# UNDERWATER IMAGE ENHANCEMENT USING DOMAIN-ADVERSARIAL LEARNING

A Thesis

by

PRITISH MILIND UPLAVIKAR

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

| | |
|---|---|
| Chair of Committee, | Zhangyang Wang |
| Committee Members, | Theodora Chaspari |
| | Xia Hu |
| | Xiaoning Qian |
| Head of Department, | Dilma da Silva |

May 2019

Major Subject: Computer Science

Copyright 2019 Pritish Milind Uplavikar

# ABSTRACT

Clean underwater images have a variety of applications in marine research, autonomous underwater vehicles and so on. The task of enhancing underwater images is especially difficult because of the diversity with which they are captured. For example, images captured in deep waters look different than those captured in shallow waters. Thus it is difficult to obtain clean underwater images due to lack of a algorithm which handles this diversity. Through our work, we aim to handle this diversity by learning the scene specific features of the images while discarding the features denoting the water type and generate clean underwater images through these learned domain agnostic features. We train our model on a dataset synthesized using NYU Depth Dataset V2 [1]. Our model outperforms quantitative metrics of existing methods for almost all water types and also generalizes well on real world datasets. Performance of underwater images on high level vision taks like object detection also shows improvement after preprocessing with our model.

ACKNOWLEDGMENTS

I would like to thank my advisor Prof. Zhangyang (Atlas) Wang for providing me with valuable guidance. I would also like to thank the committee members Prof. Theodora Chaspari, Prof. Xia Hu and Prof. Xiaoning Qian for carefully reviewing my thesis proposal and the final thesis. I would also like to thank my fellow Visual Informatics Group at Texas A&M (VITA) lab members Nitin Bansal, Zhenyu Wu and Ye Yuan for helping me whenever needed.

Finally, I would like to thank my family and friends for always supporting and continuously encouraging me throughout my years of study and through the process of researching and writing this thesis. This accomplishment would not have been possible without them.

CONTRIBUTORS AND FUNDING SOURCES

## Contributors

This work was supported by a thesis committee consisting of Professor Zhangyang Wang [advisor], Professor Theodora Chaspari and Professor Xia Hu of the Department of Computer Science and Engineering and Professor Xiaoning Qian of the Department of Electrical and Computer Engineering.

All other work conducted for the thesis was completed by the student independently.

## Funding Sources

This work did not have any funding source.

# NOMENCLATURE

| | |
|---|---|
| UIE-DAL | Underwater Image Enhancement using Domain-Adversarial Learning |
| SSIM | Structural Similarity Index |
| PSNR | Peak Signal to Noise Ratio |
| CNN | Convolutional Neural Networks |
| GAN | Generative Adversarial Networks |
| AOD-Net | All-In-One Dehazing Network |
| NYU | New York University |
| UIEBD | Underwater Image Enhancement Benchmark Dataset |
| PCA | Principal Component Analysis |

TABLE OF CONTENTS

Page

LIST OF FIGURES

FIGURE Page

LIST OF TABLES

TABLE Page

# 1. INTRODUCTION

Figure 1.1: Underwater Image Enhancement using Domain-Adversarial Learning (UIE-DAL). From left to right, input image from [2] and our corresponding UIE-DAL output.

Underwater images have an application in many systems such as underwater robotics, autonomous underwater vehicles, marine research and so on. However, the quality of the images acquired for such applications is mostly degraded due to various factors. One of the major factors for this degradation is the wavelength dependent light attenuation over the depth of the object in the scene. For example, red light is absorbed at a higher rate in water than blue or a green light as shown in figures 1.2 and 1.3. Hence we see a blueish or a greenish tint in an underwater scene. Another factor causing degradation is the scattered light due to the presence of small particles in water which adds a homogeneous background noise to the image. This can be seen in figure 1.4. Apart from these factors, one major reason for a lack of a good underwater image enhancement solution is the diversity of underwater image scenes. We can see this diversity in images captured in different types of waters. For example, underwater scenes in deep waters look different than those in shallow waters or scenes captured in clear water versus those captured in muddy water. This can be seen in figure 1.5. Such diversity makes it harder to provide a single solution or train a single model for underwater image enhancement. [7] propose one solution by training different

models for different underwater images. But models trained on single water type don't generalize well and there is no limit to the number of distinct water types to train a model for each of them. Thus these images fail at many vision tasks such as object detection, classification, segmentation and so on which causes a need to process these underwater images and enhance their quality.

Figure 1.2: Absorption of different colors at different depths in water. From left to right - before diving, diving at a depth of 5 m and diving at a depth of 15 m. Reprinted from [3].

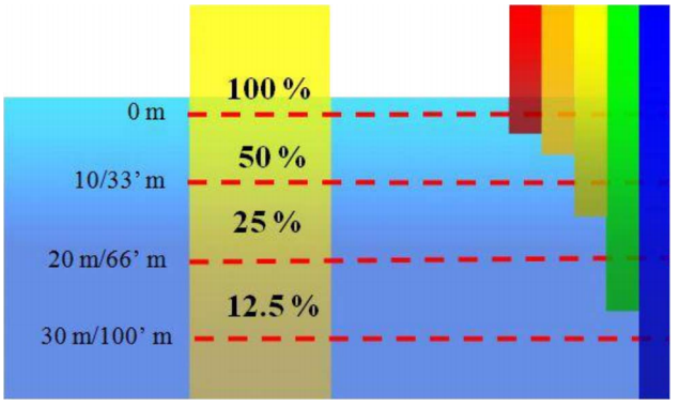


Figure 1.3: Different wavelengths of light are attenuated at different rates in water. The blue color travels the longest in the water due to its shortest wavelength. Reprinted from [3].

Another challenge in the underwater image enhancement problem is the lack of real world datasets with clean images to train models. People have synthesized underwater image datasets

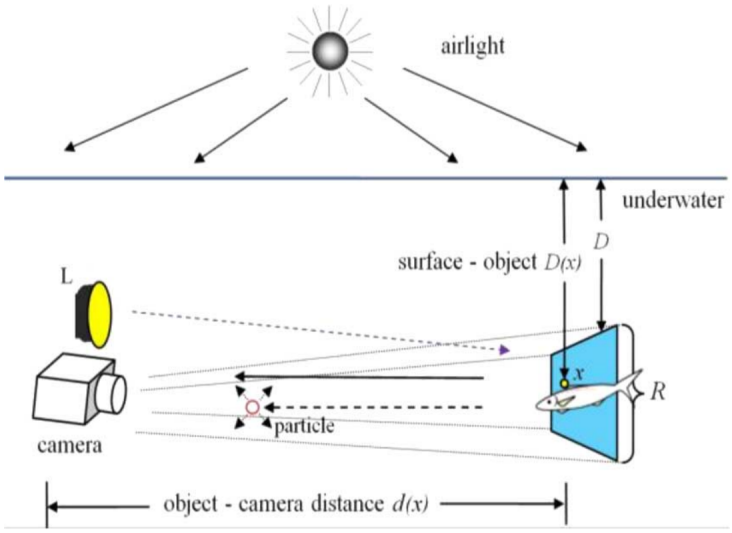


Figure 1.4: Natural light enters from air to an underwater scene point $x$. The light reflected propagates distance $d(x)$ to the camera. The radiance perceived by the camera is the sum of two components: the background light formed by multiscattering and the direct transmission of reflected light. Reprinted from [3].

Figure 1.5: Diversity of underwater scenes. Images are captured in (from left to right) coastal water, deep oceanic water and muddy water. Reprinted from [4], [5] and [6] respectively.

in the past to address this challenge. [7] synthesize such a dataset with clean images pairs for 10 types of water defined by [8]. Samples from such a dataset can be seen in figure 1.6. We use their technique to synthesize a similar dataset to train our model. The task of enhancing underwater images is therefore difficult and has its own unique challenges.

Through this thesis, we propose a solution which would generalize the underwater image enhancement to a certain level. We train a single model over a dataset of multiple underwater image types to generate clean underwater scenes for them. We do so by learning water type agnostic

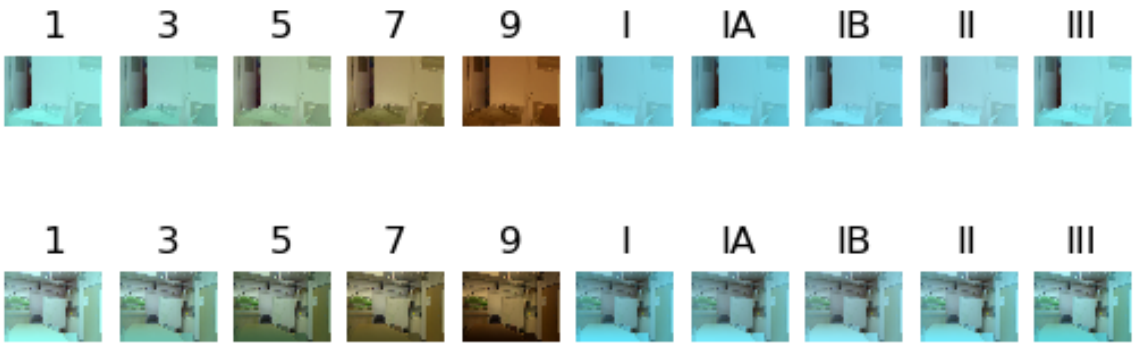


Figure 1.6: Underwater image dataset synthesized in [7].

features of underwater scenes using adversarial training to generate the clean underwater scenes.

## 2. BACKGROUND AND RELATED WORK

Previous work attempting to solve the underwater image enhancement problem was mainly done using physics-based approaches. [9] tried to solve this problem by explicitly modeling the refraction in water. [10] incorporates the inherent properties of the underwater medium such as attenuation, scattering, and the volume scattering function in order to simulate image formation. [11] define an underwater image formation model which is given as

$$U_\lambda(x) = I_\lambda(x)T_\lambda(x) + B_\lambda(1 - T_\lambda(x)) \tag{2.1}$$

where $\lambda$ is the wavelength of the light reaching the camera, $U_\lambda(x)$ is the underwater image, $I_\lambda(x)$ is the clean image, $T_\lambda(x)$ is the fraction of the light reaching the camera after reflecting from point $x$ in the scene and $B_\lambda$ is the homogeneous background light of the scene. $T_\lambda(x)$ is further given as

$$T_\lambda(x) = 10^{-\beta_\lambda d(x)} = \frac{E_\lambda(x, d(x))}{E_\lambda(x, 0)} = N_\lambda(d(x)) \tag{2.2}$$

where $\beta_\lambda$ is the wavelength dependent medium attenuation coefficient, $E_\lambda(x, d(x))$ is the energy of a light beam from point $x$ after it passes through a medium, $N_\lambda(d(x))$ is the normalized residual energy ratio for every unit of depth covered.

The above physical model is similar to that of image dehazing, except that the medium attenuation coefficient is wavelength dependent in this case where as in dehazing it does not depend on the light wavelengths. Hence, image dehazing techniques perform poorly for underwater images. We saw this trend empirically, when we trained AOD-Net [12] over underwater images. This model has been used by many approaches like [7], [13] to solve the underwater image enhancement problem. [7] use the above model to generate a synthetic dataset of 10 water types which we use in our work. The details of the synthetic dataset generation are given in section 3.1.1. [13] tries to improve on the above model by computing attenuation coefficients in the 3D RGB space.

In recent years, deep learning [14] techniques like Convolutional Neural Networks (CNN)

[15] and Generative Adversarial Networks (GAN) [16] have been very effective at solving vision problems. Naturally, these techniques have then been used for underwater image enhancement. [17] trains a GAN to learn the mapping from underwater to clean images. [7] train multiple CNN models, each for different water type in their dataset, to get enhanced images. However, these methods fail to provide a single solution which handles the diversity of underwater images along with generating their clean versions.

# 3. METHODOLOGY

## 3.1 Datasets

Building a real world dataset of a degraded underwater image and its clean version is difficult as it is hard to obtain them. Hence, previous method have tried to synthesize underwater images from their clean versions. We train our model on such a synthesized dataset built using the technique mentioned in [7]. In order to see the applicability of our model, we then test our model on a real world dataset.

### 3.1.1 Synthetic Underwater Images


Figure 3.1: Underwater images synthesized using [7].

We use the NYU-v2 RGB-D dataset [1] to provide us with clean images as it also contains the depth information required to generate the corresponding synthetic images. The synthetic images are generated using the image formation model described before and given by equations 2.1 and 2.2. We generate 6 images of different water types for each image by using different values of $N_\lambda$ for the respective color channels as given in table 3.1. We combine similar image types 1 and 3, I and IA and IB, II and III to reduce the proximity between different water types. This makes the nuisance classifier training easy as it is able to distinguish between different water types more easily. The synthesized 6 types of images for a given image can be seen in figure 3.1. We select the first 1000 of the NYU depth dataset which contains 1449 images. For each image, and for each

of its 6 water types, we augment the dataset by generating 6 images with random $B_\lambda$ and $d(x)$ parameters. Thus the total size for this dataset is 1000x6x6, i.e. 36,000 images.

| Types | I | IA | IB | II | III | 1 | 3 | 5 | 7 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|
| blue | 0.982 | 0.975 | 0.968 | 0.94 | 0.89 | 0.875 | 0.8 | 0.67 | 0.5 | 0.29 |
| green | 0.961 | 0.955 | 0.95 | 0.925 | 0.885 | 0.885 | 0.82 | 0.73 | 0.61 | 0.46 |
| red | 0.805 | 0.804 | 0.83 | 0.8 | 0.75 | 0.75 | 0.71 | 0.67 | 0.62 | 0.55 |

Table 3.1: $N_\lambda$ values to generate 10 types of water images. Reprinted from [7].

#### 3.1.2 Real-world Underwater Images

We use Underwater Image Enhancement Benchmark Dataset built by [2] as our real world underwater image dataset. The dataset consists of 890 underwater images but does not have corresponding ground truth images.

### 3.2 Model architecture

#### 3.2.1 Overview

One of our main goals, apart from generating a clean underwater image, is to train a single model for multiple water types. [7] do so by training a different model for each water type. To train a single model, we first try to get a domain agnostic encoding for the given input water type image. That means, for the same underwater scene, captured in different water types, the latent vector $Z$ extracted from an encoder $E$ for that scene should be the same for all water types. That way the decoder or the generator $G$ is able to reconstruct a clean image of the scene from only the scene specific features while discarding the domain specific features. Both $E$ and $G$ are neural networks in our model.

To do so, we introduce a novel application of a nuisance classifier $D$ along with $E$ and $G$. The nuisance classifier is a neural network which aims to classify the water type of the given input image from its extracted latent vector $Z$ from the encoder. However, we also introduce

an adversarial loss [16] over the encoder using the nuisance classifier. Our formulation of the adversarial loss forces the encoder to generate $Z$ such that the nuisance classifier outputs a uniform distribution over the possible water types. Thus, the adversarial loss combined with the nuisance loss forces the encoding to be agnostic of the input water type. The full architecture can be seen in figure 3.2.

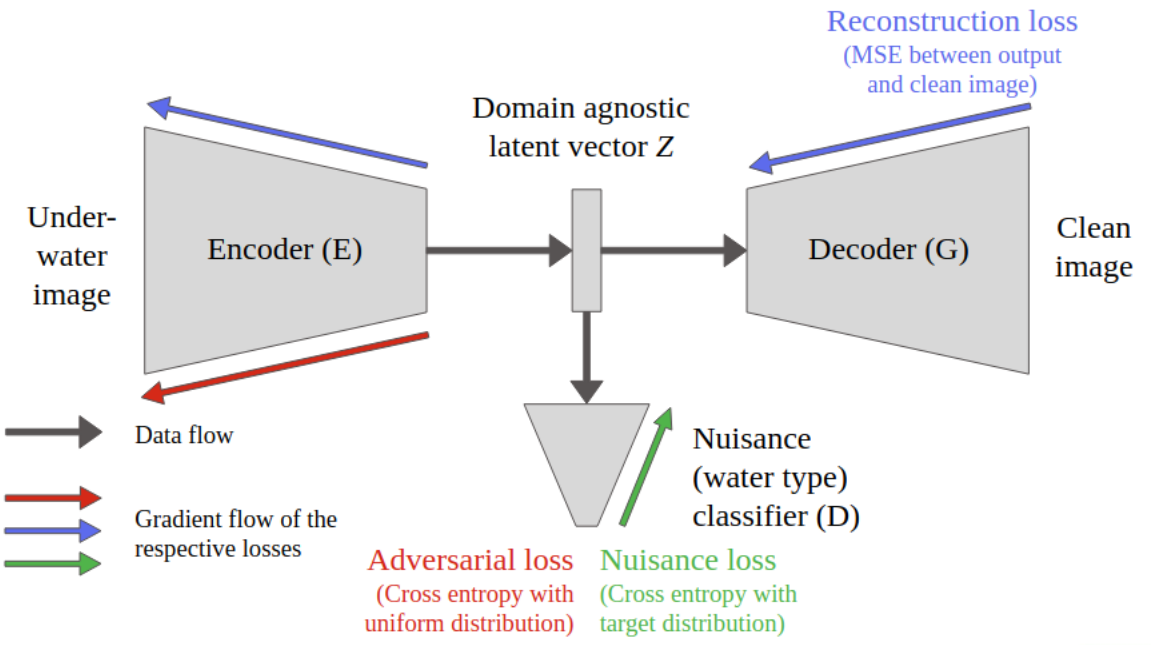


Figure 3.2: Our model architecture.

### 3.2.2 Losses

Our model consists of three losses, namely, the reconstruction loss $L_R$, the nuisance loss $L_N$ and the adversarial loss $L_A$. These losses force the model to generate a clean image while discarding the features denoting the water type. Detailed information about all the three losses can be found below.

#### 3.2.2.1 Reconstruction loss

We compute a reconstruction loss $L_R$, which is the mean squared error between the image generated by $G$ and the clean image label $Y$ for the given input image $X$. This reconstruction loss is backpropagated to update both $G$ and $E$. The reconstruction loss is given by the following equation

$$L_R(X, Y) = \frac{1}{N}.\sum_{i=1}^{N}|G(Z)_i - Y_i|^2 \tag{3.1}$$

where $Z = E(X)$ and $N$ is the number of pixels.

#### 3.2.2.2 Nuisance loss

We compute a nuisance loss $L_N$, which is the cross entropy between the predicted water type from $D$, for the latent vector $Z$ of the input image $X$, and the target water type $C$. This nuisance loss is backpropagated to only to update $D$. The nuisance loss is given as

$$L_N(X, C) = -\sum_{c=1}^{M} y_c \log D(Z)_c \tag{3.2}$$

where $y_c = 1$ if $c = C$ else $y_c = 0$, $Z = E(X)$ and $M$ is the number of classes.

#### 3.2.2.3 Adversarial loss

We compute an adversarial loss [16] $L_A$, which is the cross entropy with uniform distribution of the predicted distribution of water types from $D$, for the latent vector $Z$ of the input image $X$. This adversarial loss is backpropagated to only to update $E$. The adversarial loss is given as

$$L_A(X) = -\frac{1}{M}\sum_{c=1}^{M} \log D(Z)_c \tag{3.3}$$

where $Z = E(X)$ and $M$ is the number of classes.

Our training strategy involves jointly training all the modules by using the above losses. At each iteration, we first backpropagate the reconstruction loss, then backpropagate the adversarial

loss and finally backpropagate the nuisance loss.

### 3.2.3 Training procedure

We train our model by following a procedure which prioritizes the adversarial training of the encoder while also makes sure that the nuisance classifier is strong enough. Keeping the nuisance classifier strong is critical for good adversarial training of the encoder. Algorithm 1 shows the training procedure we follow.

**Algorithm 1** Training procedure of our model

Given an encoder $E$, decoder $G$ and nuisance classifier $D$

Compute $val_G \leftarrow$ Cross validation SSIM score for $G$

**while** $val_G < threshold_G$ **do**
  Update $E$ and $G$ using $L_R$

**end**

**for** $n$ *training epochs* **do**

  **if** $val_G < threshold_G$ **then**
    Update $E$ using $L_R$ and $L_A$, $G$ using $L_R$

  **else if** $val_D < threshold_D$ **then**
    Update $D$ using $L_N$

  **else**
    Update $E$ using $L_R$ and $L_A$, $G$ using $L_R$

  **end**

  Compute $val_G, val_D \leftarrow$ Cross validation SSIM score for $G$, Cross validation accuracy for $D$

**end**

### 3.2.4 Experimental details

We train our model on the synthetic dataset generated by following technique in section 3.1.1. The specifications of the machine on which the model was trained are - Intel i7 6700 HQ processor,

8 GB RAM, NVIDIA GeForce GTX 960M 4GB graphics card.

We use U-Net [18] as our encoder-decoder architecture. U-Net is useful as the skip connections between encoder and decoder provide local and global information for decoder to generate clean images. Also, it is a fully convolutional neural network which means it can handle images of varying sizes. The U-Net architecture can be seen in figure 3.3. Our nuisance classifier $D$ is a convolutional neural network which predicts probability of 6 classes. Its architecture can be seen in figure 3.4.

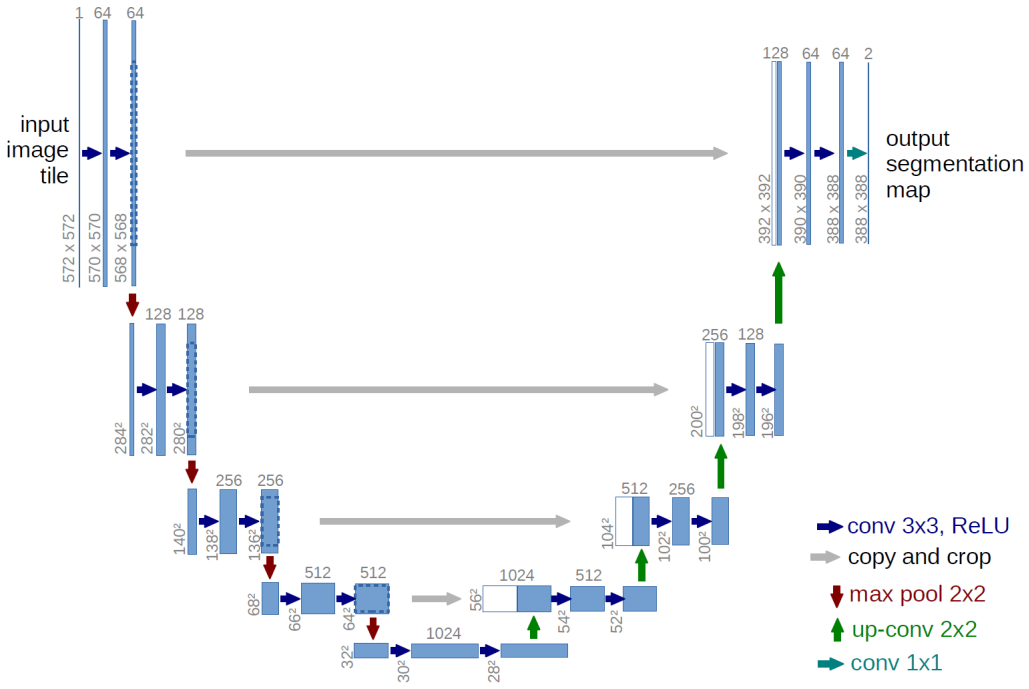


Figure 3.3: U-Net architecture.

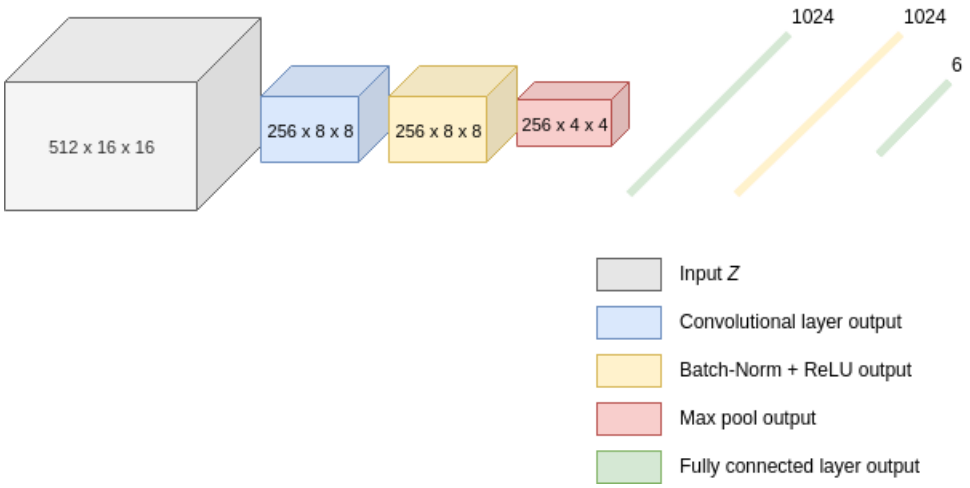


Figure 3.4: Nuisance classifier architecture.

### 4.1 Qualitative results on Underwater NYU Depth Dataset

We use the last 3000 images as the test set from our synthesized dataset of 36,000 images. Figure 4.1 shows some visual results of our model on the test set of Underwater NYU Depth Dataset which we synthesized in section 3.1.1. We can visually see that results are pretty good and that the output images recover even the minute details from the degraded input images.

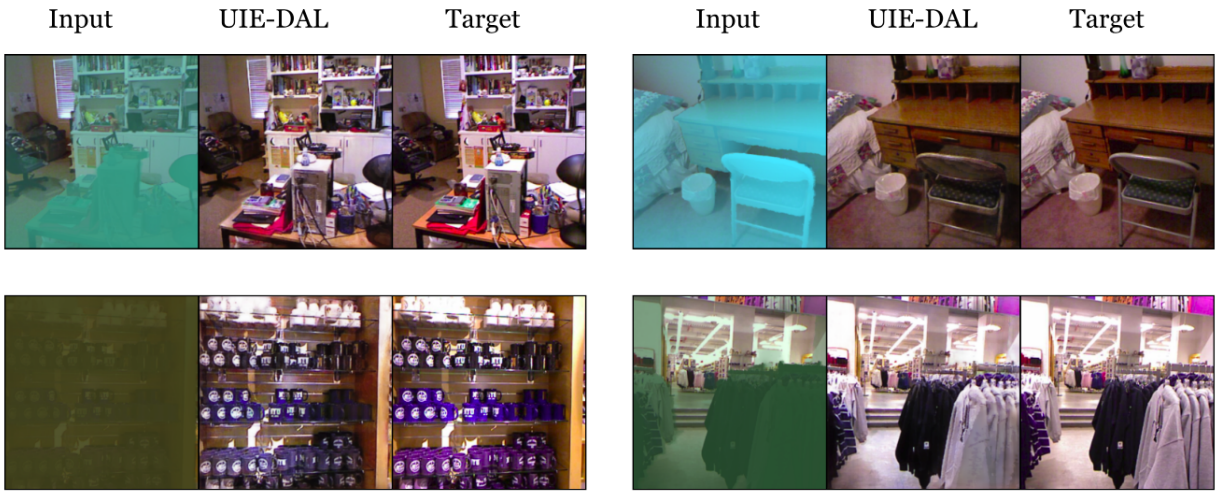


Figure 4.1: Results of our model on underwater NYU Depth Dataset.

### 4.2 Quantitative results for 10 Jerlov water types

We also compute quantitative evaluation metrics like SSIM [19] and PSNR for the generated images of different Jerlov water types [8] with respect to their clean counterparts. As seen in table 4.1, our model outperforms other methods for almost all water types.

### 4.3 Qualitative results on a real world dataset

We also test our model on a real world dataset to see the transferabiltiy of our model to different datasets. Figure 4.2 shows some visual results of our model on the Underwater Image Enhance-

| | Water Type | RAW | RED | UDCP | ODM | UIBLA | UWCNN | UIE-DAL |
|---|---|---|---|---|---|---|---|---|
| **SSIM** | 1 | 0.7065 | 0.7406 | 0.7629 | 0.724 | 0.6957 | 0.8558 | **0.9313** |
| | 3 | 0.5788 | 0.6639 | 0.6614 | 0.6765 | 0.5765 | 0.7951 | |
| | 5 | 0.4219 | 0.5934 | 0.4269 | 0.6441 | 0.4748 | 0.7266 | **0.9364** |
| | 7 | 0.2797 | 0.5089 | 0.2628 | 0.5632 | 0.3052 | 0.607 | **0.9353** |
| | 9 | 0.1794 | 0.3192 | 0.1624 | 0.4178 | 0.2202 | 0.492 | **0.925** |
| | I | 0.8621 | 0.8816 | 0.8264 | 0.8172 | 0.7449 | **0.9376** | 0.9129 |
| | II | 0.8716 | 0.8837 | 0.8387 | 0.8251 | 0.8017 | 0.9236 | **0.9235** |
| | III | 0.7526 | 0.7911 | 0.7587 | 0.7546 | 0.7655 | 0.8795 | |
| **PSNR** | 1 | 15.535 | 15.596 | 15.757 | 16.085 | 15.079 | 21.79 | **28.4488** |
| | 3 | 14.688 | 12.789 | 14.474 | 14.282 | 13.442 | 20.251 | |
| | 5 | 12.142 | 11.123 | 10.862 | 14.123 | 12.611 | 17.517 | **28.6697** |
| | 7 | 10.171 | 9.991 | 9.467 | 12.266 | 10.753 | 14.219 | **28.5793** |
| | 9 | 9.502 | 11.62 | 9.317 | 9.302 | 10.09 | 13.232 | **27.6551** |
| | I | 17.356 | 19.545 | 18.816 | 18.095 | 17.488 | 25.927 | **27.1015** |
| | II | 20.595 | 20.791 | 17.204 | 17.61 | 18.064 | 24.817 | **28.1602** |
| | III | 16.556 | 16.69 | 14.924 | 16.71 | 17.1 | 22.633 | |

Table 4.1: Comparison of our model (UIE-DAL) with SSIM, PSNR values of previous methods. Higher values mean better results. Bold values show the best performer. Values of the previous methods are reprinted from [7].

ment Benchmark Dataset (UIEBD) built by [2]. Here we see that the model performs well and is able to generalize on image distributions different than that of the training images. Handling such diversity is one of our main goals apart from generating clean underwater images.

### 4.4 Comparison with U-Net without adversarial loss

We compare our model with vanilla U-Net without the adversarial loss. To see if we are indeed learning the domain agnostic features, we plot first two principal components of the encoding $Z$ for both the vanilla U-Net and U-Net with the adversarial loss. We color the points once by the water types and once by the image content for the same set of images. The plotted PCA components can be seen in figures 4.3 and 4.4 respectively.

It can be seen from figures 4.3 and 4.4 that we are indeed learning domain agnostic features using adversarial loss. The encoding $Z$ is clustered by the water types in vanilla U-Net, whereas it is clustered by the image content in U-Net with adversarial loss.

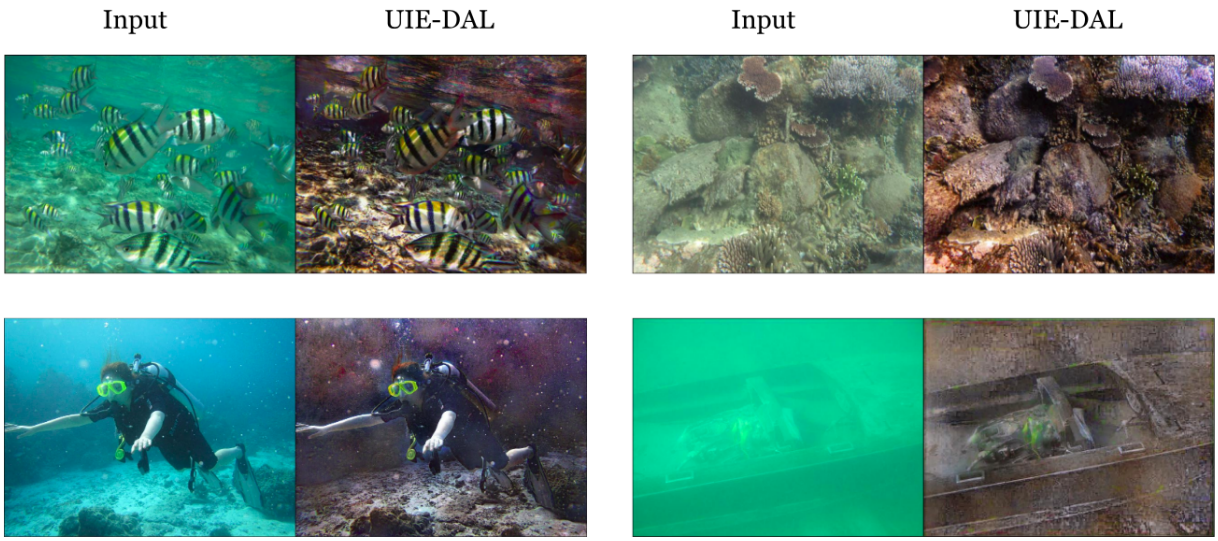


Figure 4.2: Results of our model on the real world UIEBD Dataset.

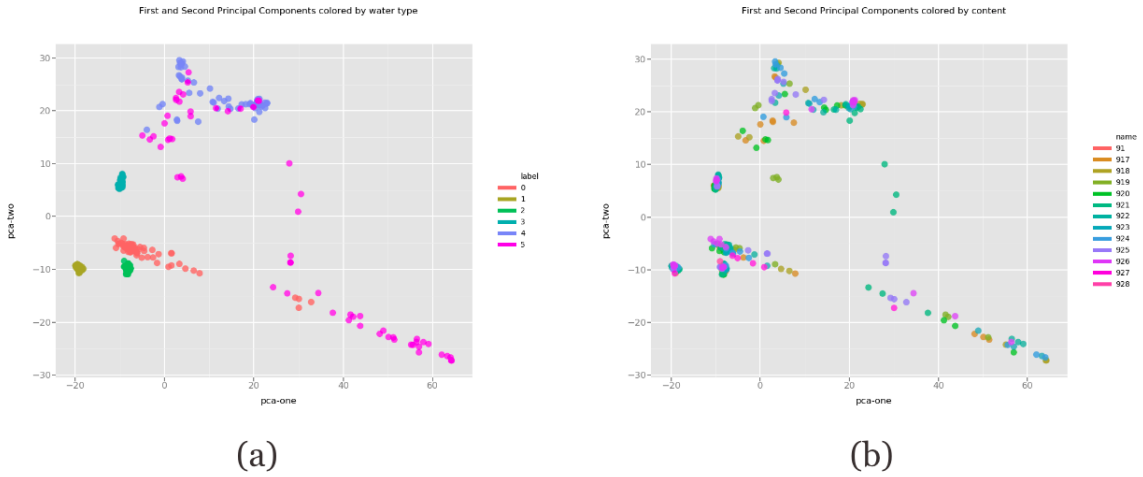


Figure 4.3: Visualizing first two PCA components of the encoding Z learnt by U-Net without adversarial loss. (a) Colors points with same water type, (b) Colors points with same content

We also visually and quantitatively compare both the models. Figure 4.5 shows us the visual results of the models on both the synthetic Underwater NYU Depth Dataset and real world UIEBD Dataset. Table 4.2 shows us the quantitative comparison.

We can see from both Figure 4.5 and Table 4.2 that U-Net with adversarial loss outperforms vanilla U-Net. U-Net with adversarial loss is able to learn domain agnostic features and hence also

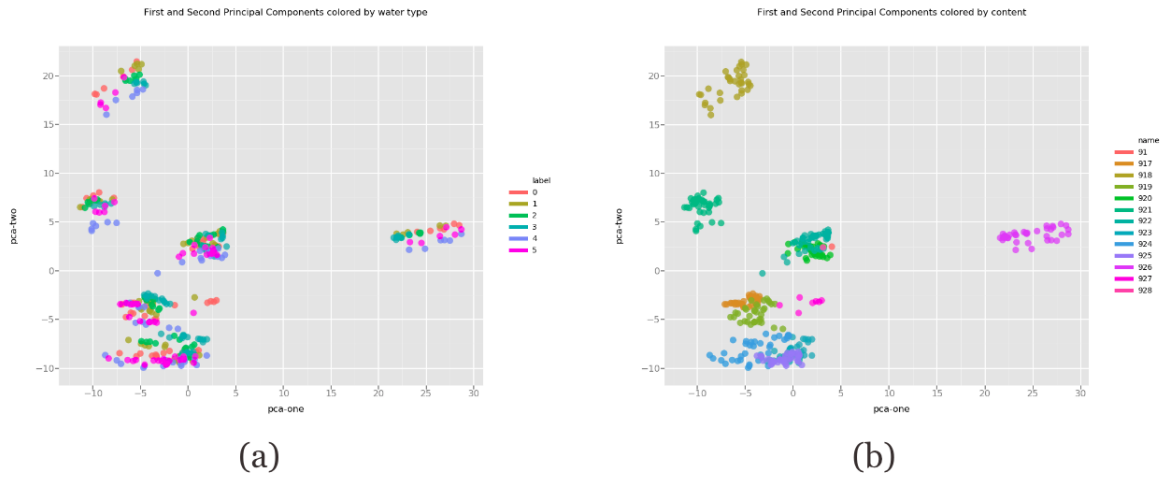


Figure 4.4: Visualizing first two PCA components of the encoding Z learnt by U-Net with adversarial loss (UIE-DAL). (a) Colors points with same water type, (b) Colors points with same content.

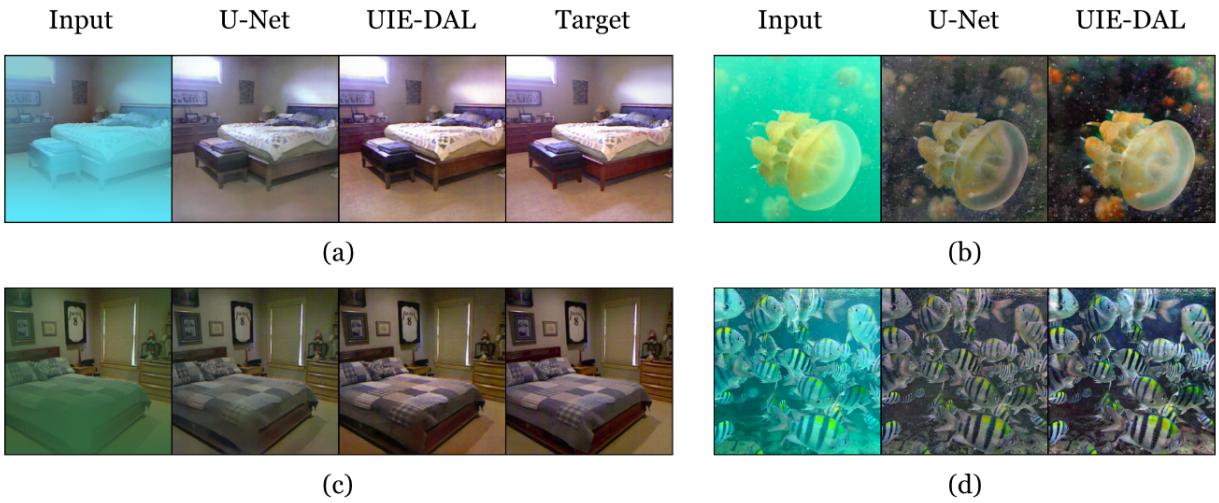


Figure 4.5: Comparison of U-Net with and without adversarial loss. Figures (a), (c) show results on synthetic data, where as (b), (d) show results on real world data

generates images with rich color quality than vanilla U-Net.

## 4.5 Object detection on enhanced images

We run object detection experiments on the images generated by our model to see if they can help in different underwater vision tasks. We run YOLO v3 [20] object detector on the degraded

| | Water Type | U-Net | UIE-DAL (Ours) |
|---|---|---|---|
| **SSIM** | 1 | 0.8691 | **0.9313** |
| | 3 | | |
| | 5 | 0.8733 | **0.9364** |
| | 7 | 0.8687 | **0.9353** |
| | 9 | 0.8614 | **0.925** |
| | I | 0.8385 | **0.9129** |
| | II | 0.8385 | **0.9235** |
| | III | | |
| **PSNR** | 1 | 21.6283 | **28.4488** |
| | 3 | | |
| | 5 | 22.6119 | **28.6697** |
| | 7 | 22.5754 | **28.5793** |
| | 9 | 22.5263 | **27.6551** |
| | I | 22.3236 | **27.1015** |
| | II | 21.8279 | **28.1602** |
| | III | | |

Table 4.2: Our comparison with SSIM, PSNR values of U-Net without adversarial loss. Higher values mean better results. Bold values show the best performer.

and underwater images. We observe that object detection is better on the images generated by our model compared to the degraded underwater images of the synthesized Underwater NYU Depth Dataset. However, we get mixed results when we run the object detector on the real world UIEBD Dataset. Figure 4.6 shows the results of YOLO v3.

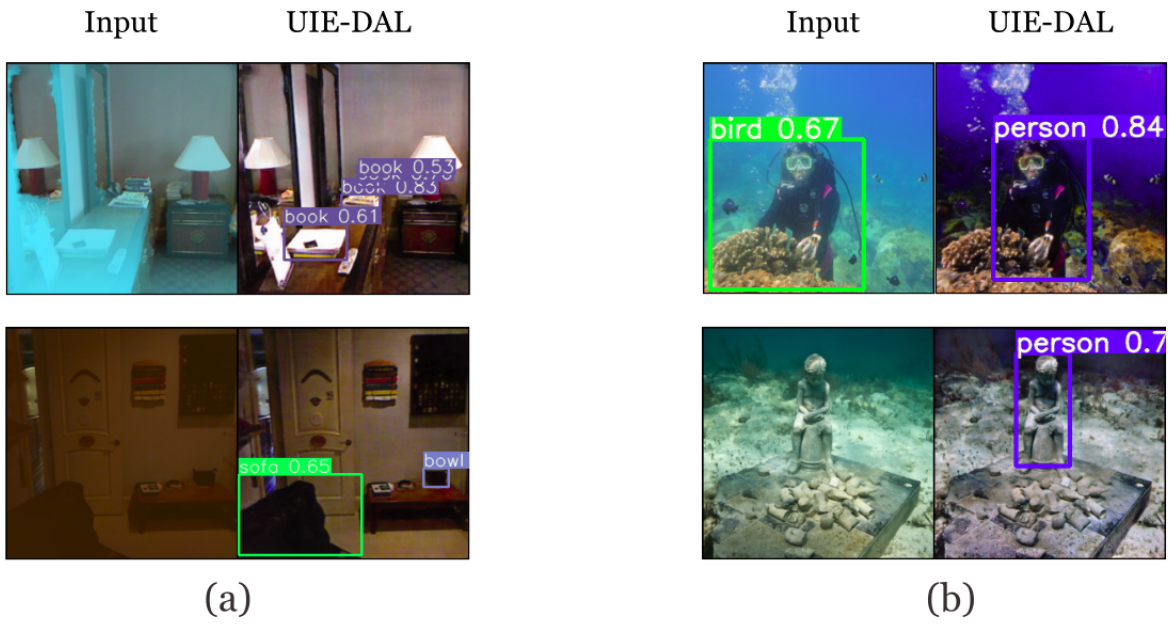


Figure 4.6: Object detection results on (a) Synthesized underwater NYU Depth Dataset and (b) UIEBD real world dataset. Results were good for the synthetic dataset where as the real world dataset had mixed results.

# 5. FUTURE WORK

We also plan to run our model on different datasets like Marine Hydrodynamics Laboratory (MHL) and Jamaica datasets collected by [21] and compute quantitative metrics like quantitative evaluation of color restoration using the dataset and method provided by [22] for them.

Some generated images from the real world dataset have reddish tones on them. We plan to eliminate them by doing a GAN-like training by adding a discriminator.

We also plan on experimenting with different model architectures, for example, using SegNet [18] for the encoder-generator module.

# 6. CONCLUSION

Through this thesis we were able to provide a novel solution for underwater image enhancement which outperforms a lot of the previous methods both qualitatively and quantitatively. The goal of the thesis was to provide a generalized solution which could handle the diversity of the underwater images as well as generating quality clean images for them. Our model was successful in doing so by learning domain agnostic features of multiple water types and then generating a clean version of the image from those features. We also showed that the model was able to generalize well on the unseen real world data. Also, object detection results showed that preprocessing underwater images with our model before high level vision tasks improves the task performance.

REFERENCES


[1] P. K. Nathan Silberman, Derek Hoiem and R. Fergus, "Indoor segmentation and support inference from rgbd images," in *ECCV*, 2012.

[2] C. Li, C. Guo, W. Ren, R. Cong, J. Hou, S. Kwong, and D. Tao, "An underwater image enhancement benchmark dataset and beyond," *CoRR*, vol. abs/1901.05495, 2019.

[3] J. Y. Chiang and Y. Chen, "Underwater image enhancement by wavelength compensation and dehazing," *IEEE Transactions on Image Processing*, vol. 21, pp. 1756–1769, April 2012.

[4] "Treasure or toxin? failed artificial reef made off socal coast is being removed after decades." https://www.mercurynews.com/wp-content/uploads/2017/ 10/ image001-1.png?w=620.

[5] "Sharks and scorpions? the world's deadliest animals aren't what you thought." https: // www.dw.com/image/15773043_304.jpg.

[6] "How to catch crappie in muddy water." www.reelchase.com/wp-content uploads/2017/03/Learn-the-Best-Tips-on-How-to-Catch-Crappie-in-Muddy-Water. jpg.

[7] S. Anwar, C. Li, and F. Porikli, "Deep underwater image enhancement," *CoRR*, vol. abs/1807.03528, 2018.

[8] N. Jerlov, *Marine Optics*. Elsevier, 1976.

[9] A. Jordt, *Underwater 3d reconstruction based on physical models for refraction and underwater light propagation*. PhD thesis, 2013.

[10] J. S. Jaffe, "Computer modeling and the design of optimal underwater imaging systems," *IEEE J. Oceanic Engin.*, vol. 15, pp. 101–111, 1990.

[11] J. Y. Chiang and Y. Chen, "Underwater image enhancement by wavelength compensation and dehazing," *IEEE Transactions on Image Processing*, vol. 21, pp. 1756–1769, April 2012.

[12] B. Li, X. Peng, Z. Wang, J. Xu, and D. Feng, "Aod-net: All-in-one dehazing network," in *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 4780–4788, Oct 2017.

[13] D. Akkaynak, T. Treibitz, T. Shlesinger, Y. Loya, R. Tamir, and D. Iluz, "What is the space of attenuation coefficients in underwater computer vision?," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 568–577, July 2017.

[14] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[15] Y. LeCun and Y. Bengio, "The handbook of brain theory and neural networks," ch. Convolutional Networks for Images, Speech, and Time Series, pp. 255–258, Cambridge, MA, USA: MIT Press, 1998.

[16] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems 27* (Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, eds.), pp. 2672–2680, Curran Associates, Inc., 2014.

[17] C. Fabbri, M. J. Islam, and J. Sattar, "Enhancing underwater imagery using generative adversarial networks," *CoRR*, vol. abs/1801.04011, 2018.

[18] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," *CoRR*, vol. abs/1505.04597, 2015.

[19] Z. Wang, A. C. Bovik, H. R. Sheikh, E. P. Simoncelli, *et al.*, "Image quality assessment: from error visibility to structural similarity," *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.

[20] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *CoRR*, vol. abs/1804.02767, 2018.

[21] J. Li, K. A. Skinner, R. M. Eustice, and M. Johnson-Roberson, "Watergan: Unsupervised generative network to enable real-time color correction of monocular underwater images," *CoRR*, vol. abs/1702.07392, 2017.

[22] D. Berman, D. Levy, S. Avidan, and T. Treibitz, "Underwater single image color restoration using haze-lines and a new quantitative dataset," *CoRR*, vol. abs/1811.01343, 2018.