Fast Accurate Fish Detection and Recognition of Underwater Images with Fast R-CNN

Xiu Li[1, 2], Min Shang[1, 2], Hongwei Qin[1, 2], Liansheng Chen[1, 2]
1. Department of Automation, Tsinghua University, Beijing 100084
2. Graduate School at Shenzhen, Tsinghua University, Shenzhen 518055
li.xiu@sz.tsinghua.edu.cn，sm14@mails.tsinghua.edu.cn
qhw12@mails.tsinghua.edu.cn，cls13@mails.tsinghua.edu.cn

***Abstract*—This paper aims at detecting and recognizing fish species from underwater images by means of Fast R-CNN (Regions with Convolutional Neural and Networks) features. Encouraged by powerful recognition results achieved by Convolutional Neural Networks (CNNs) on generic VOC and ImageNet dataset, we apply this popular deep ConvNets to domain-specific underwater environment which is more complicated than overland situation, using a new dataset of *24277* ImageCLEF fish images belonging to *12* classes. The experimental results demonstrate the promising performance of our networks. Fast R-CNN improves mean average precision (mAP) by *11.2%* relative to Deformable Parts Model (DPM) baseline-achieving a mAP of *81.4%*, and detects *80*× faster than previous R-CNN on a single fish image.**

Keywords—Fish detection and recognition, Fast R-CNN, underwater images, deep ConvNets

I. INTRODUCTION

Deep-sea observation systems such as the NEPTUNE and VENUS observatories [6], has been widely used in recent years for marine surveillance with a large amount of information-rich underwater videos and images. Estimating fish existence and quantity from these videos and images can help supporting marine biologists to understand the natural underwater environment, promote its preservation, and study behaviors and interactions between marine animals that are part of it [1]. However, it will be a tedious and time-consuming job for humans to manually analyze massive quantity of underwater video and image data daily generated. Therefore an automatic fish detection and recognition system is of crucial importance and practice, to reduce time-consuming and expensive input by human observers.

But so far, limited pattern recognition methods are used to detect and recognize fish species in underwater images. Mehdi *et al*. [2] used Haar classier to classify the shape features modelled by Principal Component Analysis (PCA). Concetto *et al*. [10] used a moving average algorithm to get balance between processing time and accuracy over the static scenarios for underwater fish detection. However, These methods mentioned above tend to be complicated in feature extraction, and have a poor ability in dealing with large amount of underwater images scalably.

Encouraged by the significant results achieved by deep convolutional neural networks on generic ImageNet [5] and VOC [16] datasets, we introduce this popular architecture to specific marine-domain fish detection and recognition for its high computing capability and scalability. The interest in CNNs has been widely raised after Krizhevsky *et al* [3] showed substantially higher image classification accuracy on the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [4,5], in 2012. Effects of varieties of computer vision tasks such as detection, recognition, semantic segmentation, scene classification, and domain adaptation, have been updated using CNNs. In 2014, Ross *et al* [7] proposed a simple and scalable detection algorithm called R-CNN that improves mAP by more than *30%* relative to the previous best result on VOC 2012—achieving a mean average precision (mAP) of *53.3%*. Concurrent with R-CNN, K. He *et al* [8] put forward SPP-net with comparable accuracy and considerable speedup on VOC 2007 dataset, compared to R-CNN. Inspired by K.He's work, Ross [9] put forward Fast R-CNN—an improvement of R-CNN ,achieving higher mAP and faster speed on VOC datasets.

Since that deep learning has achieved tremendous improvement in visual detection and recognition with thousands of categories [5], we study the performance of Fast R-CNN in underwater environment in this paper, given large-scale training data and high-performance computational infrastructure. Unlike ImageNet and VOC datasets of high-resolution images, oceanic images may be in poor quality condition because of light scattering, color change and device reasons. In this paper, we apply Fast R-CNN of high accuracy and detecting speed to complex underwater environment for fish detection and recognition. Figure 1 presents an overview of our method. The networks take as input a RGB image and its 2000 regions of interest (RoIs) collected by selective search [11] and produces a distribution over fish classes as well as related bounding-boxes. In summary ,the contributions of this paper are three-fold : first, we incorporate deep learning into complex underwater environment; second, we build an automatic fish detection and recognition system with Fast R-CNN, achieving convincing performance in detection precision and speed; third, we construct a clean fish dataset with *24272* images over *12* classes, a subset of ImageCLEF [1] training and test datasets, to evaluate our system at scale.

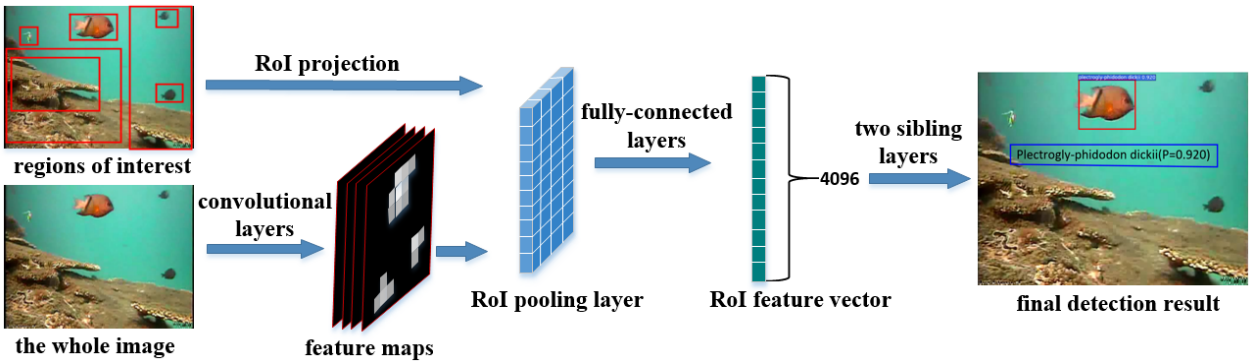

Figure 1 : Overall architecture of automatic fish detection and recognition system using Fast R-CNN. The networks take as input a RGB fish image and its 2000regions of interest (RoIs) collected by selective search and produces a distribution over fish classes as well as related bounding-box values.

Fish species	train	val	test	trainval
Acanthurus Nigrofuscus	534	536	1017	2070
Amphiprion Clarkii	1042	1031	1028	2037
Chaetodon Lunulatus	1181	1196	763	2377
Chromis Chrysura	1100	1094	1055	2194
Dascyllus Aruanus	328	339	227	667
Dascyllus Reticulatus	1729	1796	1491	3525
Hemigymnus Fasciatus	571	562	506	1133
Lutjanus Fulvus	220	220	167	440
Myripristis Kuntee	890	887	824	1777
Neoniphon Sammara	650	650	639	1300
Pempheris Vanicolensis	349	364	286	713
Plectrogly-Phidodon Dickii	921	925	934	1846
total images	8233	8227	7817	16460

Table 1 : ImageCLEF_Fish_TS dataset. The proportion of training set, validation set and testing set.

II. Dataset Acquisition

To evaluate our automatic fish detection and recognition system, we use the training and test video datasets of LifeCLEF Fish task in ImageCLEF [1]. These video datasets are derived from the Fish4Knowledge [12] video repository, which contains about 700k 10-minute video clips that were taken in the past five years to monitor Taiwan coral reefs. Each video has a resolution of either 320×240 or 640×480 with 5 to 8 fps. The videos recorded from sunrise to sunset show several phenomena, e.g. murky water, algae on camera lens, etc., which makes the fish detection task more complex and makes the datasets suitable for proving the robustness and practice of our system. However, because the fish categories of these video datasets are not balanced in quantity and the ground truth is noisy in coordinate, we must manually filter proper fish frames out from the videos for training and testing.

A. Save video frames as images

.We collected 24872 fish images with 320×240 resolution or 640×480 resolution. These images are composed of 18 fish categories. And part of the images are in poor quality, for example, there can be some fish whose color is similar to that of background and there can be some occlusion among fish and background. We retained these inferior images because it is usual to capture unsatisfying frames on account of underwater environment and equipment fault and we aim to train a strong and robust network for complicated oceanic environment.

B. Balance the new fish dataset

As mentioned above, the 18 fish categories are not balanced in quantity, and we ignored the fish classes whose occurrence times are less than 600. The final dataset includes 24277 images belong to 12 fish classes as shown in table 1 and figure 2. We set nearly equal proportion for training set, validation set and testing set.

C. Modify the ground truth of fish

We transformed the annotations for every video of ImageCLEF into the annotations for every image of our new dataset. There are some noisy annotations in fish classes and bounding box values, for example, same fish class may have two different spellings and some bounding box values may be negative or out of resolution range. we manually modify these improper annotations to apply Fast R-CNN to the new fish dataset we named ImageCLEF_Fish_TS here.

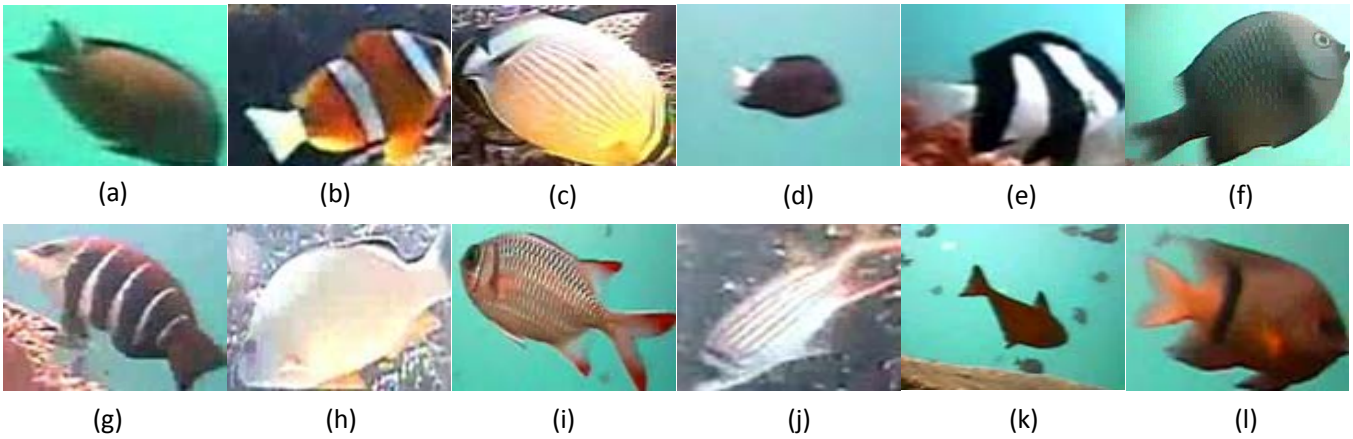

Figure 2: 12 fish species in ImageCLEF_Fish_TS dataset. (a) Acanthurus Nigrofuscus. (b) Amphiprion Clarkii. (c) Chaetodon Lunulatus. (d) Chromis Chrysura. (e) Dascyllus Aruanus. (f) Dascyllus Reticulatus. (g) Hemigymnus Fasciatus. (h) Lutjanus Fulvus. (i) Myripristis Kuntee. (j) Neoniphon Sammara. (k) Pempheris Vanicolensis. (l) Plectrogly-Phidodon Dickii.

III. Overall Architecture

As depicted in Figure 1, the networks take as input a RGB image and its 2000 regions of interest (RoIs) collected by selective search and produces a distribution over fish classes as well as related bounding-box values. The networks contain five convolutional layers, a RoI pooling layer, two fully connected layers and two sibling layers (a fully connected layer and softmax layer over 12 fish classes plus background class and bounding-box regressors). Necessary response normalization and max pooling layers follow the first and second convolutional layers. The ReLU non-linearity is applied to the output of every convolutional layer and every fully connected layer.

A. Pre-training and modifying

We pre-trained an AlexNet [3] with five convolutional layers and three fully connected layers on a large auxiliary dataset (ILSVRC2012), using the open source Caffe CNN library [13]. Before initializing Fast R-CNN with the pre-trained networks, we modified the AlexNet from three respects.

- The networks take two data inputs: a batch of N images and a list of R RoIs. We use selective search [11] to generate about 2000 category-independent region proposals and refer to these candidates as regions of interest (RoIs).
- The last max pooling layer is replaced by a RoI pooling layer. The RoI pooling layer pools RoIs of arbitrary sizes into fixed-size feature maps, which is similar to the spatial pyramid pooling layer used in SPPnet [8].
- The final fully connected layer is replaced with two sibling layers, one that outputs softmax probabilities over 12 fish classes plus a "background" class and another layer that outputs related bounding-box values of the 12 fish classes.

B. Fine-tuning for fish detection

To adapt Fast R-CNN to new specific domain (fish detection and recognition), we modify AlexNet as mentioned above and use stochastic gradient descent (SGD) training the Fast R-CNN parameters.

Multi-task loss. We use a multi-task loss L to train networks jointly for softmax classification and bounding-box regression:

$$L(p, k^*, t, t^*) = L_{cls}(p, k^*) + \lambda[k^* \geq 1]L_{loc}(t, t^*) \quad (1)$$

in which, $p = (p_0, \ldots, p_K)$ is a discrete probability distribution (per RoI) over $K+1$ categories, k^* is a true class label, and $L_{cls}(p, k^*) = -\log p_{k^*}$ is the standard cross-entropy/log loss. The second loss L_{loc} is defined over a tuple of true bounding-box regression targets $t^* = (t_x^*, t_y^*, t_w^*, t_h^*)$ for class k^* and a predicted tuple $t = (t_x, t_y, t_w, t_h)$ for class k^*:

$$L_{loc}(t, t^*) = \sum_{i \in \{x,y,w,h\}} smooth_{L_1}(t_i, t_i^*) \quad (2)$$

in which

$$smooth_{L_1}(x) = \begin{cases} 0.5x^2 & if\ |x| < 1 \\ |x| - 0.5 & otherwise \end{cases} \quad (3)$$

The hyper-parameter λ controls the balance between the two task losses and we used $\lambda = 1$ as [9].

Mini-batch sampling. we define object proposals that have intersection over union (IoU) overlap with ground truth bounding box of at least 0.5 as positive and define object proposals that have a maximum IoU with ground truth in the interval [0,0.3] as negative (background). We uniformly sample 32 positive RoIs and 96 negative RoIs from 2 images to construct a mini-batch of size 128. We ignore horizontally flipped trick adopted in [9] and other data augmentation, because our fish dataset is extracted from underwater videos and there is little difference between adjacent frames.

method	DPM	DPM(BP)	RCNN	RCNN(BB)	FRCN	FRCN(SVD)
Acanthurus Nigrofuscus	69.2	72.3	84.2	85.0	**87.4**	86.6
Amphiprion Clarkii	52.4	62.2	77.7	**88.0**	82.2	71.4
Chaetodon Lunulatus	79.6	84.5	86.0	**87.8**	85.5	82.4
Chromis Chrysura	86.6	87.2	87.7	88.0	**88.6**	**88.6**
Dascyllus Aruanus	68.1	68.8	74.9	78.8	**79.0**	72.9
Dascyllus Reticulatus	52.7	53.2	57.8	58.8	**72.0**	70.0
Hemigymnus Fasciatus	76.0	80.8	79.2	**80.7**	79.5	78.7
Lutjanus Fulvus	79.3	80.8	90.3	**90.9**	89.2	89.5
Myripristis Kuntee	75.2	78.0	88.1	**88.7**	84.2	83.2
Neoniphon Sammara	76.1	79.8	86.9	**90.6**	89.2	87.6
Pempheris Vanicolensis	34.6	31.4	61.3	63.7	**65.4**	63.7
Plectrogly-Phidodon Dickii	60.9	63.5	71.2	72.8	**74.6**	72.4
mAP	67.6	70.2	78.8	81.2	**81.4**	78.9

Table 2: Fish detection average precision (%). Here DPM (BP) means the DPM detection using a bounding-box prediction and RCNN (BB) means the RCNN detection using a bounding-box regression. FRCN (SVD) means the Fast R-CNN with singular value decomposition.

Training details. We trained our networks using stochastic gradient descent with a batch size of 128 RoIs, momentum of 0.9, and weight decay of 0.0005. The fully connected layers for softmax classification and bounding-box regression are initialized randomly from a zero-mean Gaussian distribution with standard deviations 0.01 and 0.001 following [9].We uses a global learning rate of 0.0001 for 30k mini-batch iterations and lowered the learning rate three times for every 10k iterations by ten. The update rule for weight w is

$$v_{i+1} = 0.9v_i - 0.0005\alpha \cdot w_i - \alpha \cdot \left\langle \frac{\partial L}{\partial w} \middle| w_i \right\rangle_{B_i} \quad (4)$$

$$w_{i+1} = w_i + v_{i+1} \quad (5)$$

in which i is the iteration index, v is the momentum variable, α is the learning rate and B_i is the ith mini-batch.

Truncated Singular Value Decomposition. We use truncated Singular Value Decomposition (SVD) to compress fully connected layers [14] because the computation time spent on fully connected layers is nearly half of the forward pass time, following [9]. We want this simple compression method which does not need additional fine-tuning to obtain further speedup for fish detection and recognition.

IV. Experiments and Analysis

We apply our fish detection and recognition system based on Fast R-CNN to ImageCLEF_Fish_TS dataset, and compare our method with two recent detection approaches that build on Deformable Part Models (DPM) and Regions with CNNs (R-CNN) respectively. HOG-based DPM is defined by filters that score subwindows of a feature pyramid after doing principal component analysis on HOG features [15]. HOG-based DPM has achieved state-of-the-art precision on VOC dataset before deep learning widely applied to object detection, and we refer to DPM as our baseline method. Similar to Fast R-CNN, R-CNN detects image objects combined region proposals with CNNs. But unlike single-stage Fast R-CNN, R-CNN is a piecewise method comprising of proposals generation, feature extraction and SVM classification [7].

A. Mean Average Precision (mAP)

We use mean Average Precision (mAP) to evaluate three detection methods mentioned above on the new fish dataset ImageCLEF_Fish_TS. Mean Average Precision (mAP) is define by recall (R) and precision (P) as follows

$$\text{mAP} = \int_0^1 P(R)dR \quad (6)$$

Table 2 shows the detection mAP of three approaches on the new dataset. Our fish detection and recognition method achieves an mAP of 81.4%, improving mAP by 11.2% compared to DPM baseline, slightly better than R-CNN with bounding-box regression. Compared to systems based on traditional HOG-like features, deep ConvNets can learn high-level representations that combines class-tuned features together with shape, texture, color and material properties, which leads to dramatically higher fish detection performance on ImageCLEF_Fish_TS.

B. Training and testing time

Along with high accuracy and powerful capacity, deep ConvNets are faced with large time and space cost. Table 3 compares training time (hours), testing time (seconds per image), and mAP on ImageCLEF_Fish_TS between R-CNN and Fast R-CNN.

On a computer equipped with an NVIDIA Tesla K20 GPU, Fast R-CNN tests *80* times faster than R-CNN per image, for the fact that Fast R-CNN takes 0.311s while R-CNN takes 24.945s on the same image. Moreover, Fast R-CNN trains more than 16 times faster than R-CNN, as Fast R-CNN takes 4 hours while R-CNN takes 67 hours on training. It is notable that further testing speedup is achieved by truncated SVD that tests 91 times faster than R-CNN on the same image despite slightly mAP decline.

The reason for the outstanding speedup is that the single-stage Fast R-CNN uses a RoI pooling layer to help extract RoI feature vectors, and the existence of RoI pooling layer cuts down lots of feature extraction time and space cost in multi-stage R-CNN. Therefore, faced with messes of underwater images, an automatic fish identification system with Fast R-CNN is tend to be more practical.

method	R-CNN	FRCN	FRCN(SVD)
training time (h)	67	4	-
training speedup	1×	16.75×	-
testing time (s/im)	24.945	0.311	0.273
testing speedup	1×	80.2×	91.4×
mAP (%)	81.2	81.4	78.9

Table 3: Runtime comparison between Fast R-CNN and R-CNN. Here FRCN (SVD) means the Fast R-CNN with singular value decomposition.

V. CONCLUSIONS

Encouraged by the outstanding detection precision and speed property achieved by Fast R-CNN, we apply this promising networks to automatic fish identification system to help marine biologists estimate fish existence and quantity, and effectively understand oceanic geographical and biological environments. The experimental results demonstrate the promising performance of our detection system with a higher mAP than DPM and definitely speedup to R-CNN. Faced with large amounts of underwater images and videos collected day by day, further study for deep ConvNets with high precision and computation speed will be explored in this filed.

REFERENCES

[1] C. Spampinato, R B. Fisher, and B. Boom. Image Retrieval in CLET-Fish task. http://www.imageclef.org/2014/lifeclef/fish, 2014.

[2] R. Mehdi, et al. Automated Fish Detection in Underwater Images Using Shape‐Based Level Sets. The Photogrammetric Record 30.149 (2015): 46-62.

[3] A. Krizhevsky, I. Sutskever, and G. Hinton. ImageNet classification with deep convolutional neural networks. In NIPS, 2012.

[4] J. Deng, A. Berg, S. Satheesh, H. Su, A. Khosla, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Competition 2012 (ILSVRC2012). http://www.image-net.org/challenges/LSVRC/2012/.

[5] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A large-scale hierarchical image database. In CVPR, 2009.

[6] H. Qin, Y. Peng, and X. L. Foreground Extraction of Underwater Videos via Sparse and Low-Rank Matrix Decomposition. Computer Vision for Analysis of Underwater Imagery (CVAUI), 2014 ICPR Workshop on. IEEE, 2014.

[7] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In CVPR, 2014.

[8] K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In ECCV, 2014.

[9] R. Girshick. Fast R-CNN. arXiv preprint arXiv:1504.08083 ,2015.

[10] C. Spampinato, Y-H. Chen-Burger, G. Nadarajan, R. Fisher. Detecting, Tracking and Counting Fish in Low Quality Unconstrained Underwater Videos. In Proc. 3rd Int. Conf. on Computer Vision Theory and Applications (VISAPP). Vol. 2. 2008. p. 514-519.

[11] J. Uijlings, K. van de Sande, T. Gevers, and A. Smeulders. Selective search for object recognition. IJCV, 2013.

[12] Fish4Knowledge. www.fish4knowledge.eu

[13] Y. Jia. Caffe: An open source convolutional architecture for fast feature embedding. http://caffe.berkeleyvision.org/, 2013.

[14] E. Denton, W. Zaremba, J. Bruna, Y. LeCun, and R. Fergus. Exploiting linear structure within convolutional networks for efficient evaluation. In NIPS, 2014.

[15] R. Girshick, P. Felzenszwalb, and D. McAllester. Discriminatively trained deformable part models, release 5. http://www.cs.berkeley.edu/~rbg/latent-v5/.

[16] M. Everingham, L. Van Gool, C. K. I.Williams, J.Winn, and A. Zisserman.The PASCAL Visual Object Classes (VOC) Challenge. IJCV, 2010.