All-In-One Underwater Image Enhancement using Domain-Adversarial Learning

Pritish Uplavikar[1], Zhenyu Wu[1], and Zhangyang Wang[1]

[1]Texas A&M University, College Station, TX
{*pritishuplavikar, wuzhenyu_sjtu, atlaswang*}@tamu.edu

Abstract

Raw underwater images are degraded due to wavelength dependent light attenuation and scattering, limiting their applicability in vision systems. Another factor that makes enhancing underwater images particularly challenging is the diversity of the water types in which they are captured. For example, images captured in deep oceanic waters have a different distribution from those captured in shallow coastal waters. Such diversity makes it hard to train a single model to enhance underwater images. In this work, we propose a novel model which nicely handles the diversity of water during the enhancement, by adversarially learning the content features of the images by disentangling the unwanted nuisances corresponding to water types (viewed as different domains). We use the learned domain agnostic features to generate enhanced underwater images. We train our model on a dataset consisting images of 10 Jerlov water types [1]. Experimental results show that the proposed model not only outperforms the previous methods in SSIM and PSNR scores for almost all Jerlov water types but also generalizes well on real-world datasets. The performance of a high-level vision task (object detection) also shows improvement using enhanced images with our model.

1. Introduction

Underwater images have an application in a variety of fields like marine research and underwater robotics. We need clear underwater imagery to study deteriorating coral reefs and other aquatic life. Underwater robotic systems also rely heavily on high quality images to fulfill their objectives. However, the quality of the images acquired for these applications is degraded due to various factors. One of the major factors for this degradation is wavelength dependent light attenuation over the depth of the object in the scene. For example, red light is absorbed in water at a higher rate

Figure 1. 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.

than blue or green light. Hence, we see a blueish or a greenish tint in an underwater scene. Another factor diminishing underwater image quality is the light scattered due to the small particles present in water, which introduces a homogeneous background noise to the image.

Apart from these factors, another challenge in underwater image enhancement is the diversity of underwater image distributions. We can see this diversity in figure 1, which shows how underwater scenes captured in shallow coastal waters look different than those captured in deep oceanic waters or those captured in muddy waters. It is hard for a single model to enhance underwater images for such multiple image distributions and, therefore, providing a universal solution for underwater image enhancement is difficult. While previous work has addressed the challenges of light attenuation and scattering, not many have handled the challenge of image distribution diversity explicitly. [2] proposes color restoration of underwater images by performing color correction using attenuation coefficient ratios for all the Jerlov water types and then selecting the best result out of them. Whereas, [3] proposes one solution by training multiple models, each for a different Jerlov water type. But these approaches seem inefficient and rely on the prior knowledge of the water type for the given image to perform color restoration.

One more challenge faced in underwater image enhancement is the lack of real-world datasets containing the ground truth clear images, as it is extremely difficult to find degraded and clear versions of the same real-world underwa-

ter scene, thus creating a bottleneck for data-driven methods. To address this challenge, synthetic underwater image datasets have been built in the past. One such example is the work done in [3], who synthesize an underwater image dataset consisting images of 10 Jerlov water types using the NYU Depth Dataset V2 [7] which provides the ground truth clear images and the depth, both of which are required to synthesize degraded underwater images using the underwater image formation model [8].

The task of enhancing underwater images is, therefore, difficult and has its own unique challenges. These images fail at many vision tasks like object detection, classification, segmentation which provokes a need to process them and enhance their quality. We propose a novel solution to this problem by addressing all the challenges mentioned above using a convolutional neural network [9] based encoder-decoder to reconstruct clear underwater images and a convolutional neural network based classifier to classify the Jerlov water types, which acts as our *nuisance* classifier. We synthesize a dataset of synthetic underwater images by following [3] to train our model, thus taking into account the factors of wavelength dependent light attenuation and scattering of light due to small water particles while forming an underwater image and thereby addressing the respective challenges. To address the challenge of underwater image distribution diversity, we train our model to learn the domain agnostic features for a given degraded underwater image, where the domain is the Jerlov water type of the image. The objective of our encoder, apart from learning an encoding to reconstruct a clear underwater image, is to make the prediction of the nuisance classifier as uncertain as possible by discarding the features denoting the water type and preserves only the scene related features [10]. This is similar to a generator fooling a discriminator in a generative adversarial network [11], the only difference being that in this case we want to make the classifier uncertain. We, therefore, introduce an adversarial loss on our encoder, computed at the end of the nuisance classifier, which is negative entropy. The encoder is also acted upon by a reconstruction loss computed at the end of our decoder. Thus, we propose a novel model which performs underwater color restoration for multiple types of underwater images.

Our approach has multiple highlights: i) Our proposed approach is able to learn water-type agnostic features. We adapt the adversarial training strategy proposed in [10] to our model; ii) Our proposed model outperforms the previous enhancement methods in SSIM and PSNR scores for almost all Jerlov water types; iii) Our model has good generalization ability on real-world datasets, as well as performs nicely on improving subsequent object detection on enhanced images.

2. Related Work

Many previous attempts to solve the underwater image enhancement problem have used physics-based methods. [12] tries to solve this problem by explicitly modeling the refraction in water, whereas, [13] incorporates the inherent properties of the underwater medium such as attenuation, scattering, and the volume scattering function in order to simulate image formation. [8] defines an underwater image formation model which is given as

$$U_c(x) = I_c(x)T_c(x) + B_c(1 - T_c(x)), c \in \{r, g, b\} \quad (1)$$

where $U_c(x)$ is a point x in the underwater image, $I_c(x)$ is a point x in the clear image, $T_c(x)$ is the fraction of the light reaching the camera after reflecting from point x in the scene and B_c is the homogeneous background light of the scene. $T_c(x)$ is further given as

$$T_c(x) = 10^{-\beta_c d(x)} = \frac{E_c(x, d(x))}{E_c(x, 0)} = N_c d(x) \quad (2)$$

where β_c is the wavelength dependent medium attenuation coefficient, $E_c(x, d(x))$ is the energy of a light beam from point x after it passes through a medium and $N_c(d(x))$ 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, whereas in dehazing it does not depend on the light wavelength. This model has been used by many approaches to solve the underwater image enhancement problem. [14] tries to improve on the above model by computing attenuation coefficients in the 3D RGB space, whereas [3] uses the above model to generate a synthetic dataset of 10 Jerlov water types. We generate a similar dataset in our work, the details of which are given in section 4.2.1.

In recent years, deep learning [15] techniques like Convolutional Neural Networks (CNN) [9] and Generative Adversarial Networks (GAN) [11] have been very effective at solving vision problems. Naturally, these techniques have then been used for underwater image enhancement. [16] trains a GAN to learn the mapping from underwater to clear images. [3] train multiple CNN models, each for different water type in their dataset, to get enhanced images. However, these methods fail to provide a singular solution capable of handling the diversity of underwater images apart from generating their clear versions.

3. Method

Since one of our goals, apart from underwater image enhancement, is to train a single model which can do this task
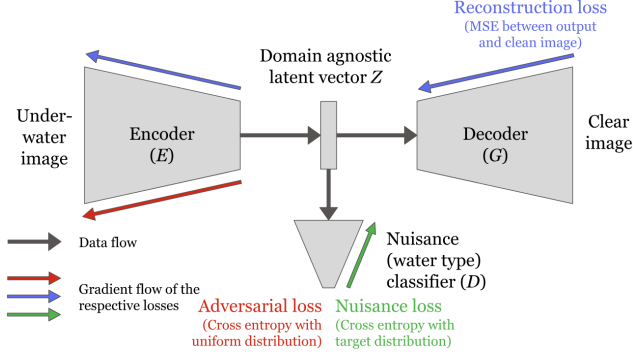
Figure 2. Our model architecture.

for multiple water types, we first try to learn a water type agnostic encoding for the given underwater image. That means, ideally, the latent vector Z extracted from an encoder E for the same underwater scene, should be the same for different water types. That way the decoder or the generator G is able to reconstruct a clear image of the scene from only the scene 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 latent vector Z extracted from the encoder. However, we also introduce an adversarial loss [11] over the encoder using the nuisance classifier. Our formulation of the adversarial loss forces the encoder to generate Z such that the nuisance classifier is unsure of the possible water types. Thus, the adversarial loss forces the encoding to be agnostic of the features denoting the water type. The full architecture can be seen in figure 2.

3.1. Losses

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

3.1.1 Reconstruction loss

We compute a reconstruction loss L_R, which is the mean squared error between the image generated by G, from the latent vector Z, and the clear image ground truth Y for the given input image X. The reconstruction loss is given as

$$L_R(X,Y) = \frac{1}{N} \cdot \sum_{i=1}^{N} |G(Z)_i - Y_i|^2 \qquad (3)$$

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

3.1.2 Nuisance loss

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

$$L_N(X,C) = -\sum_{c=1}^{M} y_c \log D(Z)_c, \qquad (4)$$

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

3.1.3 Adversarial loss

As we want to increase the uncertainty or entropy of the nuisance classifier, we try to reduce the certainty or negative entropy of the classifier prediction. We, thus, compute an adversarial loss L_A, which is the negative entropy of the predicted distribution of water types from the nuisance classifier D, for the latent vector Z of the input image X. This adversarial loss is backpropagated to only to update the encoder E. The adversarial loss is given as

$$L_A(X) = \sum_{c=1}^{M} D(Z)_c \log D(Z)_c \qquad (5)$$

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

3.2. Training procedure

We first train only our encoder and decoder till a certain threshold, defined by the performance of the model on the validation set. We do this step to make sure that the encoder outputs an encoding Z with meaningful features before we include the nuisance classifier in our model. We then train our model by following a procedure which prioritizes the adversarial training of the encoder, while also making 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.

4. Experiments

We train our model on the synthetic underwater image dataset described in detail in section 4.2.1. The model is trained on a machine with the following configuration - Intel i7 6700 HQ processor, 8 GB RAM, NVIDIA GeForce GTX 960M 4GB graphics card.

4.1. Model architectures

We use the architecture of U-Net [17] for our encoder-decoder. U-Net is useful as the skip connections between

Algorithm 1: Training procedure of our model

> **Data:** Encoder E, decoder G and nuisance classifier
> $\qquad D$, $threshold_G \leftarrow 0.9$, $threshold_D \leftarrow 0.85$
> Get $val_G \leftarrow$ Cross validation SSIM score of G
> **while** $val_G < threshold_G$ **do**
> $\quad\lfloor$ Update E and G using L_R
> **for** n *training epochs* **do**
> $\quad$ **if** $val_G < threshold_G$ **then**
> $\quad\quad\mid$ Update E using L_R and L_A, G using L_R
> $\quad$ **else if** $val_D < threshold_D$ **then**
> $\quad\quad\mid$ Update D using L_N
> $\quad$ **else**
> $\quad\quad\lfloor$ Update E using L_R and L_A, G using L_R
> $\quad$ Get $val_G \leftarrow$ Cross validation SSIM score of G
> $\quad\lfloor$ Get $val_D \leftarrow$ Cross validation accuracy of D

encoder and decoder provide local and global information for decoder to generate clear images from. Also, it is a fully convolutional neural network which means it can handle images of varying sizes. Our nuisance classifier is a convolutional neural network which predicts probability of 6 classes. Its architecture can be seen in figure 3.
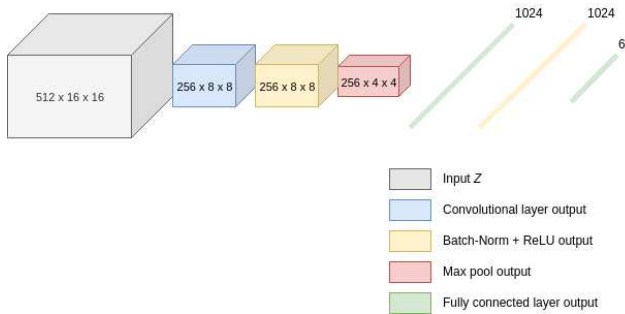

Figure 3. Our nuisance classifier architecture.

4.2. Datasets

Previous methods have tried to synthesize degraded underwater images from their clear versions. We train our model on such a synthesized dataset built using the method described in [3]. In order to see the usability of our model, we also test our model on a real-world dataset.

4.2.1 Synthetic underwater image dataset

We follow the approach mentioned in [3] to generate synthetic underwater images of multiple water types. The synthetic images are generated using the image formation model described by equations 1 and 2. We use the NYU-V2 RGB-D dataset [7] to provide us with the clear images as it also contains the depth information required to generate the corresponding synthetic images. We generate images for 6

Jerlov water types for each image in the dataset instead of generating images for 10 Jerlov water types. We combine similar image types 1 and 3, I, IA and IB and II and III from the 10 Jerlov water types to reduce the proximity between different water types. This boosts the nuisance classifier's performance as it is able to distinguish between different water types more easily. The images are synthesized using different values of N_c taken from [3] and random B_c and $d(x)$ values. For each image in the dataset, and for each of its 6 water types, we augment the dataset by generating 6 images with random B_c and $d(x)$ parameters. Thus, for each image in the dataset we have 36 corresponding underwater images of multiple water types. The synthesized 6 types of images for a given image can be seen in figure 4.

4.2.2 Real-world underwater image dataset

We use Underwater Image Enhancement Benchmark Dataset (UIEBD) built by [18] as our real-world underwater image dataset. The dataset consists of 890 underwater images.

4.3. Results on the synthetic dataset

4.3.1 Qualitative results

Figure 5 shows some visual results of our model on the test set of the synthetic underwater dataset which we synthesized in section 4.2.1. We can visually see that our model is successful in reconstructing the original color of the input images. The output images recover even the minute details from the degraded input images.

4.3.2 Quantitative results

We also compute quantitative evaluation metrics like SSIM [19] and PSNR for the generated images of different Jerlov water types [1] with respect to their clear counterparts. As seen in table 1, our model outperforms other methods for almost all water types.

4.4. Results on the real-world dataset

We also test our model on a real-world dataset to see the transferability of our model to different datasets. Figure 6 shows some visual results of our model on the Underwater Image Enhancement Benchmark Dataset [18]. 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 clear underwater images.

4.5. Comparison to no adversarial loss

We compare our model with vanilla U-Net without the adversarial loss. To see if we have learned the domain agnostic features, we plot the first two principal components

| Clear | 1,3 | 5 | 7 | 9 | I, IA, IB | II, III |

Figure 4. Underwater images synthesized following the approach in [3]. We club similar looking water types into a single class and reduce the total number of classes from 10 to 6 in order to boost the performance of our nuisance classifier.

	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 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 [3].

of the encoding Z from 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 9 and 10 respectively.

It can be seen from figures 9 and 10 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.

We also visually and quantitatively compare both the models. Figure 7 shows us the visual results of the models on both the synthetic underwater image dataset and the real-world UIEBD. Table 2 shows us the quantitative comparison.

We can see from both figure 7 and table 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 generates images with rich color quality than vanilla U-Net.

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

4.6. Object detection on enhanced images

As advocated by many previous works [20, 21, 22, 23, 24, 25, 26, 27, 28], the high-level computer vision performance (such as object detection) on enhanced images could

act as an indicator of the image enhancement performance itself. 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 [29] object
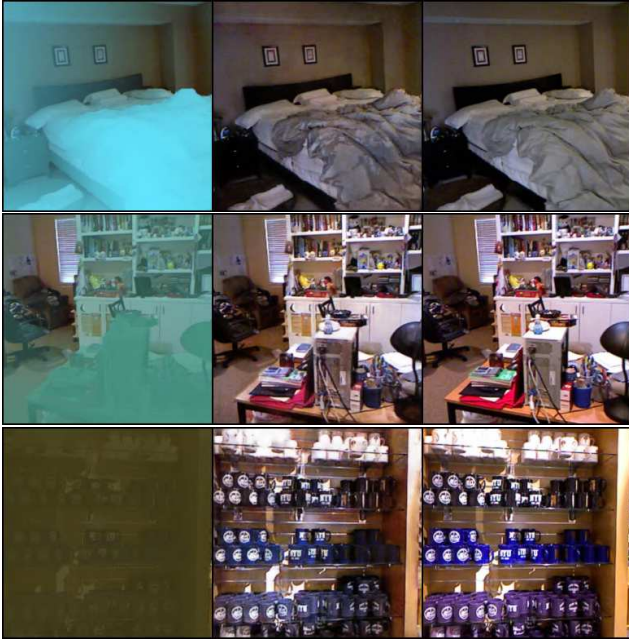
Figure 5. Results on the synthesized underwater dataset. Left column shows the input underwater images, middle column shows the results of our model and the right column shows the ground truth clear images.
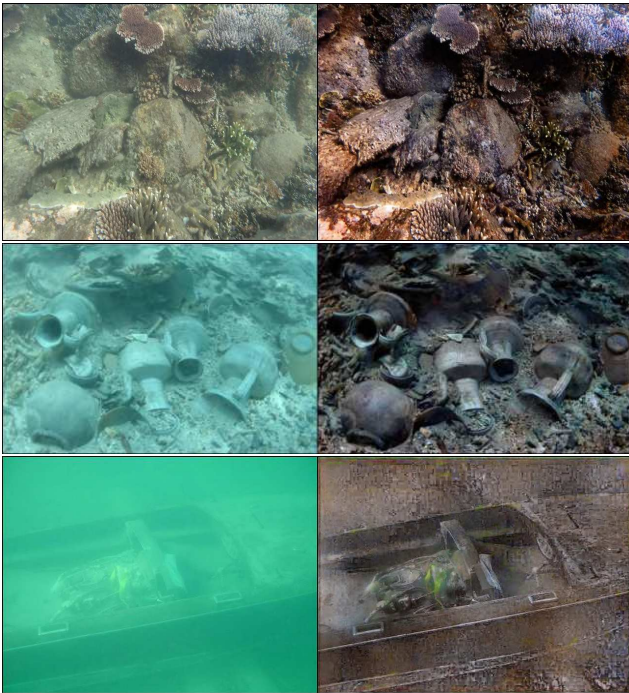

Figure 6. Results on the real-word dataset [18]. Left column shows the input underwater images and the right column shows the results of our model.

detector on the degraded underwater images and their enhanced versions generated by our model. We observe that

(a) From left to right - Input image, output of vanilla U-Net, output of U-Net with adversarial loss, ground truth image.

(b) From left to right - Input image, output of vanilla U-Net, output of U-Net with adversarial loss.

Figure 7. Comparison of U-Net with and without adversarial loss. (a) shows results on synthetic data, where as (b) shows results on real-world data.
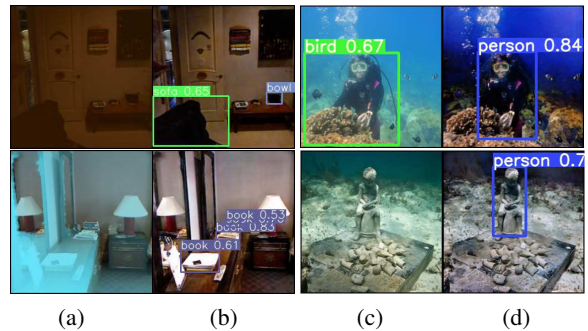

(a)   (b)   (c)   (d)

Figure 8. Object detection results before and after enhancement (a) Synthetic underwater image, (b) Output of our model for the synthetic underwater image, (c) Real-world underwater image and (d) Output of our model for the real-world image.

object detection is better on the images generated by our model compared to the degraded underwater images of the synthesized underwater dataset. However, we get mixed results when we run the object detector on the real-world UIEBD. Figure 8 shows the results of YOLO v3 before and after processing the images with our model.

5. Conclusion

We are able to provide a novel solution for underwater image enhancement which outperforms the previous methods both qualitatively and quantitatively. Our goal is to provide a generalized solution which could handle the diversity of the underwater images as well as transform them into clear images. Our model is successful in doing so by learning domain agnostic features for multiple underwater image types and then generating their clear version from those features. We also show that the model is able to generalize well on the unseen real-world data. Also, experimental
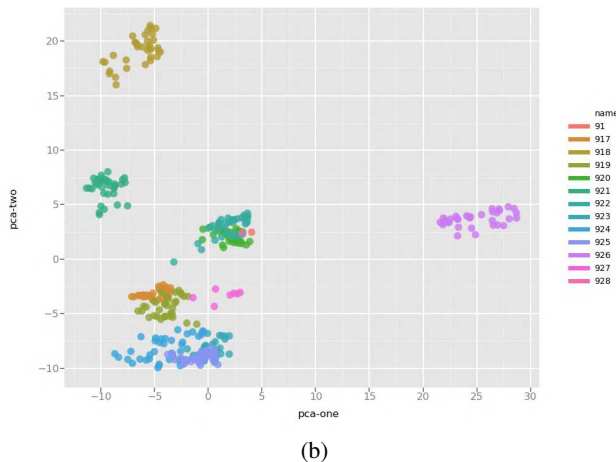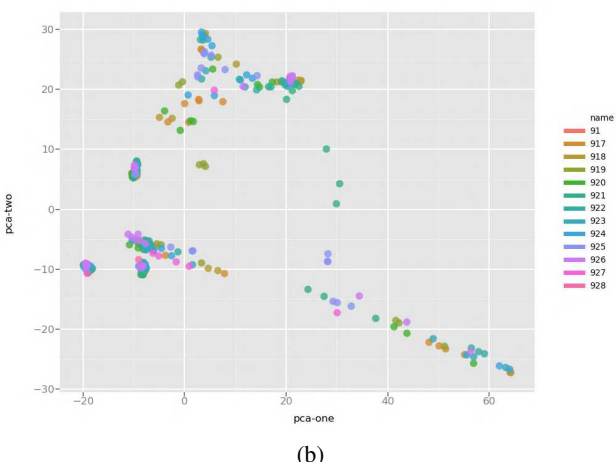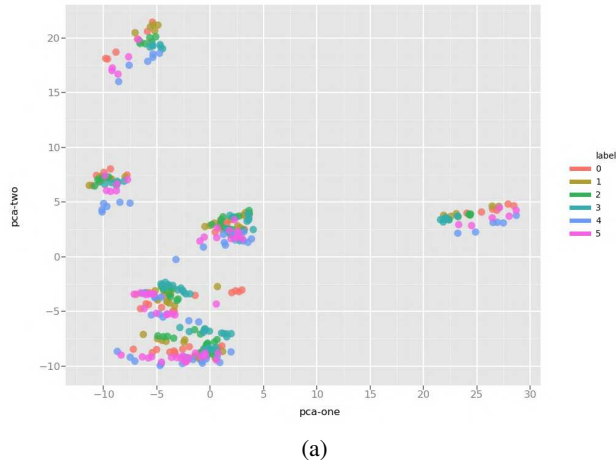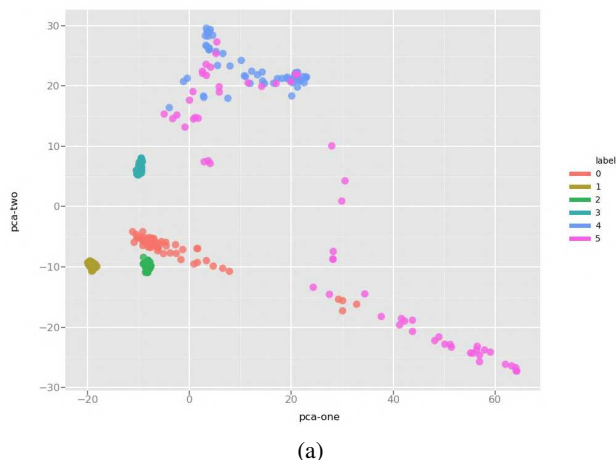
(a)

(b)

Figure 9. Visualizing first two PCA components of the encoding Z learned by U-Net without adversarial loss. (a) Colors points with the same water type, (b) Colors points with the same content.

(a)

(b)

Figure 10. Visualizing first two PCA components of the encoding Z learned by U-Net with adversarial loss (UIE-DAL). (a) Colors points with same water type, (b) Colors points with same content.

results on object detection task show that enhancing underwater images with our model before high level vision tasks improves the detection performance.

References

[1] N. Jerlov. *Marine Optics*. Elsevier, 1976. 1, 4

[2] D. Berman, T. Treibitz, and S. Avidan. Diving into haze-lines: Color restoration of underwater images. In *Proceedings of the British Machine Vision Conference*. BMVA Press, 2017. 1

[3] Saeed Anwar, Chongyi Li, and Fatih Porikli. Deep underwater image enhancement. *CoRR*, abs/1807.03528, 2018. 1, 2, 4, 5

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

[5] Sharks and scorpions? the world's deadliest animals aren't what you thought. https://www.dw.com/image/15773043_304.jpg. 1

[6] How to catch crappie in muddy water. https://www.reelchase.com/wp-content/uploads/2017/03/Learn-the-Best-Tips-on-How-to-Catch-Crappie-in-Muddy-Water.jpg. 1

[7] Pushmeet Kohli Nathan Silberman, Derek Hoiem and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012. 2, 4

[8] J. Y. Chiang and Y. Chen. Underwater image enhancement by wavelength compensation and dehazing. *IEEE Transactions on Image Processing*, 21(4):1756–1769, April 2012. 2

[9] Yann LeCun and Yoshua Bengio. The handbook of brain theory and neural networks. chapter Convolutional Networks for Images, Speech, and Time Series, pages 255–258. MIT Press, Cambridge, MA, USA, 1998. 2

[10] Zhenyu Wu, Zhangyang Wang, Zhaowen Wang, and Hailin Jin. Towards privacy-preserving visual recognition via ad-

versarial training: A pilot study. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 606–624, 2018. 2

[11] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2672–2680. Curran Associates, Inc., 2014. 2, 3

[12] A. Jordt. *Underwater 3d reconstruction based on physical models for refraction and underwater light propagation.* PhD thesis, 2013. 2

[13] J. S. Jaffe. Computer modeling and the design of optimal underwater imaging systems. *IEEE J. Oceanic Engin.*, 15:101–111, 1990. 2

[14] 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)*, pages 568–577, July 2017. 2

[15] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning.* MIT Press, 2016. http://www.deeplearningbook.org. 2

[16] Cameron Fabbri, Md Jahidul Islam, and Junaed Sattar. Enhancing underwater imagery using generative adversarial networks. *CoRR*, abs/1801.04011, 2018. 2

[17] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. *CoRR*, abs/1505.04597, 2015. 3

[18] Chongyi Li, Chunle Guo, Wenqi Ren, Runmin Cong, Junhui Hou, Sam Kwong, and Dacheng Tao. An underwater image enhancement benchmark dataset and beyond. *CoRR*, abs/1901.05495, 2019. 4, 6

[19] Zhou Wang, Alan C Bovik, Hamid R Sheikh, Eero P Simoncelli, et al. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004. 4

[20] Zhangyang Wang, Shiyu Chang, Yingzhen Yang, Ding Liu, and Thomas S Huang. Studying very low resolution recognition using deep networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4792–4800, 2016. 5

[21] Boyi Li, Xiulian Peng, Zhangyang Wang, Jizheng Xu, and Dan Feng. Aod-net: All-in-one dehazing network. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4770–4778, 2017. 5

[22] Ding Liu, Bihan Wen, Xianming Liu, Zhangyang Wang, and Thomas S Huang. When image denoising meets high-level vision tasks: A deep learning approach. *arXiv preprint arXiv:1706.04284*, 2017. 5

[23] Ding Liu, Bowen Cheng, Zhangyang Wang, Haichao Zhang, and Thomas S Huang. Enhance visual recognition under adverse conditions via deep networks. *arXiv preprint arXiv:1712.07732*, 2017. 5

[24] Boyi Li, Xiulian Peng, Zhangyang Wang, Jizheng Xu, and Dan Feng. End-to-end united video dehazing and detection. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018. 5

[25] Yu Liu, Guanlong Zhao, Boyuan Gong, Yang Li, Ritu Raj, Niraj Goel, Satya Kesav, Sandeep Gottimukkala, Zhangyang Wang, Wenqi Ren, et al. Improved techniques for learning to dehaze and beyond: A collective study. *arXiv preprint arXiv:1807.00202*, 2018. 5

[26] Boyi Li, Wenqi Ren, Dengpan Fu, Dacheng Tao, Dan Feng, Wenjun Zeng, and Zhangyang Wang. Benchmarking single-image dehazing and beyond. *IEEE Transactions on Image Processing*, 28(1):492–505, 2019. 5

[27] Rosaura G VidalMata, Sreya Banerjee, Brandon Richard-Webster, Michael Albright, Pedro Davalos, Scott McCloskey, Ben Miller, Asong Tambo, Sushobhan Ghosh, Sudarshan Nagesh, et al. Bridging the gap between computational photography and visual recognition. *arXiv preprint arXiv:1901.09482*, 2019. 5

[28] Siyuan Li, Iago Breno Araujo, Wenqi Ren, Zhangyang Wang, Eric K Tokuda, Roberto Hirata Junior, Roberto Cesar-Junior, Jiawan Zhang, Xiaojie Guo, and Xiaochun Cao. Single image deraining: A comprehensive benchmark analysis. *arXiv preprint arXiv:1903.08558*, 2019. 5

[29] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *CoRR*, abs/1804.02767, 2018. 5