-water false-positives.

For the weak positive supervision, two specialized fish-domain datasets were utilized: the LCF-15 classification challenge dataset [24], [25] with 22.4 thousand fish images, and the QUT2014 dataset [43], [44] with 4.4 thousand fish images.

In order to retain the weak nature of the external mutli-domain datasets, we proposed the following training pipeline. The FD10 was enlarged by total of 4,000 images (denoted as FD10-VLQ), where 2,000 images were randomly selected from VOC2012 (automatically labeled as negative/no-fish) and 1,000 images from each LCF-15 and QUT2014 (automatically labeled as positive/fish). Then, the new 8,000 large FD10-VLQ dataset was split 80/20% into the training/validation subsets. Since many more images remained available in all three considered domain-level datasets, at each training epoch, all 4,000 additional external-domain images were randomly re-drawn from their corresponding datasets.

E. Fish Localization

Fish detection normally implies *localization* of the detected fish within an image. XFishHmMp could be easily converted for the localization task, by removing its last max-pooling layer arriving at the XFishHm CNN. XFishHm outputs a grayscale heatmap of the input image spatially downsized by 32, that is, the 512×512 grayscale image is converted into 16×16 heatmap of $[0, 1]$-ranged values. Weak localization supervision was achieved by deliberately (and human-time efficiently) selecting the fish-containing and missing-fish FD10 video clips from the same underwater locations. Note that due to their higher labelling costs, the direct fish-level supervision via, for example, bounding-box [45], pixel-level *semantic segmentation* or point-level [38] annotations were considered outside the scope of this study.

III. Results and Discussion

A. Baseline

To establish the baseline, XFishMp and XFishHmMp were trained in Keras [39] with the Tensorflow [40] back-end on the identical random (controlled by a fixed seed value) train/validation split of the FD10 and FD10c datasets, where the label-stratified split was 80% for training and 20% for validation. The binary cross-entropy was used as the training loss.

All FD10 and FD10c trained models (Table I) were applied to the FD10-Test dataset, which was not processed by CLAHE [35]. On NVIDIA GTX 1080 Ti, the networks processed the test images at 7-8 images per second (one image per batch), which was borderline acceptable for processing the large volume of underwater videos in deployment, for example, by further optimization of running in larger batches and/or only loading every second or third frames. However, additional CLAHE pre-processing reduced the testing rate to 0.5-1 images per second and therefore was not considered as a currently-viable deployment option.

Since every 10th frame was used for training (or validation), it was reasonable to expect that the remaining test frames (from the holdout FD10-Test dataset) would be classified exactly (zero false-negatives and zero false-positives). The default 0.5 threshold was used to accept the CNN activation output as positive/fish, and classify as negative/no-fish if the output value was less than the threshold. The lowest baseline false-positive rate ($FP/N = 0.25\%$) was achieved by XFishMp (trained on FD10), see Table I, while lowest baseline false-negative rate ($FN/P = 0.84\%$) was by the heatmap-based XFishHmMp (trained on FD10). In Table I, N and P denoted the total number of negative and positive test images, respectively.

The training on the *cleaner* CLAHE-processed FD10c images, reduced the CNN's generalization ability, where the best baseline false-positive rate deteriorated from $FP/N = 0.25\%$ to 2.39% for the XFishMp+FD10c CNN. A conceptually similar result was reported by [13], where training on the noisy LCF-15 dataset achieved higher accuracy (tested on cleaner LCF-14) than training on clean LCF-14 and testing on noisy LCF-15. Therefore, while visually appealing, the image-cleaning pre-processing is not necessary and could even be detrimental to the CNN performance.

B. Multi-Domain Image-Level Supervision

To observe the effect of the additional domain-level weak supervision, the baseline-trained XFishMp and XFishHmMp CNNs were fine-tuned on the FD10-VLQ and FD10c-VLQ dataset (see Section II-D). Note that the CLAHE pre-processing was not applied to any of the external images. The training pipeline (Section II-C) remained nearly identical, where the corresponding starting learning rates were reduced by the factor of 10, and only one training cycle was used. This means the training was not restarted once aborted.

The weak supervision by external images improved all of the baseline cases (Table I) to some degree. The heatmap-based XFishHmMp CNN (trained on raw FD10-VLQ) achieved the lowest possible false-positive ($FP/N = 0.17\%$) and false-negative ($FN/P = 0.61\%$) rates. Only two cases did not improve with the selected *default* detection threshold of 0.5: false-positives (FP) of XFishMp(FD10-VLQ) and false-negatives (FN) of XFishMp(FD10c-VLQ).

TABLE I
CONFUSION MATRIX FOR THE FD10-TEST DATASET

Actual	Model / *Train dataset*	*Negative*	*Positive*
Negative (no-fish)	**XFishMp**	***TN***	***FP (FP/N%)***
	FD10c	19,623	481 (2.39%)
	FD10c-VLQ	19,849	255 (1.27%)
N=20,104	FD10	20,053	51 (0.25%)
	FD10-VLQ	20,005	99 (0.49%)
	XFishHmMp		
	FD10c	18,378	1,726 (8.58%)
	FD10c-VLQ	18,638	1,466 (7.29%)
	FD10	19,998	106 (0.53%)
	FD10-VLQ	20,070	34 (**0.17%**)
Positive (fish)	**XFishMp**	***FN (FN/P%)***	***TP (AUC%)***
	FD10c	876 (5.28%)	14,725 (99.24%)
	FD10c-VLQ	884 (5.32%)	14,717 (99.31%)
P=16,601	FD10	162 (0.98%)	15,439 (99.92%)
	FD10-VLQ	117 (0.71%)	15,484 (99.96%)
	XFishHmMp		
	FD10c	1,713 (10.32%)	13,888 (96.48%)
	FD10c-VLQ	1,564 (9.42%)	14,037 (96.90%)
	FD10	139 (0.84%)	15,462 (99.92%)
	FD10-VLQ	101 (**0.61%**)	15,500 (99.94%)

However, the receiver operating characteristics' (ROC) area under the curve (AUC) [46] revealed that even in the two cases, a better separation of positive and negative activation values was achieved (see the bottom-right sub-column of Table I).

Clearly, the additional positive weak supervision could not improve the false-negative rate (FN) significantly, where the external fish images (in LCF-15 [24], [25] and QUT2014 [43], [44]) were very different from the project-domain fish images. The external negative weak supervision was more likely to improve the false-positive rate (FP), which indeed decreased in XFishMp(FD10c) from 481 to 255 (Table I).

C. Weakly Supervised Localization

The heatmap-based XFishHmMp CNN achieved the lowest final FP and FN errors (Table I). After removing the last max-pooling later, XFishHmMp was converted to the localization XFishHm CNN (see Section II-E). Detailed analysis of the localization accuracy was left to future work as it required the ground-truth bounding-boxes or segmentation masks for the FD10-Test images. Nevertheless, XFishHm was applied to all FD10 images and results were visually inspected to verify good consistency of the heatmap fish localization. Typical heatmap segmented examples are presented in Fig. 4, where the heatmaps were re-scaled to the original training size of 512×512 (from the XFishHm 16×16 output).

Since the XFishMp architecture did not exhibit consistently superior accuracy compared to XFishHmMp (Table I), and XFishMp could not be instantly converted to output the heatmaps, we accepted XFishHmMp as the starting architecture for future work. Note, that the Xception CNN base in XFishHmMp could be trivially replaced by any other modern CNN, where any required input image normalization is automatically taken care by the trainable gray-to-RGB conversion layer.

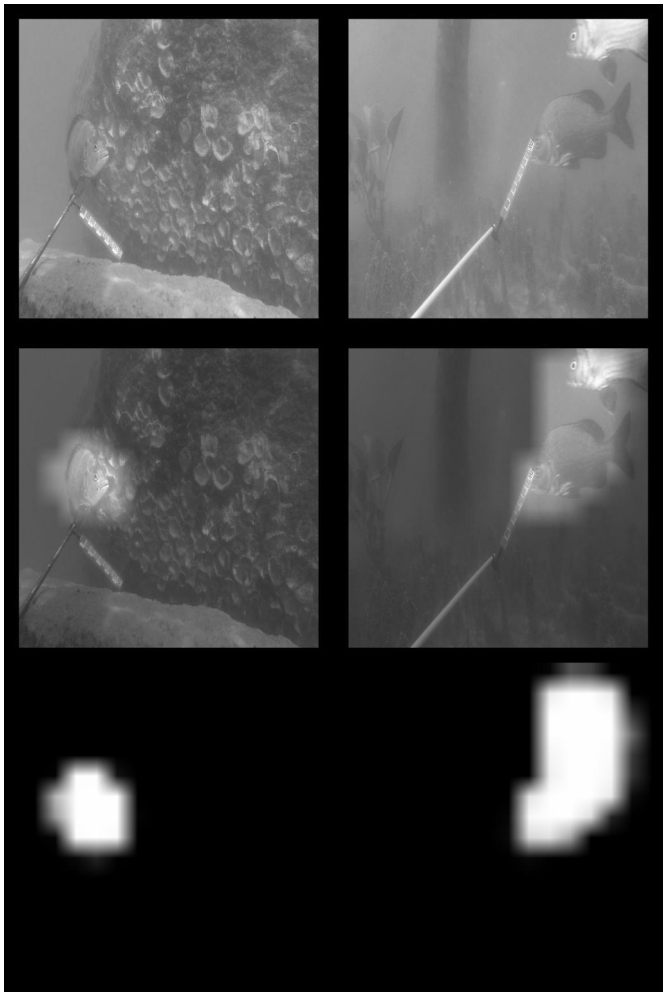

Fig. 4. Typical examples of fish correctly detected and localized by XFishHm (trained as XFishHmMp), where the top subfigures are the padded and re-scaled original grayscale images, middle are the images overlapped with the prediction heatmaps, and the bottom row are the prediction heatmaps.

IV. CONCLUSION

In conclusion, we developed a novel training procedure for a relatively small number of project-domain images to be utilized more effectively when training a project-specific CNN fish-detector together with a much larger pool of multi-domain images. A human-time efficient labelling procedure was successfully tested. The regularizing effect of the weak supervision by external large multi-domain image collections was verified. Pre-processing image cleaning could reduce the model generalization performance.

REFERENCES

[1] F. Shafait, A. Mian, M. Shortis, B. Ghanem, P. F. Culverhouse, D. Edgington, D. Cline, M. Ravanbakhsh, J. Seager, and E. S. Harvey, "Fish identification from videos captured in uncontrolled underwater environments," *ICES Journal of Marine Science*, vol. 73, pp. 2737–2746, 2016.

[2] S. A. Siddiqui, A. Salman, M. I. Malik, F. Shafait, A. Mian, M. R. Shortis, and E. S. Harvey, "Automatic fish species classification in underwater videos: exploiting pre-trained deep neural network models to

compensate for limited labelled data," *ICES Journal of Marine Science*, vol. 75, pp. 374–389, 2018.
[3] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, 2015.
[4] A. Rivas, P. Chamoso, A. Gonzlez-Briones, and J. M. Corchado, "Detection of cattle using drones and convolutional neural networks," *Sensors*, vol. 18, no. 7, 2018.
[5] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR2017)*, 2017.
[6] A. Joly, H. Goëau, H. Glotin, C. Spampinato, P. Bonnet, W.-P. Vellinga, R. Planque, A. Rauber, R. Fisher, and H. Müller, "Lifeclef 2014: Multimedia life species identification challenges," in *Information Access Evaluation. Multilinguality, Multimodality, and Interaction*, ser. Lecture Notes in Computer Science, E. Kanoulas, M. Lupu, P. Clough, M. Sanderson, M. Hall, A. Hanbury, and E. Toms, Eds., vol. 8685. Cham: Springer International Publishing, 2014, pp. 229–249.
[7] B. J. Boom, J. He, S. Palazzo, P. X. Huang, C. Beyan, H.-M. Chou, F.-P. Lin, C. Spampinato, and R. B. Fisher, "A research tool for long-term and continuous analysis of fish assemblage in coral-reefs using underwater camera footage," *Ecological Informatics*, vol. 23, pp. 83 – 97, 2014, special Issue on Multimedia in Ecology and Environment.
[8] R. B. Fisher *et al.* [Online]. Available: www.fish4knowledge.eu
[9] ——. [Online]. Available: https://bit.ly/2Ex7dnZ
[10] C. Spampinato, S. Palazzo, P. H. Joalland, S. Paris, H. Glotin, K. Blanc, D. Lingrand, and F. Precioso, "Fine-grained object recognition in underwater visual data," *Multimedia Tools and Applications*, vol. 75, pp. 1701–1720, 2016.
[11] A. Vedaldi and B. Fulkerson, "VLFeat: An open and portable library of computer vision algorithms," 2008. [Online]. Available: http://www.vlfeat.org
[12] ——, "VLfeat: An open and portable library of computer vision algorithms," in *Proceedings of the 18th ACM International Conference on Multimedia*, ser. MM '10. New York, NY, USA: ACM, 2010, pp. 1469–1472.
[13] A. Salman, A. Jalal, F. Shafait, A. Mian, M. Shortis, J. Seager, and E. Harvey, "Fish species classification in unconstrained underwater environments based on deep learning," *Limnology and Oceanography: Methods*, vol. 14, pp. 570–585, 2016.
[14] O. Barnich and M. V. Droogenbroeck, "ViBe: A universal background subtraction algorithm for video sequences," *IEEE Transactions on Image Processing*, vol. 20, no. 6, pp. 1709–1724, 2011.
[15] K. Blanc, D. Lingrand, and F. Precioso, "Fish species recognition from video using svm classifier," in *Proceedings of the 3rd ACM International Workshop on Multimedia Analysis for Ecological Data*, ser. MAED '14. New York, NY, USA: ACM, 2014, pp. 1–6.
[16] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, ser. NIPS'12. USA: Curran Associates Inc., 2012, pp. 1097–1105.
[17] R. Girshick, "Fast r-cnn," in *The IEEE International Conference on Computer Vision (ICCV)*, December 2015.
[18] X. Li, M. Shang, H. Qin, and L. Chen, "Fast accurate fish detection and recognition of underwater images with fast r-cnn," in *OCEANS 2015 - MTS/IEEE Washington*, 2015, pp. 1–5.
[19] Y. Hu, A. S. Mian, and R. Owens, "Face recognition using sparse approximated nearest points between image sets," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, pp. 1992–2004, 2012.
[20] Y.-H. Hsiao, C.-C. Chen, S.-I. Lin, and F.-P. Lin, "Real-world underwater fish recognition and identification, using sparse representation," *Ecological Informatics*, vol. 23, pp. 13 – 21, 2014, special Issue on Multimedia in Ecology and Environment.
[21] C. Spampinato, Y.-H. Chen-Burger, G. Nadarajan, and R. B. Fisher, "Detecting, tracking and counting fish in low quality unconstrained underwater videos," in *Proceedings of 3rd International Conference on Computer Vision Theory and Applications (VISAPP)*, vol. 2, 2008, pp. 514–519.
[22] Z. Zivkovic and F. van der Heijden, "Efficient adaptive density estimation per image pixel for the task of background subtraction," *Pattern Recognition Letters*, vol. 27, pp. 773 – 780, 2006.
[23] Itseez, "Open source computer vision library," https://github.com/itseez/opencv, 2017.
[24] A. Joly, H. Goëau, H. Glotin, C. Spampinato, P. Bonnet, W.-P. Vellinga, R. Planqué, A. Rauber, S. Palazzo, B. Fisher, and H. Müller, "Lifeclef 2015: Multimedia life species identification challenges," in *Experimental IR Meets Multilinguality, Multimodality, and Interaction*, ser. Lecture Notes in Computer Science, J. Mothe, J. Savoy, J. Kamps, K. Pinel-Sauvagnat, G. Jones, E. San Juan, L. Capellato, and N. Ferro, Eds., vol. 9283. Cham: Springer International Publishing, 2015, pp. 462–483.
[25] "Fish species recognition." [Online]. Available: https://bit.ly/2LomRTp
[26] E. S. Harvey, M. Cappo, G. A. Kendrick, and D. L. McLean, "Coastal fish assemblages reflect geological and oceanographic gradients within an australian zootone," *PLOS ONE*, vol. 8, 2013.
[27] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2014.
[28] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.
[29] J. Deng, W. Dong, R. Socher, L. J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248–255.
[30] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "Cnn features off-the-shelf: An astounding baseline for recognition," in *2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2014, pp. 512–519.
[31] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, "Learning and transferring mid-level image representations using convolutional neural networks," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2014, pp. 1717–1724.
[32] S. Marini, E. Fanelli, V. Sbragaglia, E. Azzurro, J. Del Rio Fernandez, and J. Aguzzi, "Tracking fish abundance by underwater image recognition," *Scientific Reports*, vol. 8, p. 13748, 2018.
[33] "The western mediterranean expandable seafloor observatory (OBSEA)." [Online]. Available: http://www.obsea.es
[34] L. Corgnati, S. Marini, L. Mazzei, E. Ottaviani, S. Aliani, A. Conversi, and A. Griffa, "Looking inside the ocean: Toward an autonomous imaging system for monitoring gelatinous zooplankton," *Sensors*, vol. 16, 2016.
[35] K. Zuiderveld, *"Contrast Limited Adaptive Histogram Equalization"*. San Diego: "Academic Press Professional", 1994, pp. 474–485.
[36] M. Bradley, R. Baker, I. Nagelkerken, and M. Sheaves, "Context is more important than habitat type in determining use by juvenile fish," *Landscape Ecology*, 2019, in press.
[37] International Color Consortium, *Specification ICC.1:2004-10 (Profile version 4.2.0.0) Image technology colour management - Architecture, profile format, and data structure*, 2004. [Online]. Available: http://www.color.org/icc1v42.pdf
[38] O. Russakovsky, A. L. Bearman, V. Ferrari, and F. Li, "What's the point: Semantic segmentation with point supervision," *CoRR*, vol. abs/1506.02106, 2015.
[39] F. Chollet *et al.*, "Keras: The python deep learning library," 2015. [Online]. Available: https://keras.io/
[40] M. Abadi *et al.*, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015. [Online]. Available: http://tensorflow.org/
[41] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014.
[42] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes challenge: A retrospective," *International Journal of Computer Vision*, vol. 111, pp. 98–136, 2015.
[43] K. Anantharajah, Z. Ge, C. McCool, S. Denman, C. Fookes, P. Corke, D. Tjondronegoro, and S. Sridharan, "Local inter-session variability modelling for object classification," in *IEEE Winter Conference on Applications of Computer Vision*, 2014, pp. 309–316.
[44] "QUT fish dataset." [Online]. Available: https://bit.ly/2APDvGB
[45] I. H. Laradji, N. Rostamzadeh, P. O. Pinheiro, D. Vazquez, and M. Schmidt, "Where are the blobs: Counting by localization with point supervision," in *Computer Vision – ECCV 2018*, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds. Cham: Springer International Publishing, 2018, pp. 560–576.
[46] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognition Letters*, vol. 27, pp. 861–874, 2006.