Vision based Real-time Fish Detection Using Convolutional Neural Network

Minsung Sung and Son-Cheol Yu
Department of Creative IT Engineering
Pohang University of Science and Technology
(POSTECH)
Pohang, Republic of Korea
kyjor23@postech.edu and sncyu@postech.ac.kr

Yogesh Girdhar
Applied Ocean Physics & Engineering
Woods Hole Oceanographic Institution (WHOI)
Massachusetts, United States
yogi@whoi.edu

***Abstract*— Underwater vision has specific characteristics such as high attenuation of lights, severe noise and haze in the images. For real-time fish detection using underwater vision, this paper proposes convolutional neural network based techniques based on You Only Look Once algorithm. Actual fish video images were used to evaluate the reliability and accuracy of the proposed method. As a result, the network recorded 93% classification accuracy, 0.634 intersection over union between predicted bounding box and ground truth, and 16.7 frames per second of fish detection. It also outperforms another fish detector using sliding window algorithm and classifier trained with histogram of oriented gradient features and support vector machine.**

Keywords—fish detection; convolutional neural network; object detection; you only look once; yolo;

I. INTRODUCTION

Marine ecosystem and fisheries resources are important. Many researchers are observing fish habitat or fish species changes to protect and utilize marine resources [1]. For example, changes in species in certain areas can be used as an indicator of climate change.

Underwater optical vision provides most enrich information in underwater sensing. The vision automatic fish detection could provide important information to understand the marine echo system.

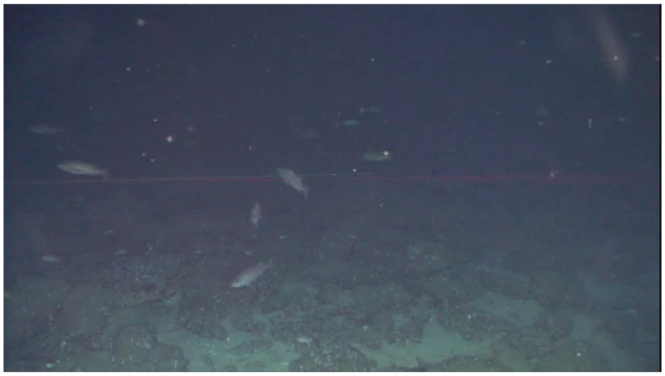


Fig. 1 Example of Underwater image that shows low light, high noise, and haze effect.

However, there are many difficulties in underwater image recognition due to environmental characteristics. Because water attenuates light sharply [2], underwater images have less illumination. It also causes high noise. In addition, underwater images are hazy that occurs when light is absorbed or scattered many times by floating matter in the water. These problems make it difficult to apply existing object detection algorithms such as Histogram of Oriented Gradients (HOG) algorithm and Scale-Invariant Feature Transform (SIFT) algorithm. Fig. 1 shows the representative examples; dim light, high noise, and haze effect. These difficulties should be overcame to detect the fishes in optical vision.

Currently, many underwater images are obtained by Autonomous Underwater Vehicles (AUVs) due to their excellent mobility. If the fish detection method could be implement to AUVs, they could survey the fish related data significantly. For the implementation, the detection method should be highly efficient, because the AUVs' computing power is strictly restricted.

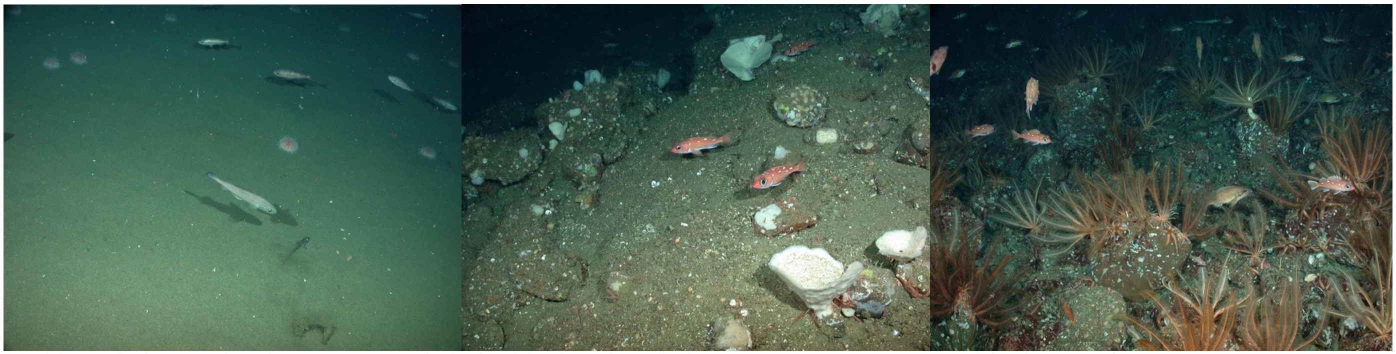
In section II, we dealt with previous methods to fish in the water. In section III, we discussed the neural network that we built, image set that we used, and way to train built neural network. In section IV, we provided result of the convolutional neural network (CNN). To validate the quality of algorithm, we measured three metrics; classification accuracy, intersection over union (IOU) processing frames per second (FPS). Then, we compared our fish detector with another detector using HOG algorithm.

II. RELATED WORKS

Many researches have proposed methods to detect fish in underwater. They count echoes of sonar beams [3]. It is easy to implement and widely used. However, these methods have limitations due to the problem of high noise and missing echoes. The quality has been improved by helping to analyze the contour of the fish in the sonar image [4]. Furthermore, methods using only acoustic images have been developed [5].

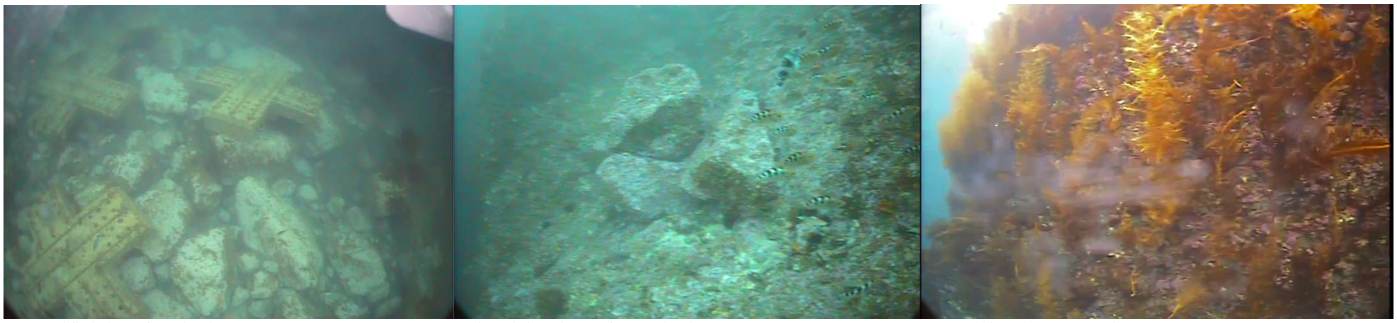
Another approach to detecting fish is to use optical images instead of acoustic images. Shrivakshan *et al.* [6] developed a shark classifier using various types of edge detection algorithm such as Canny edge detector, Prewitt filter and so on [6]. However, they used segmented shark images rather

978-1-5090-5278-3/17/$31.00 ©2017 IEEE



(a) Fishes and other organisms in monotonous seabed (b) Fishes in rocky seabed (c) Fishes in seaweed

Fig. 2 Samples of images from NOAA dataset.



(a) Artificial reef seabed (b) Fishes in rocky seabed (c) Fishes in seaweed

Fig. 3 Samples of images taken on Ullungdo Island

than underwater shark images. Spampinato *et al.* [7] developed a machine vision system that detects, tracks and counts fish in underwater videos. In their system, they used the Adaptive Gaussian Mixture Model [8] for detection and the Continuously Adaptive Mean Shift Algorithm [9] for tracking.

Some researchers have introduced neural networks to detect fish in images. Ramani, *et al.* [10] detected and identified four species of fish in sonar images using parallel networks of three perceptron layers. Storbeck, *et al.* [11] proposed a method of classifying moving fish on a conveyor belt using image processing algorithms and three-layer CNN. Marburg et al. [12] detected and identified ten classes of benthic macrofauna in optical underwater images using series of convolutional neural networks.

Some methods have shown real-time performance in detecting underwater objects using learning or neural networks. Byeongjin *et al.* [13] use Haar-like features and AdaBoost algorithm to detect underwater objects in real-time. Juhwan *et al.* [14] developed a method to use CNN to detect and track small ROV in subsequent sonar images. We also adopted CNN to detect fish in optical underwater images in real-time.

III. Proposed Method

As a result, we proposed an algorithm using CNN to detect fish in real-time

A. *Implementation of YOLO Network*

The classical approach for object detection involves matching of low-level features, such as a fixed shape of contour and a pattern of color. However, this method did not show good performance in underwater images due to low illumination, high noise, and haze of image. A CNN, on the other hand, learns high-level features as well as the classifier that uses these features at the same time, using a multiple layer architecture. The learned high-level features in this case are specific to the training data, and hence likely to perform better than generic low-level features, as long as the test.

In recent years, many object detection algorithms using CNN have been developed. Two main purposes of research are improving accuracy and processing speed. Several object detection algorithms have been developed that focus on speed improvement for real-time processing. Among these algorithms, we propose the use of You Only Look Once (YOLO) introduced by Redmon Joseph, *et al.* [15], for real-time fish detection task.

In contrast to other CNN architectures that detect object by sliding trained classifier, YOLO is unified network that simultaneously predicts the position, size and the class probability of object. Therefore, YOLO shows fast performance. We adopted the architecture of YOLO and implemented a CNN of 24 convolutional layers and two fully connected layers. We then trained the network using our custom dataset to detect fish in underwater images.

B. *Specificity Enhancement*

To improve the accuracy of detection, we assumed that separating the fish from the seabed was the essential because usually fish have similar color with seabed to protect themselves. We also assumed that the fish images of various seabed were necessary for this. Therefore, we performed field experiment on Ullungdo Island, Republic of Korea, and collected optical underwater images of various seabed. Then, we randomly cropped the images of various seabed and labeled them as ‘negative’. As a result, we reduced the rate of misclassification that seabed is classified as fish.

In addition, to prevent the non-fish objects from being classified as fish, we manually cropped invertebrates and float in the images manually and then recorded the x, y coordinates, width and height of invertebrates and float. We then labeled the cropped images as ‘negative’ class and trained that the network did not classify them as fish.

C. Hyper-parameters Optimization

In order to train network efficiently and well to predict the desired outcome, hyper-parameters of the network should be properly determined. For the value of momentum, number of epoch, mini batch size, and learning rate, the grid search was performed to optimize the hyper-parameters. All possible set of values described in Table I was tested to train the network. Then, after training the network using each set of values, the sensitivity was measured using 100 test images. Then, the values that maximize the quality of network were adopted for hyper-parameters.

IV. Experiments

A. Training using Custom Dataset

We used the labeled fishes in the wild image dataset provided by National Oceanic and Atmospheric Administration (NOAA) Fisheries [16]. It consists of 929 fish images and its annotation. For training and testing the CNN, we divided them into two groups; 829 images for training and 100 images for testing. Fig. 2 is examples of the images we used to train.

In addition, we gathered additional underwater image from field experiment on Ullungdo Island, Republic of Korea. Images are taken by small ROV and consist of 983 underwater images regardless of whether it contains fish or not. All of these images taken on Ullungdo Island are used for training the network. Fig. 3 is examples of images taken on Ullungdo Island.

In order to train the network to detect fish in a given image, it is necessary to provide not only the class of the object but also the bounding box data of the object as correct answer. To do this, we parsed the x, y coordinates, width and height of the fish in the annotation of the images of NOAA dataset. In addition, we manually cropped the fish and recorded bounding boxes for images taken on Ullungdo Island.

TABLE I. TESTED VALUES FOR GRID SEARCH OF HYPER-PARAMETERS

Momentum	0.5, 0.6, 0.7, 0.8, 0.9, 1.0
Number of epoch	1,000, 2,000, …, 40,000
Mini batch size	1, 2, 4, 8, 16, 32, 64
Learning rate	0.0001, 0.0002, 0.0003, 0.0004, 0.0005

TABLE II. HYPER-PARAMETERS TO TRAIN

Momentum	**Number of epoch**	**Mini batch size**	**Learning rate**
0.9	20,000	32	0.0005

B. Hyper-parameters optimization

We performed a grid search to optimize the hyper-parameters of our network. The network shows better performance as number of epoch increases. However, when number of epoch exceeds 20,000, the network seem to be overfitted to training data. In addition, when mini batch size is bigger, the network shows better performance. However, we cannot raise the number blindly due to limitation of memory. As a result, Table II shows the hyper-parameters used to train and test the CNN.

V. Result

A. Training result

Fig. 4 is a graph showing relationship between training epoch and average loss. For each epoch, 32 images are randomly selected and used to train neural network. Because the number of samples is limited, each image is used multiple times. The graph shows that the average loss is reduced to 2.5% as training epoch progresses. That means training data affects the CNN model. A total of 20,000 epochs were run and it took 12 hours to complete the training.

B. Object detection result

After training the neural network, we tested the network with 100 fish images of NOAA dataset. The neural network detects fish in given images and displays bounding boxes around the detected fishes. Fig. 5 shows result of fish detection in still images. We used three metrics to measure the quality of the network.

First, we measured the accuracy of the classification to see how well the neural network detects the fish. We have prepared 100 images containing fish and labeled ‘positive’. In addition, we prepared 100 images that did not contain fish and labeled ‘negative’. We then checked that the network detected the fish in the positive image and did not detect the fish in the negative image. AS a result, sensitivity was 93% and specificity was 62%.

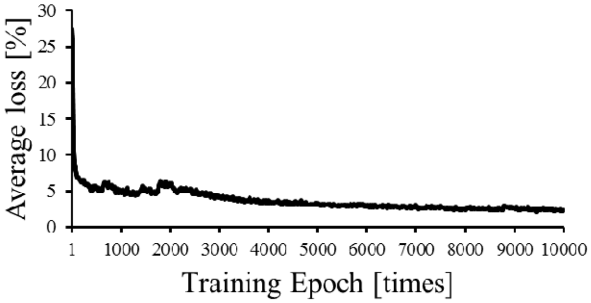
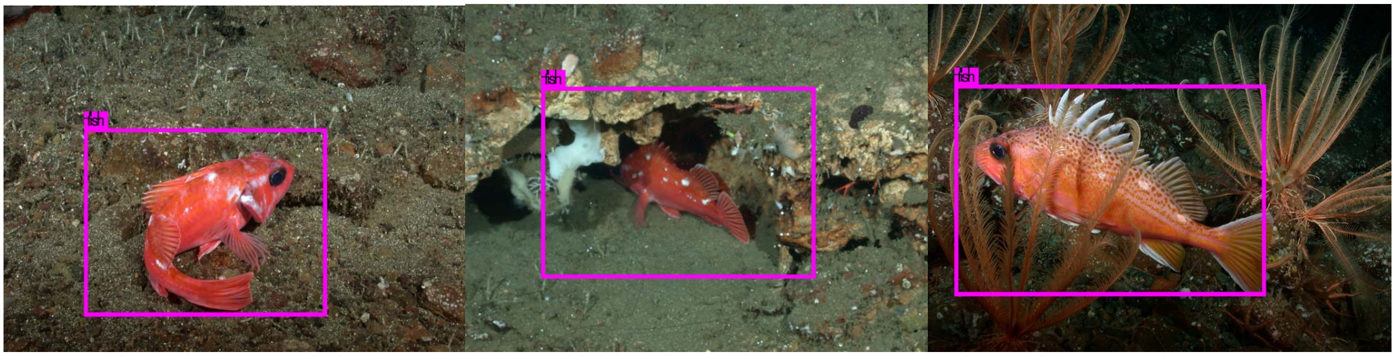
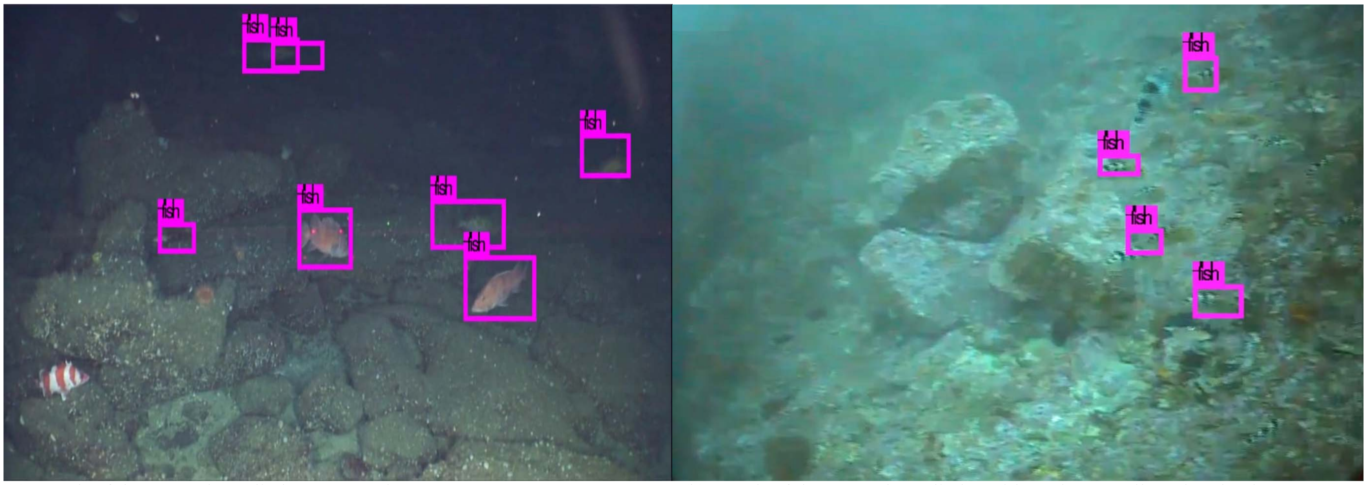


Fig. 4 The average loss function of training



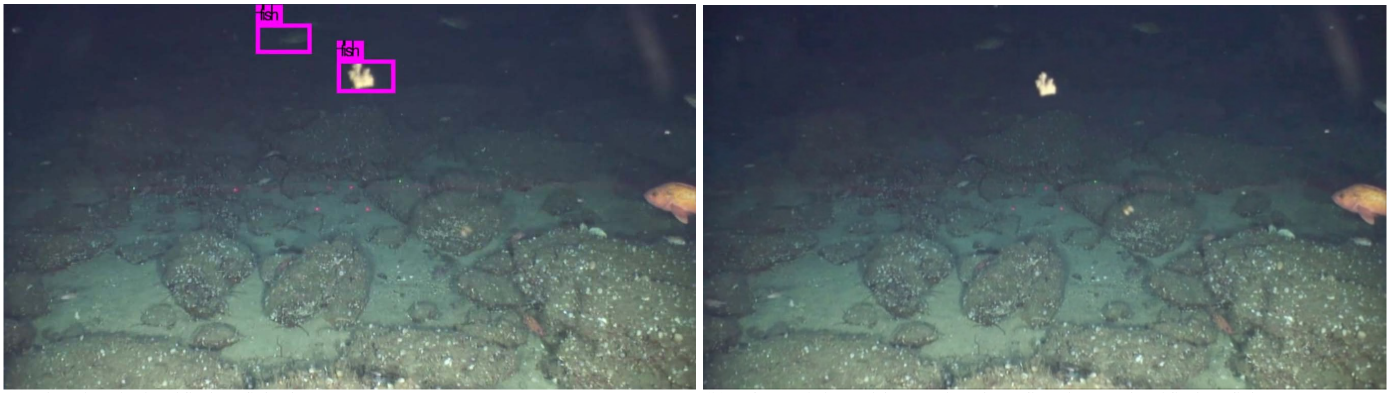
(a) Detected fish in monotonous seabed. (b) Detected fish in rocky seabed (c) detected fish in seaweeds

Fig. 5 Result of the network on still images



(a) Result of the network applied to NOAA dataset (b) Result of the network applied to video taken to Ullungdo Island

Fig 6. Result of real-time fish detection in video



(a) Float is misclassified as fish class (b) After training with negative data, float is not classified as fish no more.

Fig. 7 Effect of training network using negative labeled data

The second metric is IOU to determine how accurately the neural network predicted the bounding box. We calculated the IOU between the ground truth and the predicted bounding box for the 93 images where the fish were detected. The average value for 93 images was 65.3%.

Finally, we measured the FPS to verify the processing speed. As a result, it recorded 16.7 FPS on a GeForce Pascal Titan Graphics Processing Unit (GPU). We have confirmed that the network can simultaneously detect fish in real-time while the video is playing. Fig. 6 (a) shows the result of applying fish detection to the video in the NOAA dataset. We also applied the network to video taken on Ulleungdo Island, and Fig. 6 (b) shows the result. Table III summarizes the result for three metrics that explained above.

C. *Effect of specificity enhancement.*

Fig. 8 shows how some negative class images could improve specificity. Fig. 7 (a) shows the result of the network trained using only fish images. Fig. 7 (b) is the result of the network trained by negative labeled images of manually-cropped seabed, invertebrates and floats. As a result, the float misclassified in Fig 7 (a) shows that it is no longer detected as a fish in Fig. 7 (b).

TABLE III. OBJECT DETECTION RESULT

Classification Accuracy		IOU	FPS
Sensitivity	*Specificity*		
93%	62%	65.2%	16.7

D. Comparison to another method

We compared our network to another fish detector. We implemented a fish classifier using the HOG features of the images and trained the classifier using Support Vector Machin (SVM). The sliding window algorithm then moved the classifier and detected the position and size of fish in given images.

We applied both HOG algorithm and SVM-based detectors and our network to 100 still images in the NOAA dataset. Fig. 8 and 9 are the Precision-Recall Curve and Receiver Operating Characteristics (ROC) curve, which compare the two methods. Table IV also shows comparative performance between our networks and method using sliding window. In both detection accuracy and processing speed, it shows that our network outperforms the method using HOG algorithm and SVM.

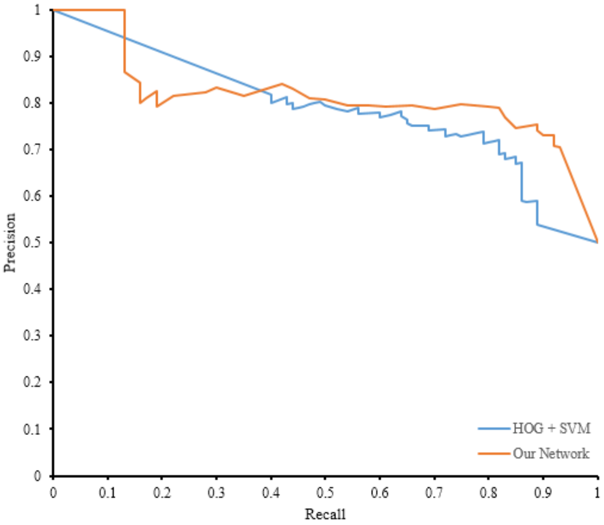


Fig. 8 Precision-Recall Curve that compares two methods.

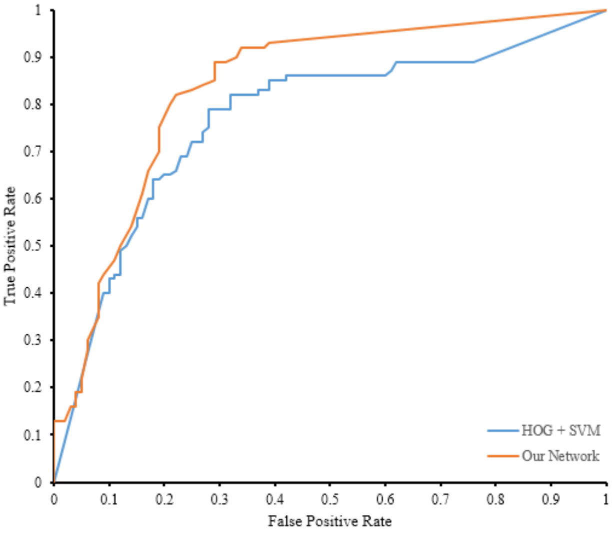


Fig. 9 ROC Curve that compares two methods.

TABLE IV. COMPARISON BETWEEN OUR NETWORK AND SLIDING WINDOW METHOD ABOUT FRAMES PER SECOND

Our Network	Sliding Window
16.7 fps	0.00759 fps

VI. CONCLUSION

In this paper, the CNN based real-time fish detection method was proposed. We adopted architecture of YOLO for real-time detection, and trained the network using custom fish images. In addition, we trained the network using non-fish organisms and various type of seabed to enhance specificity of the network. As a result, we could detect fish precisely in real-time.

When compared with HOG classifier-based algorithms, the proposed method outperforms the HOG classifier and shows much faster processing speed. It shows that neural network can successfully process noisy, dim-light, and hazy underwater image.

To detect target object using neural network, the network is trained using various images of target object. For fish detection, because fish have protective coloration of their body, there are many cases that seabed was misclassified as fish. Therefore, training the network using randomly cropped seabed images was helpful to enhance the accuracy.

The network can be used to investigate marine ecosystem and fishery resources. Therefore, we plan to conduct field test to detect and track fish using hovering-type AUV equipped with the network.

In addition, our network still classifies all fish as ‘positive’ class regardless of species. However, our network uses optical images rather than soar images, so if we collect more images by species, we would be able to detect, observe, and track fish for certain species.

ACKNOWLEDGMENT

This research was supported by the MSIP(Ministry of Science, ICT and Future Planning), Korea, under the “ICT Consilience Creative Program”(IITP-R0346-16-1007) supervised by the IITP(Institute for Information & communications Technology Promotion) and the project titled “Gyeongbuk Sea Grant Program” funded by the Ministry of Oceans and Fisheries, Korea and Civil Military Technology Cooperation Center, Korea.

This work and Y. Girdhar was also supported by Independent Research & Development Program at WHOI

REFERENCES

[1] Angermeier, Paul L., and James R. Karr. "Relationships between woody debris and fish habitat in a small warmwater stream." Transactions of the American Fisheries society 113.6, 1984, pp.716-726.

[2] Gordon, Howard R. "Can the Lambert□Beer law be applied to the diffuse attenuation coefficient of ocean water?." Limnology and Oceanography 34.8, 1989, pp.1389-1409.

[3] Xie, Yunbo, George Cronkite, and Timothy James Mulligan. "split-beam echosounder perspective on migratory salmon in the Fraser River." (1997)

[4] Balk, Helge, and Torfinn Lindem. "Improved fish detection in data from split-beam sonar." Aquatic Living Resources 13.5 (2000): 297-303.

[5] Holmes, John A., et al. "Accuracy and precision of fish-count data from a “dual-frequency identification sonar”(DIDSON) imaging system." ICES Journal of Marine Science: Journal du Conseil 63.3 (2006): 543-555.

[6] Shrivakshan, G. T., and C. Chandrasekar. "A comparison of various edge detection techniques used in image processing." IJCSI International Journal of Computer Science Issues 9.5 (2012): 272-276.

[7] Spampinato, Concetto, et al. "Detecting, Tracking and Counting Fish in Low Quality Unconstrained Underwater Videos." VISAPP (2) 2008 (2008): 514-519.

[8] Zivkovic, Zoran. "Improved adaptive Gaussian mixture model for background subtraction." Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on. Vol. 2. IEEE, 2004.

[9] Fukunaga, Keinosuke. "Introduction to statistical pattern recognition.", Academic Press, NY, 1990.

[10] Ramani, Narayan, and Paul H. Patrick. "Fish detection and identification using neural networks-some laboratory results." IEEE journal of oceanic engineering 17.4 (1992): 364-368.

[11] Storbeck, Frank, and Berent Daan. "Fish species recognition using computer vision and a neural network." *Fisheries Research* 51.1 (2001): 11-15.

[12] Kim, Byeongjin, and Son-Cheol Yu. "Imaging sonar based real-time underwater object detection utilizing AdaBoost method." Underwater Technology (UT), 2017 IEEE. IEEE, 2017.

[13] Kim, Juhwan, and Son-Cheol Yu. "Convolutional neural network-based real-time ROV detection using forward-looking sonar image." Autonomous Underwater Vehicles (AUV), 2016 IEEE/OES. IEEE, 2016.

[14] Marburg, Aaron, and Katie Bigham. "Deep learning for benthic fauna identification." OCEANS 2016 MTS/IEEE Monterey. IEEE, 2016.

[15] Redmon, Joseph, et al. "You only look once: Unified, real-time object detection." arXiv preprint arXiv:1506.02640, 2015.

[16] Cutter, George, Kevin Stierhoff, and Jiaming Zeng, “Automated detection of rockfish in unconstrained underwater videos using Haar cascades and a new image dataset: labeled fishes in the wild,” Applications and Computer Vision Workshops (WACVW), 2015 IEEE Winter, IEEE, 2015.