WaterDetect: Using Monocular Camera to Detect and Classify Underwater Marine Animals

Tushar Aggarwal and Xinyu Dong
Department of MIE, University of Toronto, Canada
tushar.aggarwal@mail.utoronto.ca, xinyu.dong@mail.utoronto.ca

***Abstract*—This paper describes a detection and classification method used in underwater environments for marine animals using a monocular camera. Our method, called WaterDetect, consists of two stages which are image processing stage, and detection and classification stage, respectively. The image processing stage is based on the existing Fusion method [1]. It firstly takes underwater images as inputs and enhances color and photographic sharpness of original images. Then, the second stage takes processed images and feeds them into a YOLOv4 classifier to detect and classify marine animals. In the detection stage, we train the model using different hypotheses to pick the best parameters for training the YOLOv4 detector. The enhanced images of Brackish Dataset [2] were used for training and comparison for the hypothesis. Based on our evaluation, if YOLOv4 [3] is trained with higher number of images and for longer iteration, it improves the detection performance and therefore the final selected detector for stage 2 is the one trained on 800 images for 6000 iterations. The final pipeline of our Fusion-YOLO model can successfully detect marine animals in low visibility brackish water conditions to detect different marine life.**

I. INTRODUCTION

The underwater environment is one of the most challenging conditions for object detection. Different from capturing images on the ground, underwater perception is limited due to the higher device costs, the complexity of setup and especially the interferences of water medium on signal transmissions. In particular, the light received by cameras in underwater environment can be significantly attenuated by the water medium due to the absorption and scattering effects. Water absorbs warmer colors with their longer wavelengths, such as red and orange, while scattering the cooler colors that have shorter wavelengths. As a result, light intensity is weakened, and the true colors are faded. This significantly degrades the performance of object detection on underwater environment operations. In fact, the object at a distance of 10 m in underwater environments is almost indistinguishable. However, with the development of technology, human beings become not only satisfied with exploring the world on ground, they also have a growing interest in underwater creatures. Driven by this interest, underwater operation is getting more and more required either in industry or academia, such as fish farming and marine life research. Therefore, there is a strong motivation to investigate the robust detection and classification methods used in the underwater environment for marine animals.

Common object detection methods can be roughly classified into two categories, which are vision-based and lidar-based. Lidar which stands for light detection and ranging is a remote sensing method using pulsed light. It emits pulsed laser light and obtains depth information by measuring the time of flight of emissive light and its reflection. This method is frequently used in autonomous driving scenarios due to its capability to acquire accurate depth information of the surrounding environment. However, lidar-based methods are not useful in the underwater environment since its working mechanism is using laser light which can be significantly attenuated by the water medium. Moreover, Lidar also has the disadvantage of high cost and sparse information compared to vision-based methods. In contrast to Lidar-based methods, vision-based methods do not obtain or directly obtain the depth information, instead, it captures the high-resolution images and feed into neural networks to detect and classify objects.

With this paper, we strive to accommodate the need of investigating on underwater creatures by proposing a monocular camera based marine animal detection and classification method called WaterDetect. In this method, we utilize the advantages of monocular cameras such as cheap, easy to set up and extensive number of related datasets which ease our preparation on training models. The network structure of our method is shown in figure 1 which can be divided into two stages which are the image processing stage, and the detection and classification stage. The image processing stage is included in order to compensate for the degradation of images by water medium, it is built on the basis of an existing Fusion method [1] published on IEEE in 2012. It takes original underwater images and outputs images with considerable improvements on colors and photographic sharpness. The second stage, detection and classification stage, contains a classifier which is trained on YOLOv4 using Brackish dataset. This stage takes the processed images from stage 1 then detects and classifies the observed marine animals using the classifier.

The crucial component that ensures the success of this method is the dataset used. A relatively large amount of training data is indispensable to state-of-the-art detection and classification methods. The dataset we use is the Brackish dataset [2], it trains our classifier and sets the benchmark for the evaluation. This dataset includes five clusters of the most commonly seen marine animals which are fishes, shrimps, jellyfishes, starfishes and crabs.

The method was evaluated on picked images from brackish dataset. The result shows our method of training enhanced images achieved better mean average precision and better detection compared to training on unenhanced brackish dataset.

In this paper, we examined the potential of monocular camera on object detection in underwater environment. Our main contribution is the validation of the advantages of image enhancement on marine animals' detection and classification.

This paper is organized as following: section 2 is related work, section 3 is methodology, section 4 is implementation of our YOLOv4 results and section 5 is conclusion.

II. Related Work

This section briefly introduces the common object detection and classification methods and datasets of marine animals, respectively.

A. Detection and Classification Methods

Not like overland situations, vision-based methods are more widely used than lidar-based methods since the scattering effect of water medium significantly interferences the transmission of pulsed laser light. Many vision-based algorithms are developed based on Fast R-CNN (regions with Convolutional Neural and Networks) with some modifications such as [16, 17]. Some algorithms are also designed using supervised machine learning. For instance, [18]. This one proposes a classifier use Deep Neural Networks to create feature vectors and classify them.

Sonar, which stands for sound detection and ranging, is a great substitute to Lidar in underwater environment. Different from Lidar, Sonar utilizes sound waves to detect surrounding objects instead of using pulsed laser light, it benefits from the nature of sound waves which can effectively transmit in water medium without much attenuation. When Sonar is used with multiple receivers, it is possible to precisely locate fishes as performed in the experiments [19, 20], but not able to classify them. Even though it is possible to get a 3D image of some object, which can be potentially used for classification, this is only possible when the object to detect does not move which unfortunately is not achievable on marine animal detection tasks.

Other than common vision-based or sonar-based method, other sensors maybe also useful in the marine animal's detection and classification. A novel thermal-camera-based animal detection and classification method with classification accuracy of 84.7% is proposed by researchers at Aarhus University [5]. Not like common vision-based methods, this method uses information obtained from thermal imaging. The detection of animals is based on a threshold dynamically adjusted to each frame. Classification is achieved through a novel thermal feature extraction algorithm which uses morphological operations to find out thermal signature, then applies discrete cosine transformation to parametrize thermal signature, and finally uses k-nearest-neighbor classifier to discriminate animals. This method is novel, but if it can be adjusted to work on the marine animals is doubtful.

B. Marine Animals Image Datasets

Training data are heavily required by the detection and classification of marine animals no matter what methods are chosen. When investigating fish detection and classification, F4K dataset [10] is one of the most commonly used datasets and it is treating as the benchmark dataset due to its extensive annotations. This large dataset contains complex scenes with various marine animal species in the format of videos and images. The build of this dataset takes three years, using 10 monocular cameras to record at reef level, starts from 2010 and ends in 2013 in Taiwan. [6,7,8] projects benefit from this dataset.

J-EDI [11], which stands for Jamstec E-Library of Deepsea Images, is also a commonly used large dataset about underwater marine animals. It contains over 1.5 million clear images taken by remotely operated underwater vehicles at the deep-sea level using monocular cameras. This dataset has been used for years on the training for detection of deep-sea marine animals such as project [9].

QUT Fish Dataset [12] and Croatian Fish Dataset [13] are the two datasets that are focusing on fish, where QUT Fish Dataset is featuring from including both in and out water images, and Croatian Fish Dataset includes more than 10 species of fishes.

Not only fishes, but there are also some datasets monitoring benthic organisms and taken by stereo cameras, such as HabCam dataset [14]. About 2.5 million images of scallops and starfishes are included in this dataset.

A summary of these commonly used datasets is shown in Table 1 which comes from the paper [2], the table illustrates the environment, sensor, number of images in details.

Table 1: Summary of underwater animal datasets

	ENVIRONMENT	**SENSORS**	**AMOUNT**
f4k	Reef	Monocular	27,375 images and 197 videos
J-EDI	Reef	Monocular	1.5 mil. images
Croatian F Dataset	-	Monocular	794 images
QUT Fish Dataset	-	Monocular	3960 images
Habcam	Shelf sea	Stereo	2.5 mil. images

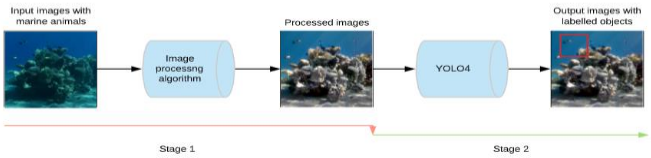

Figure 1:Network structure of WaterDetect

III. METHODOLOGY

In this section, we describe the network structure of our approach in details and introduce the preparation of the dataset. The overall network structure of our method is shown in figure 1 which consists of image processing stage and detection and classification stage. The image processing algorithm is obtained from existing work [1], the researchers proposed a novel underwater image enhancement approach based on Fusion principle. The detection and classification stage is done with the use of YOLOv4 [3] which is the fastest and most accurate detector in terms of both speed and accuracy.

A. Dataset Preparation

The dataset we used in this project is the Brackish dataset [2]. This dataset contains 89 videos with marine animals annotated in 6 different categories include fish, small fish, crab, shrimp, jellyfish, and starfish. However, the videos are separated into 5 folders based on their predominant occurrence of marine animals in video. This means in each folder, not only one category of marine animals will present; the five folders are namely: fish school, fish small shrimp, jellyfish, fish big and crab. Since we are interested in images instead of videos, we further process the dataset by extracting the frames of these videos using OpenCV library. Finally, we got 15085 images in 5 folders.

Then these images are feed into the image processing stage to get enhanced images. Figure 3 illustrate the improvements of image quality before and after image enhancement.

Figure 2: Before and After Images of Brackish Water Dataset

B. Stage One: Image Processing

After having images, the existing work [1] is used in to enhance the quality. This approach is built on the fusion principle, however, instead of using multiple images as inputs, this approach only processes one original degraded image while deriving two inputs and weights from it then applying fusion on the inputs. To be more specific, the image enhancement contains three sequential steps. After feeding the original underwater image to the framework, it first defines two inputs that represent color corrected and contrast enhanced versions of the original image, then defines four weight maps and finally defines multiscale fusion of the inputs and weight maps.

The aim of the first step is to solve the problem of visibility. It first derives an input that represents the color corrected version of the original image. To obtain this input, a white balancing processing is performed to enhance the image appearance by removing unwanted color casts due to various sources of illumination. Gray-World approach of Buchsbaum et al. [24] is revised to perform this task. Nevertheless, only has white balancing is not enough to solve the problem of visibility, therefore, the second input represents the contrast corrected version of original image is derived. This second input is computed from the color corrected and noise-free version of the original image. The noise can be removed by using a temporal bilateral filter [25]. Then, to obtain the contrast corrected image, the classical contrast local adaptive histogram equalization [22] is applied to the noise-free image.

The purpose of the second step is to find the optimized combination of first and second inputs to restore the desired appearance of original images. In this step, four weight measures are defined which are Laplacian contrast weight (W_L), Local contrast weight (W_{LC}), Saliency weight (W_S) and Exposedness weight (W_E). The Laplacian contrast weight (W_L) deals with global contrast by applying a Laplacian filter on each input luminance channel and computing the absolute value of the filter result. Since this weight cannot distinguish between ramp and flat regions, it is not sufficient to restore the contrast, therefore, a Local contrast weight (W_{LC}) is used to assess the local distribution. This measure relates each pixel and its neighborhood average and can be used to strengthen the local contrast appearance. It is computed as the standard deviation between pixel luminance level and the local average of its neighborhoods:

$$w_{LC}(x,y) = \left\| I^k - I^k_{whc} \right\|$$

Where I^k stands for the luminance channel of the input and I^k_{whc} represents the low-passed luminance channel.

The third weight measure, Saliency weight (W_S), is used to emphasize the discriminating objects that lose their prominence in the underwater scene. To do this, the saliency algorithm of Achanta et al. is employed [23]. Since the saliency weight measure tends to emphasize highlighted regions, another weight measure called Exposedness weight (W_E) is used to preserve the mid tones. This weight measure evaluates the exposure level of a pixel. It provides an estimator to preserve a constant appearance of the local contrast. This weight measure is computed as a Gaussian-modeled distance to the average normalized range value (0.5):

$$W_E(x,y) = \exp\left(-\frac{(I^k(x,y)-0.5)^2}{2\sigma^2}\right)$$

Where $I^k(x, y)$ stands for the value of pixel location of the input image I^k, the standard deviation is set to $\sigma = 0.25$.

Furthermore, in order to keep the consistency of results, normalized weight values $\overline{w}$ is employed by constraining the sum of the weight measures at each pixel equals one ($\Sigma\overline{w}^k = 1$):

$$\overline{w}^k = \frac{w^k}{\sum_{k=1}^{K} w^k}$$

Where k represents the index of input.

After having all the above, the final step is to perform the fusion process. The enhanced image $R(x,y)$ is obtained by fusing the inputs with the weight measures at each pixel location (x,y) while performing both Gaussian pyramid and Laplacian pyramids:

$$R^l(x,y) = \sum_{k=1}^{k} G^l\{\overline{w}^k(x,y)\}L^l\{I^k(x,y)\}$$

Where I^k represents the input, which is weighted by the normalized weight maps $\overline{w}^k$.

After using these three steps proposed by [1], we finally obtained enhanced images and ready to move to the object detection and classification stage.

C. Stage Two: Detection and Classification

At this stage, the detector is selected to be YOLOv4 [3] due to its state-of-art performance on operating speed and accuracy, and suitability of training on conventional GPU due to our limited computing resource. YOLOv4 is a single Convolutional Neural Network based detector. It divides an image into regions and then predicts the boundary boxes and probabilities for each region. Its network consists of 3 main parts which are the backbone – CSPDarknet53, Neck – SPP and Head – YOLOv3. Its superior performance on object detection training is achieved by utilizing plenty of tuning methods which are categories into two classes:

- Bag of Freebies: CutMix and Mosaic data augmentation, DropBlock regularization, Class label smoothing, CIoU-loss, CmBN, Self-Adversarial Training, Eliminate grid sensitivity, Using multiple anchors for a single ground truth, Cosine annealing scheduler, Optimal hyperparameters, Random training shapes
- Bag of Specials: Mish activation, Cross-stage partial connections, Multi input weighted residual connections, SPP-block, SAM-block, PAN path-aggregation block, DIoU-NMS

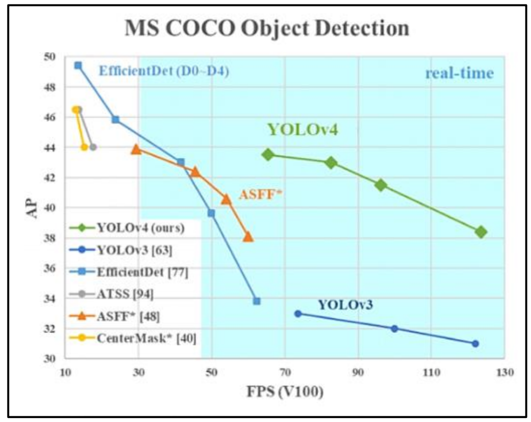

Figure 3: YOLOv4 Performance

Where Bag of Freebies refers to the tuning methods that are widely used in order to improve accuracy without increasing the inference cost, Bag of Specials refers to those methods that significantly improve the accuracy of object detection while only increase the inference cost by a small amount. Beneficial from these tuning methods, the YOLOv4 achieved the state-of-art, comparison of YOLOv4 and other object detectors is shown in figure 2.

After selecting the most suitable object detector for our project, we start to train our model. More details about training are shown in the implementation details section.

IV. Implementation Details

After all images were sorted into the 6 classification categories (fish, small fish, crab, jellyfish, shrimp, and starfish) we implemented Stage 1: Image Processing using MATLAB code [26]. The enhanced images are shown in figure 3.

Images after stage 1 were then divided into different category folders to ensure we are using a balanced dataset for stage 2. The split of the images was as follow:

Table 2: Images in Dataset Categories

Category					
Fish	Small Fish	Crab	Shrimp	Jelly Fish	Starfish
2965	1519	2078	400	367	3666

Note: The images were divided into different category folders based on the maximum number of certain marine animal in an image. This helped to provide a balanced dataset for training our YOLOv4 detector.

For detection and classification stage (stage 2) we tested multiple hypotheses for detector performance under different scenarios to pick the best detector. The training process for different scenarios is listed in Table 3. Equal number of images were randomly selected from each category folder (listed in table 2) and were reseized to 960x540 images from

1920X1080. Annotations for the images were also rescaled to 960X540 before training them on the different detectors. For the training process, we picked equal number of images from each category. Test 1 to 4 were trained using enhanced images after stage 1. Test 5 detector was trained using original unenhanced images (i.e. without stage 1).

Table 3: Stage 2 Detector training scenarios

	Test 1	Test 2	Test 3	Test 4	Test 5
Training Images	120	600	800	800	600
Testing Images	120	120	100	100	100
# of iterations	3000	4000	4000	6000	4000

The hypotheses tested using different training test above are:

1. How number of images used during training stage can improve the detector's performance. The following tests were compared for this hypothesis:
    a. Test 2 - 600 Images
    b. Test 3 - 800 Images
2. How number of iterations during training stage can improve detector's performance. The following tests from training were used for this hypothesis:
    a. Test 3 – 4000 Iterations
    b. Test 4 – 6000 Iterations
3. How training the detector using unenhanced images effect performance compared to testing the detector with enhanced images. The following tests were compared for this hypothesis:
    a. Test 3 – 800 Enhanced Images
    b. Test 5 – 600 Unenhanced Images

Test 1 with 120 images was used to fine tune our YOLOv4 detector and to remove bugs from the code. We selected 120 random images (20 from each category) and trained the detector for 3000 iterations. Next, we used the trained YOLOv4 model to check the bounding boxes on the images to ensure our dataset setup, image resizing and YOLOv4 code are working properly. Since we had a small dataset with 3000 iterations the model should overfit the dataset. If overfitting shows up, the test setup is correct otherwise we pick another 120 images and redo the training.

Table 4: Configuration file parameters

Parameter	Value
Input Image Size	416X416
Batch Size	64
Subdivision	16
Random	1
Alpha	0.001

After all the bugs were removed, the YOLOv4 model was trained using different test scenarios. The parameters of the configuration file for YOLOv4 tests were same (except iteration) to test our hypothesis. Details of the configurations are shown in table 4.

V. EXPERIMENT RESULTS

The stage 1 detector showed drastic improvement in images when compared to the original dataset. An example is shown in figure 3.

For stage 2 of the experiment results are summarized in Table 5. Comparing results between tests 2 and 3 for hypothesis 1 we can see that training with more images almost doubled the precision, recall and F-1 score of the YOLOv4

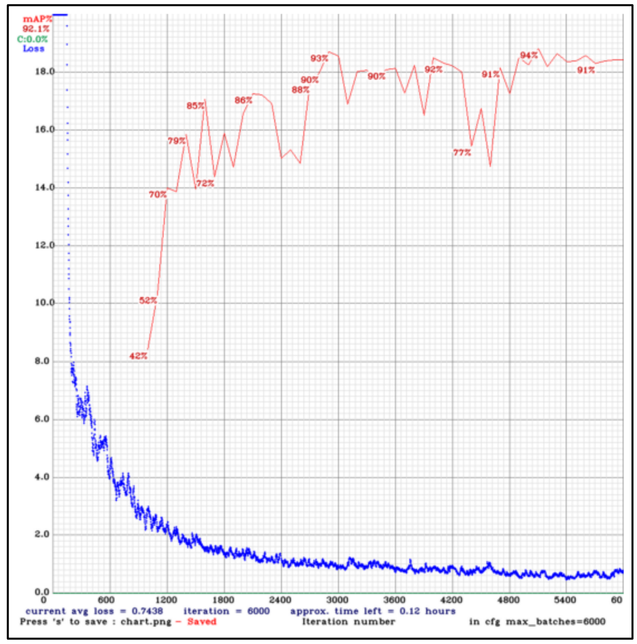

Figure 4: Test 4 Training loss and mAP chart

detector. The mean average precision (mAP) was higher for test 3 compared to test 2 confirming that training with more images is better for object detection.

Table 5: Experiment Result for Different Hypothesis

	Test 1	Test 2	Test 3	Test 4	Test 5
Training Images	120	600	800	800	600
Testing Images	120	120	100	100	100
# of iterations	3000	4000	4000	6000	4000
Precision	0.99	0.48	0.92	**0.93**	0.92
Recall	0.87	0.34	0.86	**0.87**	0.71
F1-Score	0.93	0.39	0.89	**0.90**	0.80
True Positive	392	152	141	**143**	116
False Positive	4	167	12	**10**	10
False Negative	56	299	23	**21**	48
Average IOU	89.45	33.18	74.74	**77.65**	74.73
MAP:	-	85.8%	92%	**92.1%**	79%

Comparing results between tests 3 and 4 for hypothesis 2 we can see that there was a slight improvement in precision, recall and F1-Score. Given more resources, we would like to train for a longer period to check for higher performance improvement. The overall performance of the YOLOv4 detector with higher number of training iteration showed the highest performance. Comparing test 3 and test 5 for hypothesis 3 (both trained for 4000 iterations) it can be inferred from the table that recall and mAP was better for enhanced images in test 3 while all the other parameters were similar. This shows that the detector trained on the enhanced images performs better compared to detector trained on unenhanced images for brackish water environment.

Based on the hypothesis, we selected test 4 detector as our final detector to implement for detecting marine animals in brackish water environment. Figure 4 shows the training loss and mAP graph for test 4. YOLOv4 algorithm only gives the training loss and mAP (for every 4 epochs) graph as per the code from the author. From the figure, we can see that training loss descends smoothly from 2000 average loss to about 0.7438 average loss. The mAP line in red increases, however, has a lot of ups and down which could be due to the testing images. As a lot of images after enhancement were still blurry it could be hard for the detector to recognize marine animals in the frame. Additionally, since we only had 800 images, it was hard for the detector to initially generalize, however, as we reach over 5000 iterations, we can see that the mAP is more stable. While the dataset for training were equally picked, some images had more than one category of marine animal which could make our dataset a little bit skewed resulting in changing mAP during the training process.

Figure 5 below shows some of the detected bounding boxes on the original dataset using the trained detector in test 4.

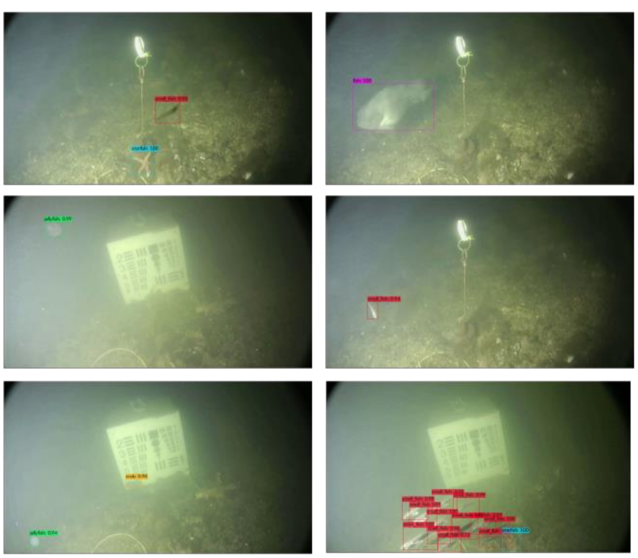

Figure 5: Test 4- Bounding Boxes on Brackish water Dataset

We can see from the images that the bounding boxes are correctly identifying marine animals in the original brackish water dataset with harsh conditions. We can also see that the detector was able to identify crabs on the surface which was hard to see by the human eyes.

VI. CONCLUSION

From all the results we can conclude that our WaterDetect network performs best in harsh underwater brackish environment. The detector was successfully able to identify all the marine animals with high precision and high recall value. When the final detector was compared to the YOLOv4 detector trained on unenhanced brackish water training set, our detector performed better with higher precision and higher mean average precision.

For future work, we would like to train the YOLOv4 detector on all 15000 images for at least 12000 iterations. Additionally, we will benchmark the detector on a standard marine animal detector dataset. We would also like to investigate improved fusion technique that can be implemented to enhance the images further for better training. The future goal of the final detector will be to implement detectors in underwater robots and deploy robots for fish farming, help maintain large aquariums, sharks and jellyfish alert near beaches, to maintain coral reefs and to find plastic in the ocean.

ACKNOWLEDGMENT

We would like to acknowledge Professor Steven Waslander for giving us the opportunity to work on this assignment. I would also like to thank all the researchers that we referenced for their hard work.

PROJECT WORK SPLIT

Both members worked together on researching the dataset and literature review for the project. The project proposal was written by both members of the group. For main project work, Kyle was responsible for writing the report and also helped in running image enhancement code on his computer as that required a lot of computational power. Tushar was responsible for coding, debugging and all training/testing simulation for the project. Tushar setup the coding pipeline and wrote scripts to extract frames, resize images and implement stage 1 and 2 of the projects in python. Tushar also wrote the Implementation, Experiment Result and Conclusion section of the report. Overall, the work was evenly distributed between both team members based on their abilities.

REFERENCES

[1] C. Ancuti, C. O. Ancuti, T. Haber and P. Bekaert, "Enhancing underwater images and videos by fusion," *2012 IEEE Conference on Computer Vision and Pattern Recognition*, Providence, RI, 2012, pp. 81-88, doi: 10.1109/CVPR.2012.6247661.

[2] Pedersen, Malte et al. “Detection of Marine Animals in a New Underwater Dataset with Varying Visibility.” CVPR Workshops ,2019.

[3] Bochkovskiy, Alexey & Wang, Chien-Yao & Liao, Hong-yuan. YOLOv4: Optimal Speed and Accuracy of Object Detection. 2020.

[4]https://oar.princeton.edu/jspui/bitstream/88435/pr13m5n/1/Animal_Detection_Pipeline_for_Identification_2018.pdf

[5] Article Automated Detection and Recognition of Wildlife Using Thermal Cameras

[6] Zheng Cao, Jose C. Principe, Bing Ouyang, Fraser Dalgleish, and Anni Vuorenkoski. Marine animal classification using combined CNN and hand-designed image features. In OCEANS 2015 - MTS/IEEE Washington. IEEE, 2015.

[7] Snigdhaa Hasija, Manas Jyoti Buragohain, and S. Indu. Fish species classification using graph embedding discriminant 25analysis. In 2017 International Conference on Machine Vi-sion and Information Technology (CMVIT). IEEE, 2017.

[8]Ahmad Salman, Shoaib Ahmad Siddiqui, Faisal Shafait, Ajmal Mian, Mark R Shortis, Khawar Khurshid, Adrian Ulges,and Ulrich Schwanecke. Automatic fish detection in under-water videos by a deep neural network-based hybrid motionlearning system. ICES Journal of Marine Science, 2019.

[9] Huimin Lu, Yujie Li, Tomoki Uemura, Zongyuan Ge, XingXu, Li He, Seiichi Serikawa, and Hyoungseop Kim. FDCNet: filtering deep convolutional network for marine organism classification. Multimedia Tools and Applications, 77(17):21847–21860, 2017.

[10]Robert B. Fisher, Yun-Heh Chen-Burger, Daniela Giordano,Lynda Hardman, and Fang-Pang Lin. Fish4Knowledge: Collecting and Analyzing Massive Coral Reef Fish Video
Data. Springer Publishing Company, Incorporated, 1st edition, 2016.

[11]Japan Agency for Marine-Earth Science and Technology. JAMSTEC E-library of deep-sea images, 2016.

[12]Kaneswaran Anantharajah, ZongYuan Ge, Chris McCool, Simon Denman, Clinton Fookes, Peter Corke, Dian Tjondronegoro, and Sridha Sridharan. Local inter-session variability modelling for object classification. In IEEE Winter Conference on Applications of Computer Vision. IEEE,2014.

[13]Jonas Jager, Marcel Simon, Joachim Denzler, Viviane Wolff, Klaus Fricke-Neuderth, and Claudia Kruschel. Croatian fish dataset: Fine-grained classification of fish species in their
natural habitat. In Procedings of the Machine Vision of Animals and their Behaviour Workshop 2015. British Machine Vision Association, 2015.

[14]Northeast Fisheries Science Center. Habitat mapping camera (HABCAM), 2017.

[15] Detection of Marine Animals in a New Underwater Dataset with Varying Visibility

[16]X. Li, M. Shang, J. Hao and Z. Yang, "Accelerating fish detection and recognition by sharing CNNs with objectness learning," OCEANS 2016 - Shanghai, Shanghai, 2016, pp. 1-5, doi: 10.1109/OCEANSAP.2016.7485476.

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

[18] Villon S., Chaumont M., Subsol G., Villéger S., Claverie T., Mouillot D. (2016) Coral Reef Fish Detection and Recognition in Underwater Videos by Supervised Machine Learning: Comparison Between Deep Learning and HOG+SVM Methods.

[19] Helge Balk and Torfinn Lindem. Fish detection in rivers with split-beam sonars. 25nd Scandinavian Symposium on Physics Acoustics, January 2002.

[20] Kieser R. Mulligan T.J. A split-beam echo-counting model for riverine use. ICES Journal of Marine Science, 53:403–406(4), 1996

[22] J. Zimmerman. An evaluation of the effectiveness of adaptive histogram equal-
ization for contrast enhancement. IEEE Trans. on Med. imaging, 1988.

[23] R. Achantay, S. Hemamiz, F. Estraday, and S. Susstrunky. Frequency-tuned
salient region detection. IEEE CVPR, 2009.

[24] M. Ebner. Color Constancy, Wiley 1st edition. 2007.

[25] E. P. Bennett, J. L. Mason, and L. McMillan. Multispectral bilateral video
fusion. IEEE Trans. on Image Processing, 2007.

[26] Github Repository for MATLAB Code Image Enhancement Using Vi a Fusion: https://github.com/IsaacChanghau/ImageEnhanceViaFusion