The Synthesis of Unpaired Underwater Images for Monocular Underwater Depth Prediction

Qi Zhao [1#], Ziqiang Zheng [1#], Huimin Zeng, Zhibin Yu* [1,2], Haiyong Zheng,[1], Bing Zheng,[1,2]

[1] *No.238, Songling Road, Ocean University of China, Qingdao, Shandong, China*
[2] *Sanya Oceanographic Institution, Ocean University of China, Sanya, Hainan, China*

Correspondence*:
Zhibin Yu
Qi Zhao and Ziqiang Zheng contribute to this work equally.

ABSTRACT

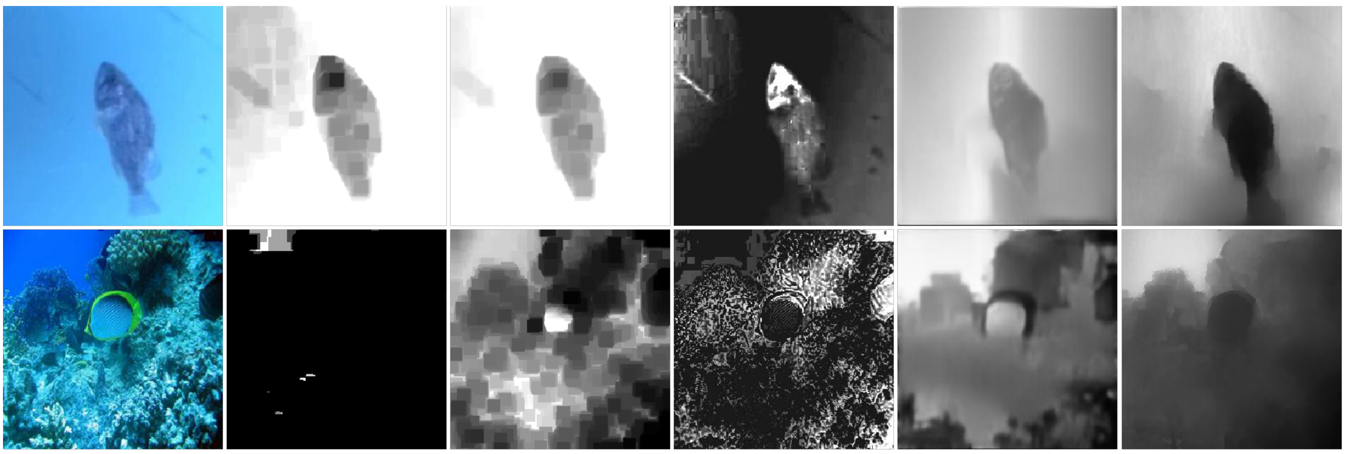
Underwater depth prediction plays an important role in underwater vision research. Because of the complex underwater environment, it is extremely difficult and expensive to obtain underwater datasets with reliable depth annotation. Thus, underwater depth map estimation with a data-driven manner is still a challenging task. To tackle this problem, we propose an end-to-end system including two different modules for underwater image synthesis and underwater depth map estimation, respectively. The former module aims to translate the hazy in-air RGB-D images to multi-style realistic synthetic underwater images while retaining the objects and the structural information of the input images. Then we construct a semi-real RGB-D underwater dataset using the synthesized underwater images and the original corresponding depth maps. We conduct supervised learning to perform depth estimation through the pseudo paired underwater RGB-D images. Comprehensive experiments have demonstrated that the proposed method can generate multiple realistic underwater images with high fidelity, which can be applied to enhance the performance of monocular underwater image depth estimation. Furthermore, the trained depth estimation model can be applied to real underwater image depth map estimation. We will release our codes and experimental setting in `https://github.com/ZHAOQIII/UW_depth`.

Keywords: Underwater vision, underwater depth map estimation, underwater image translation, generative adversarial network, image-to-image translation

1 INTRODUCTION

As an important part of underwater robotics and 3D reconstruction, underwater depth prediction is crucial for underwater vision research. However, the quality of collected images is restricted by light refraction and absorption, suspended particles in the water, and color distortion, making it difficult and challenging to obtain reliable underwater depth maps. Due to the influence of strong absorption and scattering, some widely used devices designed to obtain in-air depth maps, such as Kinect units (Dancu et al., 2014), lidar (Churnside et al., 2017) and binocular stereo cameras (Deris et al., 2017), exhibit limited performance in underwater environments (Massot-Campos and Oliver-Codina, 2015; Pérez et al., 2020). As quite a few underwater RGB-D datasets (Akkaynak and Treibitz, 2019) (Gomez Chavez et al., 2019) (Berman et al., 2020) are currently available, many researchers have sought to adopt image processing methods to estimate the depth from a single monocular underwater image or a consecutive underwater image sequence. To perform single monocular underwater depth prediction, several restoration-based methods have been developed (e.g. UDCP (Drews et al., 2016)) (Ueda et al., 2019). The transmission map is regarded as an intermediate step for obtaining depth maps and restoring underwater images. In theory, the physical process is highly dependent on the calibrated intrinsic parameters and the well-described structural information of the scene. However, it is extremely laborious to select and measure these parameters relevant to the physical process (Abas et al., 2019), and limited to some special task.

Recently, deep learning methods have shown great potential in image processing (Li et al., 2018) applications, such as image-to-image translation (Zhu et al., 2017a; Choi et al., 2018; Isola et al., 2017; Wang et al., 2018c; Zheng et al., 2020), image restoration (Peng et al., 2015) and depth estimation (Gupta and Mitra, 2019). Due to the lack of the underwater depth ground truth to formulate full supervision, supervised learning models cannot be directly adopted for underwater depth estimation. Due to the introduction of cycle-consistency loss designed for unpaired image-to-image translation, many researchers aim to translate the in-air images to the desired underwater images and preserve the original depth annotation (Li et al., 2017, 2018; Gupta and Mitra, 2019). With the synthetic underwater images from the original in-air images paired with the corresponding depth annotation, we can obtain the pseudo underwater and depth image pairs. Previous methods such as WaterGAN (Li et al., 2017) and UMGAN (Li et al., 2018) adopted a two-stage optimization framework for underwater depth estimation. The former underwater image synthesis and the downstream vision task (such as depth prediction or underwater image restoration) are optimized separately. The two models have no direct connection at the training stage. UW-Net (Gupta and Mitra, 2019) has addressed this problem and aims to perform underwater image synthesis and underwater depth estimation parallel. However, two competitive tasks with cycle-consistent learning lead to low training efficiency and inaccurate depth estimation outputs. The leakage of texture is another challenge. The depth value of a fish should be about equal. However, the bright color and textures of a fish may lead to an incorrect depth estimation result(Figure 1(b)-(e)).



(a) Input (b) DCP (c) UDCP (d) Berman *et al.* (e) UW-Net (f) Ours

Figure 1. Examples of texture leakage during the underwater depth map estimation process using different methods. (a)real underwater images. (b)DCP (He et al., 2010), (c)UDCP (Drews et al., 2016), (d) Berman *et al.* (Berman et al., 2017), (e) UW-Net (Gupta and Mitra, 2019), (f) ours.

To address these problems, we propose a novel joint-training generative adversarial network for both multi-style underwater image synthesis and depth estimation performed in an end-to-end manner. For the former image synthetic task, we aim to transfer the hazy in-air RGB-D images to multi-style underwater images while retaining the objects and the structural information of the in-air images and controlling the underwater style through one conditional input message. To take advantage of multi-task learning (Zhang and Yang, 2017) between underwater image synthetic and depth estimation tasks, we design a joint-training generator to estimate the depth from the synthesized underwater images through full supervision. Overall, our system includes two consecutive generators (responsible for the underwater image synthesis and underwater depth estimation, separately), which are trained simultaneously. To ensure that the generated underwater images retain the objects and the structural information of the in-air images, we consider perceptual loss (Johnson et al., 2016) computed at the selected layers as a structural loss along with the adversarial loss to optimize the whole network. Furthermore, we develop a depth loss to alleviate the texture leakage phenomenon as shown in Figure 1. Finally, we evaluate the effectiveness of our proposed method to synthesize underwater images and estimate the depth map of real underwater images, and the comprehensive experimental results demonstrate the superiority of the proposed method. Overall, our main contributions of this paper are summarized as follows:

- We propose a novel joint-training generative adversarial network, which can simultaneously handle the controllable translation from the hazy RGB-D images to the multi-style realistic underwater images by combining one additional label, and the depth prediction from both the synthetic and real underwater images.
- To construct a semi-real underwater RGB-D dataset, we take the hazy in-air RGB-D image pairs and conditional labels as inputs to synthesize multi-style underwater images. During the training process, we introduce perceptual loss to preserve the objects and structural information of the in-air images during the image-to-image translation process.
- To improve the results of underwater depth estimation, we design the depth loss to make better use of high-level and low-level information. We verify the effectiveness of our proposed method on a real underwater dataset.

2 RELATED WORK

2.1 Image-to-Image Translation

In the past several years, a series of image-to-image translation methods based on generative adversarial networks (GANs) (Odena et al., 2017; Mirza and Osindero, 2014) have been proposed. These approaches can mainly be divided into two categories of paired training and unpaired training methods. Pix2pix (Isola et al., 2017) is a typical powerful paired model and first proposes cGAN (Mirza and Osindero, 2014) learns the one-side mapping function from the input images to target images. To achieve the image-to-image translation of unpaired datasets, CycleGAN (Zhu et al., 2017a) translates images into two domains using two generators and two discriminators and proposes the cycle-consistent loss to tackle the mode collapse of unpaired image translation. To address the multimodal problem, methods including BicycleGAN (Zhu et al., 2017b), MUNIT (Huang et al., 2018), DRIT (Lee et al., 2018), StarGAN (Choi et al., 2018), etc. have been proposed. The BicycleGAN (Zhu et al., 2017b) learns to transfer the given input with a low-dimensional latent code to more diverse results. It takes advantage of the bijective consistency between the latent and target spaces to avoid the mode collapse problem. MUNIT (Huang et al., 2018) achieves multidomain translation by assuming two latent representations that present style and content respectively and combining different representations of content and style. StarGAN Choi et al. (2018) learns multiple mapping functions

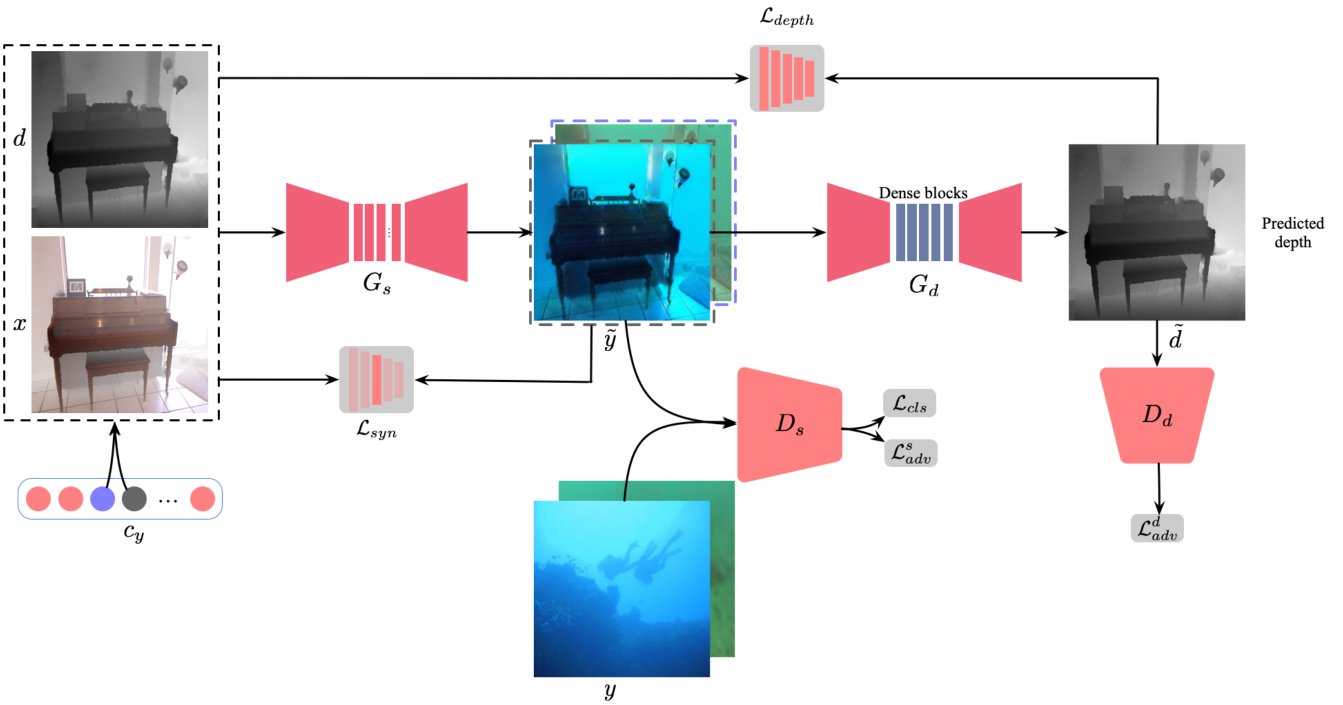


Figure 2. The network framework of our proposed model is designed to synthesize multi-style underwater images and estimate underwater depth maps. The generator G_s and the discriminator D_s are used to synthesize multi-style underwater images, and the generator G_d and discriminator D_d learn to estimate underwater depth map based on the synthesized underwater RGB-D dataset.

between multiple domains. It only uses a single generator and a discriminator to transfers the source images to the target domain. Then to avoid mode collapse, the generator takes the generated images and the original labels as input and transfers them to the original domain. The subsequently developed image-to-image translation methods, such as pix2pixHD (Wang et al., 2018c), GauGAN (Park et al., 2019), vid2vid (Wang et al., 2018b), FUNIT (Liu et al., 2019), NICE-GAN (Chen et al., 2020) and StarGAN v2 (Choi et al., 2020) pay more attention to generate higher visual quality, multiple outputs and have been applied in video and small sample studies.

To synthesize underwater images, due to the lack of a large paired underwater image dataset, studies have mainly focused on unsupervised learning. In a pioneering approach of underwater image synthesis, WaterGAN (Li et al., 2017) synthesized the underwater images from the in-air image and the paired depth map for real-time color correction of monocular underwater images. To achieve multidomain translation, UMGAN (Li et al., 2018) proposes an unsupervised method that combines CycleGAN (Zhu et al., 2017a) and cGAN (Mirza and Osindero, 2014) with an additional style classifier to synthesize multi-style underwater images. UW-Net developed by Gupta *et al.* (Gupta and Mitra, 2019) learns the mapping functions between unpaired hazy RGB-D images and arbitrary underwater images to synthesize underwater images and estimate the underwater depth map. This method translates the hazy RGB-D image to underwater images while it learns to convert underwater images to the hazy RGB-D images. However, WaterGAN (Li et al., 2017) and UW-Net (Gupta and Mitra, 2019) only provide a solution for single domain underwater image generation. UMGAN (Li et al., 2018) does not consider the transmission map as an extra clue to generate underwater images. Moreover, all of the synthesized underwater images using these methods still lack the characteristics of real underwater images and clear structural information.

2.2 Underwater Depth Map Estimation

Underwater depth map estimation has mainly been studied in the field of traditional image processing. Since He *et al.* (He et al., 2010) first proposed a dark channel prior (DCP) for dehazing, many methods based on DCP (He et al., 2010) have been proposed for underwater depth map estimation in recent years. Drews *et al.* (Drews et al., 2016) proposed a method based on a physical model of light propagation and the statistical priors of the scene to obtain the medium transmission and scene depth in typical underwater scenarios. Peng *et al.* (Peng et al., 2015) proposed a three-step approach consisting of pixel blurriness estimation, rough depth map generation, and depth map refinement for depth map estimation. Berman *et al.* (Berman et al., 2017) took different optical underwater types into account and proposed a more comprehensive physical image formation model to recover the distance maps and object colors. They mainly considered transmission map estimation as an intermediate step to obtain a depth map. Due to the unknown scattering parameters and multiple possible solutions, the results of these methods are most likely to be incorrect (Gupta and Mitra, 2019).

Recently, many deep learning-based methods have been proposed for depth estimation. However, most of these approaches focus on depth estimation from in-air RGB images with full supervision, which are not suitable for underwater depth map estimation due to the lack of the paired RGB-D data. The above mentioned UW-Net developed by Gupta *et al.* (Gupta and Mitra, 2019) proposed an unsupervised method to learn depth map estimation. It considers an in-air transmission map as a cue to synthesize underwater images and obtains the required depth map from the synthesized underwater images. However, this method cannot estimate the depth map from underwater images of multiple water types. Because two competitive tasks (hazy in-air image reconstruction and depth estimation) are assigned to one generator, the depth prediction results of UW-Net lack sharp outlines. Ye *et al.* proposed another unsupervised adaptation networks Ye et al. (2019). They developed a joint learning framework which can handle underwater depth estimation and color correction tasks simultaneously. Unlike their work, in which the two networks (style adaptation network and task network) should be trained separately, our model is more simple and can be trained simultaneously. The depth loss and a fine-tune strategy make our model more efficient in practice for underwater depth map prediction.

3 MATERIALS AND METHODS

3.1 Overall Framework

In this paper, we aim to estimate the depth map from real underwater images. Because there are no paired underwater RGB-D images, we cannot perform supervised learning directly. Therefore, we choose to translate the original in-air images with corresponding depth to underwater images and obtain pseudo-paired images. To perform this task, we design an end-to-end system with two joint-training modules: **multi-style underwater image synthesis** and **underwater depth estimation** based on the synthetic paired samples. The former module is trained through unpaired training, while the latter adopts supervised training to achieve precise underwater depth estimation. The overall framework is shown in Figure 2 and consists of two generators, namely, G_s: $x \rightarrow \tilde{y}$ and G_d: $\tilde{y} \rightarrow \tilde{d}$, where x and $\tilde{y}$ are the original in-air image and the synthesized underwater image with specific underwater style. $\tilde{d}$ is the estimated depth output. For discrimination, we also design two discriminators D_s and D_d to perform adversarial training to boost the underwater image synthesis and depth estimation, respectively. D_s aims to distinguish between real and fake images and identify the domains from which both the real images and the generated images originate. The discriminator D_d only learns to distinguish between the real and fake depth maps.

multi-style underwater image synthesis. As shown in Figure 2, we refer to the training of StarGAN (Choi et al., 2018) to generate multi-style underwater images. To synthesize specified underwater style images, we adopt an additional one-hot vector c to represent domain attributes. To make the generator G_s depth-aware and preserve the original depth representation after translation, we concatenate the three inputs, namely, the in-air image (x), the target underwater style (c_y), and the corresponding in-air depth (d) to synthesize an underwater image $\tilde{y} = G_s(\mathcal{C}(x, d, c_y))$ with the required style (c_y), where $\mathcal{C}$ denotes depthwise concatenation. To guarantee that the synthetic image $\tilde{y}$ has the target underwater style, we include an adversarial domain classifier D_s with two branches (one for domain classification and another for real/fake discrimination). The classification branch with the domain classification loss $\mathcal{L}_{cls}$ aims to recognize the underwater style (c_y) of both the synthesized image $\tilde{y}$ and the real underwater image y. Noted that y does not have the corresponding depth annotation due to the lack of underwater ground truth. The adversarial loss $\mathcal{L}^s_{adv}$ is computed to promote the naturalness of the synthetic images. The generator G_s from CycleGAN (Zhu et al., 2017a) and StarGAN (Choi et al., 2018) is one symmetric encoder-decoder architecture with 6 residual blocks.

Underwater depth estimation. In the training stage, we perform underwater estimation on the above-mentioned synthetic underwater images $\tilde{y}$ by adopting a generator G_d with dense-block architectures. The output of generator G_s $(\tilde{y})$ is the input of generator G_d used to estimate its depth map $G_d(\tilde{y})$. Considering that we have the depth annotation d of the in-air images, we can obtain pseudo pairs to compute the $\mathcal{L}_{depth}$ between d and $\tilde{d}$. The discriminator D_d is also designed and has only one discrimination output. Furthermore, the adversarial loss $\mathcal{L}^d_{adv}$ in the depth space is conducted. For underwater depth map estimation, we use DenseNet (Jégou et al., 2017) as the generator. In UW-Net (Gupta and Mitra, 2019), the authors proved the importance of using hazy above-water images and compared the results of underwater depth maps estimation with different generator networks, including ResNet (He et al., 2016), Unet (Ronneberger et al., 2015), DenseNet (Jégou et al., 2017) and so on. In their work, DenseNet is proved to be the best choice.

3.2 Loss Functions

3.2.1 multi-style underwater image synthesis

Adversarial Loss. Regular GANs use sigmoid activation output and the cross-entropy loss function (Goodfellow et al., 2014), which may cause a vanishing gradient during the learning process. To stabilize the training process and generate underwater images with higher quality, we adapt the least-squares loss (Mao et al., 2017) in our method. $\mathcal{L}^s_{adv}$ can be expressed as follows:

$$\begin{aligned} \mathcal{L}^s_{adv} = \underset{G}{min}\, \underset{D}{max}\{\mathbb{E}x, y \sim P_{dta}(x, y)[(D_s(y) - 1)^2] \\ + \mathbb{E}_{x \sim P_{data}(x)}[(D_s(\tilde{y})^2]\}, \\ where \quad \tilde{y} = G_s(\mathcal{C}(x, d, c_y))), \end{aligned} \tag{1}$$

where G_s targets the transfer of a hazy in-air RGB-D image x by concatenating an underwater condition label c_y to synthesize image $G_s(\mathcal{C}(x, d, c_y))$. The discriminator D_s attempts to distinguish the real underwater image y and the synthesized underwater image $\tilde{y}$.

Domain Classification Loss. For the given hazy in-air image x and an underwater domain style c_y, G_s translates x into an underwater image $\tilde{y}$, which can be properly classified to the desired target domain by D_s. To achieve this goal, the classification branch of D_s imposes the domain classification. For the real underwater image y, the domain classification loss $\mathcal{L}_{cls}^{r}$ is computed as:

$$\mathcal{L}_{cls}^{r} = \mathbb{E}_{y,c_y}[-\log D_s(c_y|y)]. \quad (2)$$

where the term $D_s(c_y|y)$ denotes a probability distribution over the underwater domain labels (c_y) computed by D_s. By minimizing this objective, D_s learns to classify an underwater image y to its original domain c_y. We assume that the underwater image and domain label pair (y, c_y) is given by the training data. For generator G_s, the loss function for the domain classification of synthetic underwater images is defined as:

$$\mathcal{L}_{cls}^{f} = \mathbb{E}_{\tilde{y},c_y}[-\log D_s(c_y|\tilde{y})]. \quad (3)$$

During the training, G_s tries to synthesize underwater image $\tilde{y}$ that can fool the classification branch of D_s.

Feature-level loss. Beyond the pixel-level loss, we design feature-level loss functions between the feature representations extracted from a pre-trained VGG19 network. The hybrid feature-level loss can effectively preserve the similarity of the object between the hazy in-air images and the synthesized underwater images. For the multi-style underwater image synthesis, we introduce a perceptual loss, namely, $\mathcal{L}_{syn}$. $\mathcal{L}_{syn}$ is designed to preserve the object content and loosen the restrictions on the color and textile changes after translation. $\mathcal{L}_{syn}$ is expressed as follows:

$$\mathcal{L}_{syn} = [||\Phi^{(i)}(x) - \Phi^{(i)}(G_s(x|c_y))||_1]. \quad (4)$$

where $\Phi^{(i)}$ denotes the parameters at the i-th layer of a pre-trained VGG19 network. Following the work by Kupyn *et al.* (Kupyn et al., 2019), we compute the 1-norm distance at the same selected $i = 14$ layer of the VGG19 network between the hazy in-air images and the synthesized underwater images.

Reconstruction Loss. To perform unpaired training between in-air and underwater images, we include the cycle consistency loss (Zhu et al., 2017a) in our framework. The reconstruction loss $\mathcal{L}_{rec}$ between $\hat{x}$ and x is defined as follows:

$$\begin{aligned} \mathcal{L}_{rec} &= \mathbb{E}_{x,c_y,c_x}[||x - \hat{x}||_1], \\ \hat{x} &= G_s(\mathcal{C}(G_s(\mathcal{C}(x, d, c_y)), d, c_x)), \end{aligned} \quad (5)$$

where c_x and c_y indicate the original hazy in-air domain label and the target underwater domain style, respectively. G_s takes the counterpart G_s(x$|c_y$), its corresponding depth, and the original domain label c_x as input and tries to reconstruct the original hazy in-air image. We adapt the L1 loss as our reconstruction loss. Note that we use the generator G_s twice, first to translate the hazy in-air RGB-D images into an underwater image in the target domain and then to reconstruct the hazy in-air RGB images from the translated images.

3.2.2 Underwater depth estimation

Adversarial Loss. For the second underwater depth estimation procedure, the adversarial loss $\mathcal{L}_{adv}^{d}$ is described as:

$$\begin{aligned} \mathcal{L}_{adv}^{d} = &\min_{G} \max_{D}\{\mathbb{E}_{G_s(\tilde{y}),d\sim P_{data}(\tilde{y},d)}[(D_d(d) - 1)^2] \\ &+ \mathbb{E}_{\tilde{y}\sim P_{data}(\tilde{y})}[(D_d(\tilde{d}))^2]\}, \\ &where \quad \tilde{d} = G_d(G_s(\mathcal{C}(x, d, c_y))), \end{aligned} \quad (6)$$

where G_d learns the mapping function from the synthesized underwater images $\tilde{y}$ to the in-air depth d as $G_d(\tilde{y}) \rightarrow d$. D_d is responsible to recognize the fake ingredient from the synthesized depth output $\tilde{d}$.

Depth loss. For underwater depth estimation, the pixel-level distance between the estimated value and the ground truth, such as 1-norm and 2-norm, is generally adopted to favor less blurring. However, we find that only the pixel-level loss between the predicted depth map and the ground truth often leads to poor performance due to the influences of noise, water with various turbidity, etc (Please refer to section 4.3 for more details). To force the model to pay more attention to the objects, we make use of the feature representations extracted from a pre-trained VGG19 network for multi-level information. We also introduce pixel-level distance for low-level details. Finally, to obtain improved results, we combine 1-norm loss and the multi-layer feature constraint between $\tilde{d}$ and d and define the depth loss, namely $\mathcal{L}_{depth}$:

$$\mathcal{L}_{depth} = [||d - G_d(G_s(x|c_y))||_1] + \sum_{i=0}^{N}[||\Phi^{(i)}(d) - \Phi^{(i)}(G_d(G_s(x|c_y)))||_1]. \tag{7}$$

Similarly, $\Phi^{(i)}$ represents the pre-trained parameter of the i-th layer. Here, following the work of Wang *et al.* (Wang et al., 2018c) and Wang *et al.* (Wang et al., 2018a), we compute the L1 distance at the same selected 6 layers: $i = 1, 6, 11, 20, 29$.

3.3 Full Objective

Finally, the objective functions can be written, respectively, as:

$$\mathcal{L}_{D_s} = \mathcal{L}^s_{adv} + \alpha\mathcal{L}^r_{cls} \tag{8}$$

$$\mathcal{L}_{G_s} = \mathcal{L}^s_{adv} + \gamma\mathcal{L}_{rec} + \alpha\mathcal{L}^f_{cls} + \lambda\mathcal{L}_{syn} \tag{9}$$

$$\mathcal{L}_{D_d} = \mathcal{L}^d_{adv} \tag{10}$$

$$\mathcal{L}_{G_d} = \mathcal{L}^d_{adv} + \eta\mathcal{L}_{depth} \tag{11}$$

where α, γ, λ and η are the hyperparameters that control the effect of each loss in the final objective function. We set $\alpha = 5, \gamma = 10, \lambda = 0.1, \eta = 50$ in all of our experiments, and we optimize the objective function with the Adam optimizer (Kingma and Ba, 2014). To choose appropriate weights, we design ablation studies for each hyperparameter except for γ. We follow StarGAN (Choi et al., 2018) to set γ = 10. For the choice of the rest of hyperparameters, please refer to Sec. 4.3 for more details.

4 RESULTS

4.1 Datasets and Implementation Details

In our experiments, we translate the hazy in-air images to two underwater domains (*green and blue*). We also choose the hazy in-air D-Hazy dataset (Ancuti et al., 2016) as the input images; this dataset contains the indoor scenes. For the two underwater domains, we adapt the real underwater images from the SUN (Xiao et al., 2010), URPC [1], EUVP (Islam et al., 2020), UIEB (Li et al., 2019) and Fish datasets [2]. We collect 1,031 blue and 1,004 green underwater images from these datasets and the Google website, respectively. The D-Hazy dataset (Ancuti et al., 2016) includes 1,449 images. We randomly choose 1,300 images as the in-air images x to train the model. The remaining 149 images of the dataset are selected for evaluation.

[1] `http://www.cnurpc.org/`

[2] `http://www.fishdb.co.uk/`

We use random-crop to obtain 128×128 patches for training. For the evaluation stage, we take complete images of 256×256. The entire network is trained on one Nvidia GeForce GTX 1070 using the Pytorch framework. To avoid the mode collapse problem, we apply spectral normalization (Miyato et al., 2018) in both the discriminators and the generators. Because of the introduction of spectral normalization (Miyato et al., 2018), we use a two-timescale update rule (TTUR) based on BigGAN (Brock et al., 2018) and SAGAN (Zhang et al., 2018). The Adam algorithm is applied with a learning rate of 0.0002 for the discriminators while 0.00005 for the generators. Because of the limited computing resources, we set the batch size to 10 and perform 100,000 training iterations in our experiments.

4.2 Comparison Methods

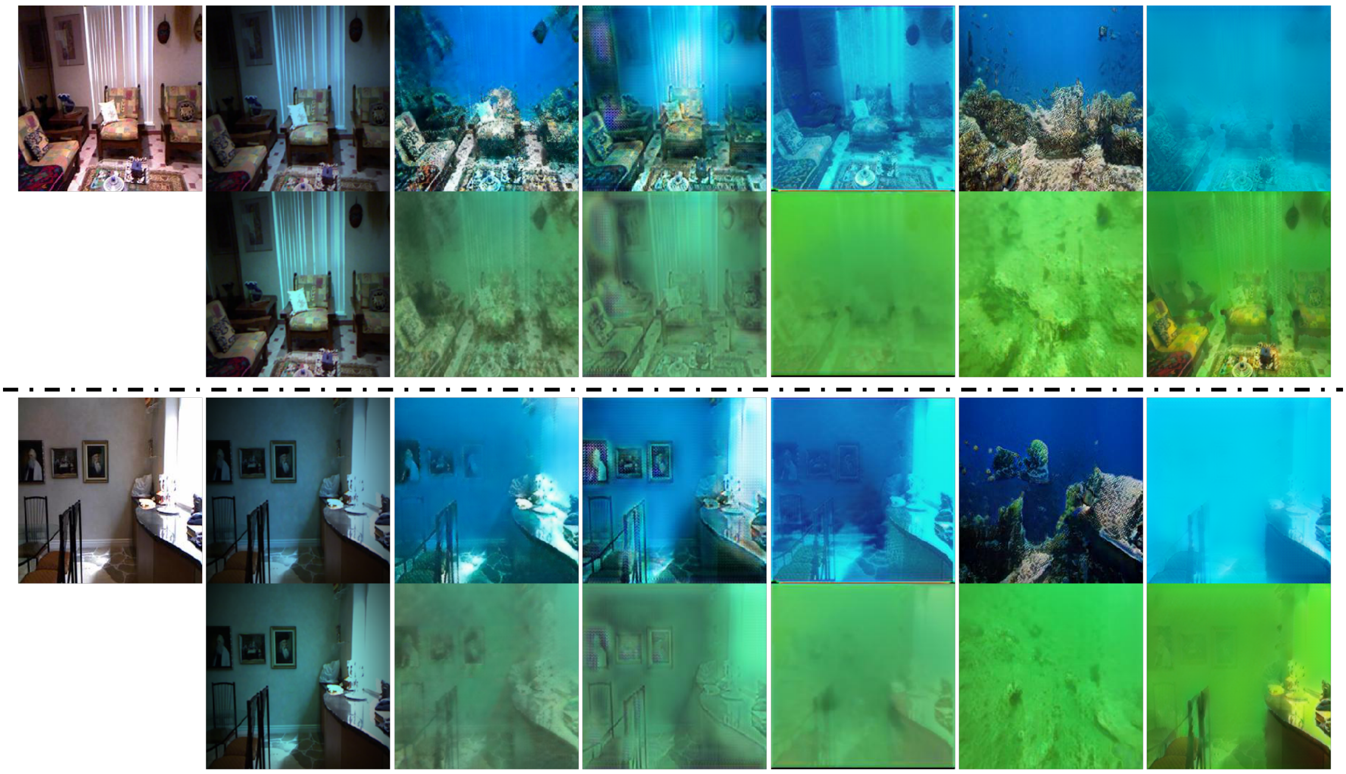
Our method achieves underwater depth map estimation using multi-style synthesized underwater images. In this section, we first evaluate the performance of WaterGAN (Li et al., 2017), CycleGAN (Zhu et al., 2017a), StarGAN (Choi et al., 2018), UW-Net (Gupta and Mitra, 2019), StarGAN v2 (Choi et al., 2020) and our method on multiple synthetic underwater images. Additionally, to evaluate the effectiveness of underwater depth map estimation, we compare the results obtained using DCP (He et al., 2010), UDCP (Drews et al., 2016), Berman *et al.* (Berman et al., 2017), Gupta *et al.* (Gupta and Mitra, 2019) and our method.

4.2.1 Qualitative Evaluation

To evaluate the effectiveness of the proposed method, we perform underwater image synthesis on the NYUv2 (Silberman et al., 2012) and D-Hazy (Ancuti et al., 2016) datasets. Figure 3 shows a visual comparison of the synthesized underwater images generated by different methods. WaterGAN (Li et al., 2017) takes advantage of in-air RGB-D images to synthesize underwater images. As shown in Figure 3(b), the results are somewhat single-hued and lack water characteristics. Although WaterGAN supports multi-style image generation, the two styles (blue and green) obtained by WaterGAN in Figure 3(b) are difficult to distinguish. The results of CycleGAN (Zhu et al., 2017a) retain most of the contents and structures of the original images. Compared to WaterGAN, they are similar to the natural underwater scenes shown in Figure 3(c). By contrast, the outputs of CycleGAN (Zhu et al., 2017a) include serious distortions of the details of the image with incorrect depth information. StarGAN (Choi et al., 2018) can simultaneously translate in-air images into multiple underwater styles. However, the results lack the characteristics of real underwater images, such as depth information, and clear structural information of the objects. Besides, many artifacts are observed in Figure 3(d). UW-Net (Gupta and Mitra, 2019) also takes hazy in-air RGB-D images as input, the results are presented in Figure 3(e) and show fuzzy structures for the objects. The results of StarGAN v2 (Choi et al., 2020) are shown in Figure 3(f). There is no denying that StarGAN v2 (Choi et al., 2020) possesses a powerful style network to extract style codes from reference images. However, the underwater images provided by StarGAN v2 fail to help the depth estimation tasks. As shown in Figure 3(f), StarGAN v2 removed some objects and structural information during the image synthetic process, which makes the synthetic underwater images and their corresponding in-air depth maps unmatched. The quantitative results in section 4.2.2 further confirm this point.

Our model is optimized to synthesize underwater images with multiple styles based on the unpaired datasets. The results of our method (Figure 3(g)), in which the structural information is well preserved, are better than those obtained from other methods in terms of visual quality.

For underwater depth map estimation, Figure 4 shows the results of our method and other methods developed by He *et al.* (DCP) (He et al., 2010), Drews *et al.* (UDCP) (Drews et al., 2016), Berman *et al.* (Berman et al., 2017) and Gupta *et al.* (Gupta and Mitra, 2019) based on the underwater images obtained by Berman *et al.* (Berman et al., 2017). In Figure 4(b)-4(d), these methods fail to capture relative depth of



(a) In-air (b) WaterGAN (c) CycleGAN (d) StarGAN (e) UW-Net (f) StarGAN v2 (g) Ours

Figure 3. Comparison of the visual quality of synthesized underwater images obtained by different methods. From left to right, (a) are original in-air images, (b)–(g) are the results of the WaterGAN (Li et al., 2017), CycleGAN (Zhu et al., 2017a), StarGAN (Choi et al., 2018), UW-Net (Gupta and Mitra, 2019), StarGAN v2 (Choi et al., 2020) and our method.

the scene with respect to the camera. Moreover, these methods mainly obtain the transmission maps of the scene and have excessive texture leakage in the results. Gupta *et al.* (Gupta and Mitra, 2019) used an unsupervised method to estimate the depth map, obtaining the results shown in Figure 4(e), and this method appears to be better than the other methods, whose results are presented in Figure 4(b)-4(d). However, this method still suffers from excessive texture leakage and only estimates the depth map for single-domain underwater images. Our results have a much more reasonable appearance with a linear depth variation. On the other hand, we observe that our network successfully captures the depth information from multi-style underwater images. More results for real underwater images with different underwater characteristics are seen in Figure 5. Furthermore, the UW-Net (Gupta and Mitra, 2019) and our method synthesize underwater images using the underwater dataset provided by Berman *et al.* (Berman et al., 2017) to fine-tune the models of the depth map estimation. We fine-tune our model for 10,000 iterations on Berman *et al.*'s dataset for better depth map estimation.

4.2.2 Quantitative Evaluation

The dataset of Berman *et al.* (Berman et al., 2017) consists of 114 paired underwater RGB-D images from Katzaa, Michmoret, Nachsholim, and Satil. We use 71 images belonging to the three regions Katzaa, Nachsholim, and Satil. Because the Michmoret region has very few natural objects and is of the same scene. Following UW-Net (Gupta and Mitra, 2019), we use two metrics for comparison, namely, log scale-invariant mean squared error (SI-MSE) (Eigen et al., 2014) and the Pearson correlation coefficient (ρ). Considering the fact that the depth map provided by the stereo camera is not complete (e.g. the ground

(a) Input (b) DCP (c) UDCP (d) Berman *et al.* (e) UW-Net (f) Ours (g) GT

Figure 4. Comparison of our method with other underwater depth estimation methods. From left to right, (a) are real underwater images from the dataset of Berman *et al.* (Berman et al., 2017), (b)–(f) are the results of DCP (He et al., 2010), UDCP (Drews et al., 2016), Berman *et al.* (Berman et al., 2017), Gupta *et al.* (Gupta and Mitra, 2019) and our method, and (g) are the ground truths.

truth of the white regions in Figure 7(h) are not provided), we only calculate the pixels with a defined depth-value in the ground truth (GT).

The underwater image synthesis assists to estimate depth maps from real underwater images. Thus, how much the synthetic underwater images can be used to boost the performance of underwater image-based depth prediction is the key evaluation index. We evaluate performance on depth prediction tasks with a series of the state-of-the-art methods, which consist of WaterGAN (Li et al., 2017), CycleGAN (Zhu et al., 2017a), StarGAN (Choi et al., 2018), UW-Net (Gupta and Mitra, 2019) and StarGAN v2 (Choi et al., 2020). We aim to calculate the depth map estimation results on a semi-real underwater RGB-D dataset. UW-Net suggests that fine-tuning the models with a few unlabeled images from the target underwater environment could further boost the depth prediction performance. During the fine-tuning process, we only use the RGB underwater images without considering the depth ground truth of the data from Berman et al. to show the ability that our model can adapt itself to a new environment well. To make it fair, we fine-tune all models to generate a similar underwater style of the dataset of Berman *et al.*.

Although our model already provides a solution for a depth estimation task, we choose a typical independent supervised image-to-image model, pix2pix (Isola et al., 2017), to fairly evaluate the potential of synthetic underwater images on the application of depth prediction. We use identical pix2pix models to learn the mapping function between the generate underwater images of different underwater image synthetic methods and their corresponding in-air depth maps. Finally, we test and evaluate all models on the dataset of Berman *et al.*. Table 1 shows the results, and our model obtains higher ρ values and lower SI-MSE.

For the underwater depth estimation task, Table 2 shows the quantitative results. Our method obtains the least scale-invariant error (SI-MSE) (Eigen et al., 2014) and the highest Pearson correlation coefficient (ρ).

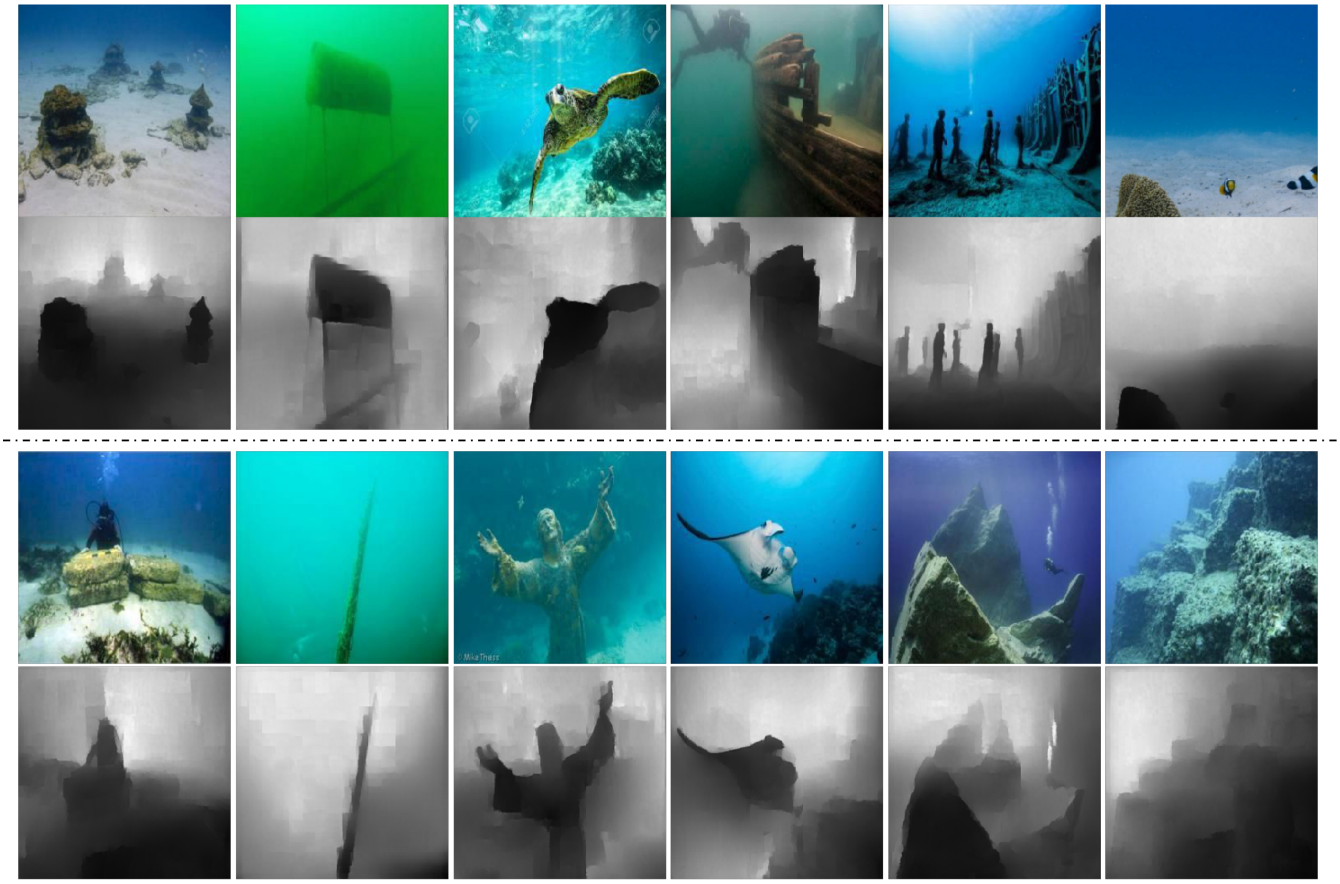


Figure 5. The results of our model for depth map estimation. Every two rows from top to bottom are real underwater images with different illumination and scattering conditions and the results of our model for depth map estimation.

We also investigate the parameters and Floating Point Operations Tan and Le (2019) (FLOPs) among different generators in Table 3. In the case of CycleGAN, we only count the FLOPs and parameters of a single generator. We can find that the proposed method can achieve better performance with fewer network parameters and computational cost. Benefiting from the dense blocks, the G_d of our model has fewer parameters and FLOPs than G_s. Please note that G_s is only used in training stage. In testing phase, we only need G_d to estimate the depth map.

Table 1. Quantitative comparison of our method and other methods for underwater image synthesis. We evaluate all models for underwater depth map estimation using the generated RGB-D datasets. FT represents a fine-tuned (FT) underwater model on the dataset of Berman *et al.* (Berman et al., 2017). Higher ρ values and lower SI-MSE (Eigen et al., 2014) values represent a better result.

ine	WaterGAN (FT)	CycleGAN (FT)	StarGAN (FT)	UW-Net (FT)	StarGAN v2 (FT)	Our (FT)
ine SI-MSE	0.5994	0.3514	0.4597	0.3594	0.5454	**0.2709**
ine ρ	0.5031	0.6024	0.5339	0.5795	0.4561	**0.6917**
ine						

Table 2. Quantitative comparison of our method and other methods on the dataset of Berman *et al.* (Berman et al., 2017). FT represents a fine-tuned (FT) underwater model. Higher ρ values and lower SI-MSE (Eigen et al., 2014) values represent a better result.

ine	DCP	UDCP	Berman *et al.*	UW-Net(FT)	Ours(FT)
ine SI-MSE	1.3618	0.6966	0.6755	0.3708	**0.1771**
ine ρ	0.2968	0.4894	0.6448	0.6451	**0.7796**
ine					

4.3 Ablation Study

4.3.1 Loss Selection of Underwater image Synthesis

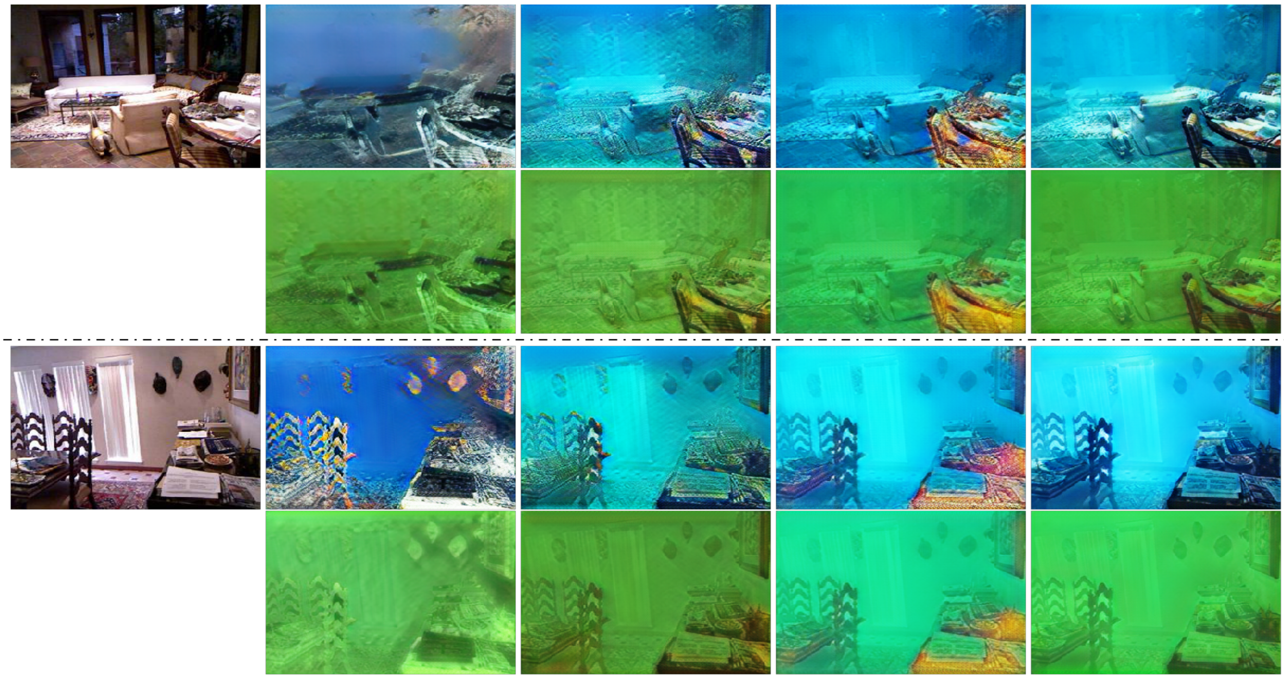
To preserve clear structural information, we consider the perceptual loss $\mathcal{L}_{syn}$, structural similarity index (SSIM) $\mathcal{L}_{ssim}$, and multiscale structural similarity index (MS-SSIM) $\mathcal{L}_{msssim}$ as the structural loss. We evaluate the efficiency of each loss, including $\mathcal{L}_{syn}$, $\mathcal{L}_{ssim}$ and $\mathcal{L}_{msssim}$, and based on the visual effect of the synthesized underwater images and the results of depth map estimation, we choose the perceptual loss. To verify the effectiveness of the extra losses in our network, we design ablation experiments and perform a comparison on D-Hazy (Ancuti et al., 2016) which consists of 1449 images. Figure 6 shows that each loss affects the quality of the generated underwater images. It is observed from Figure 6(b), that the generated underwater images using ResNet without any extra loss have more color blocks and artifacts. Additionally, during the training, it is extremely unstable and tends to produce color inversions and serious distortions situations. In Figure 6(c)$-$ Figure 6(d), many artifacts are still retained for ResNet with $\mathcal{L}_{ssim}$ or $\mathcal{L}_{msssim}$. Table 4 shows the results of depth map estimation based on different synthetic underwater image datasets, which are generated by ResNet and ResNet with extra losses, separately. Using $\mathcal{L}_{syn}$, we obtain the best results of underwater depth map estimation. Based on the experiments mentioned above, we introduce a perceptual loss $\mathcal{L}_{syn}$ to preserve the details and restrain the artifacts in Figure 6(e). To minimize the negative effects of the synthesized images, we design experiments to determine the proper weight of α and λ. In Table 5, we show the results of different weights, including α and λ. We note that both UW-Net and our model can be fine-tuned on the dataset of Berman *et al.* to obtain better results of underwater depth map estimation. Fine-tuning processing provides a flexible approach for adjusting our model and the estimation of depth maps from unexplored underwater regions within a relatively short period.

4.3.2 The Design of Underwater Depth Map Estimation

With the support of synthetic paired RGB-D data, we consider L1 loss, L2 loss, L_{ssim} loss, or L_{msssim} loss to learn the mapping functions for supervised depth map prediction. During the training, we observe the all above-mentioned losses are not enough to generate more correct depth maps. The results in Figure 7(b) - 7(e) show that depth prediction based on the above-mentioned losses are easily affected by the shape, noise, etc. As mentioned in section 3.2.2, we design depth loss L_{depth} to make better use of low-level and

Table 3. Comparison of Floating Point Operations (FLOPs) and total number of parameters among different generators with a size of 256×256

ine Methods	FLOPs	Params
ine StarGAN Choi et al. (2018)	52.32	8.417
CycleGAN Zhu et al. (2017a)	56.83	11.38
StarGANv2 Choi et al. (2020)	198.0	33.89
WaterGAN Li et al. (2017)	132.7	24.18
Ours (G_s)	52.93	8.426
Ours (G_d)	12.98	1.348
ine		



(a) In-air images (b) Baseline (c) w/ $\mathcal{L}_{ssim}$ (d) w/ $\mathcal{L}_{msssim}$ (e) w/ $\mathcal{L}_{syn}$

Figure 6. Sample results of our method for synthesizing underwater images using different losses. $\mathcal{L}_{ssim}$, $\mathcal{L}_{msssim}$ and $\mathcal{L}_{syn}$ respectively represent SSIM loss, MS-SSIM loss and perceptual loss. (a) are in-air images, (b) are the results without any structural loss (Baseline), (c)–(e) are the results with $\mathcal{L}_{ssim}$, $\mathcal{L}_{msssim}$ and $\mathcal{L}_{syn}$, respectively.

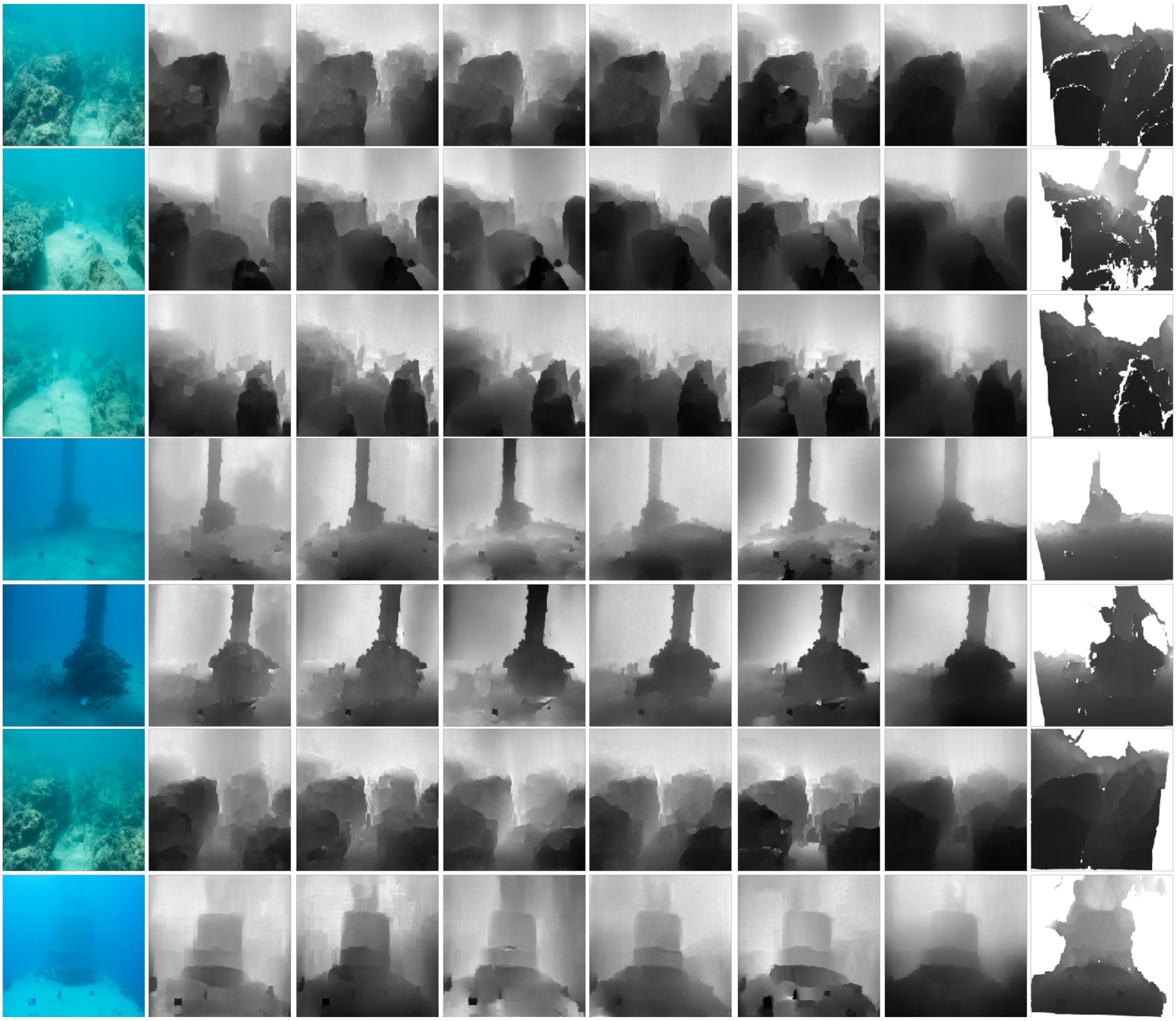
Table 4. Comparison of our method for the synthesis of underwater images with different combinations. ResNet (He et al., 2016) represents a basic network for the synthesis of underwater images (Baseline). Our synthesized underwater images are mainly used to estimate depth maps. We show the results of depth maps estimation using ResNet (He et al., 2016) and ResNet (He et al., 2016) with extra losses.

ine	Baseline	w/ $\mathcal{L}_{ssim}$	w/ $\mathcal{L}_{msssim}$	w/ $\mathcal{L}_{D_d}$	w/ $\mathcal{L}_{syn}$
ine SI-MSE	0.3538	0.2308	0.3331	0.2864	**0.1771**
ine ρ	0.6986	0.7547	0.7111	0.7355	**0.7796**
ine					

Table 5. Comparison of weights used in the objective function of our model, including α and λ. We separately set α = 1, 3, 5, 7 and λ = 0.05, 0.1, 0.2, 0.4. We discover that α = 5 and λ = 0.1 perform better.

ine SI-MSE/ρ	$\alpha = 1$	$\alpha = 3$	$\alpha = 5$	$\alpha = 7$
ine $\lambda = 0.05$	0.2586/0.7438	0.2676/0.7502	0.2325/0.7593	0.2957/0.7402
ine $\lambda = 0.1$	0.2291/0.7513	0.2020/**0.7844**	**0.1771**/0.7796	0.2321/0.7717
ine $\lambda = 0.2$	0.2955/0.7331	0.2164/0.7688	0.2548/0.7524	0.2535/0.7331
ine $\lambda = 0.4$	0.2966/0.7236	0.2882/0.7306	0.2929/0.7499	0.2577/0.7577
ine				

high-level feature information and avoid the risk of texture leakage. We take advantage of a pre-trained VGG19 network to extract feature maps between the generated depth maps and the ground truths. We assume the feature maps between the generated depth map and its corresponding ground truth in each layer from a pre-trained VGG19 network should be equal. The loss L_{depth} makes our model pay more attention to the objects and the relative distance in the underwater images. Inspired by Wang *et al.*'s work (Wang et al., 2018a), we also attempt to extract feature maps from the discriminator D_d, namely



(a) Input (b) $\mathcal{L}_1$ (c) $\mathcal{L}_2$ (d) $\mathcal{L}_{ssim}$ (e) $\mathcal{L}_{msssim}$ (f) $\mathcal{L}_{pan}$ (g) $\mathcal{L}_{depth}$ (h) GT

Figure 7. Effectiveness evaluation of the $\mathcal{L}_1$, $\mathcal{L}_2$, $\mathcal{L}_{ssim}$, $\mathcal{L}_{msssim}$ and $\mathcal{L}_{depth}$. From left to right, respectively, (a) are real underwater images, (b)–(h) are the results of depth map estimation with L1 loss, L2 loss, $\mathcal{L}_{ssim}$, $\mathcal{L}_{msssim}$, $\mathcal{L}_{pan}$, $\mathcal{L}_{depth}$ and their corresponding ground truths.

$\mathcal{L}_{pan}$, rather than a pre-trained VGG19 network. In Figure 7(f), we can see that our model with $\mathcal{L}_{pan}$ are often overwhelmed with incorrect boundary prediction due to the insufficient layers of our discriminator D_d to extract high-level feature maps comparing with $\mathcal{L}_{depth}$. Furthermore, we investigate the optimal parameter setting of η with a greedily searching strategy (Table 7), and we discover that η = 50 is the best choice among all the parameters.

Based on Figure 7 and Table 6, we can easily conclude that the results of depth map estimation using L_{depth} loss are more accurate and continuous. The results show sharper outlines. We can clearly distinguish the relative distance and the objects.

Table 6. Quantitative comparison of our method with different losses on the dataset of Berman *et al.* (Berman et al., 2017). Higher ρ values and lower SI-MSE (Eigen et al., 2014) values indicate better results.

ine	$\mathcal{L}_1$	$\mathcal{L}_2$	$\mathcal{L}_{ssim}$	$\mathcal{L}_{msssim}$	$\mathcal{L}_{pan}$	L_{depth}
ine SI-MSE	0.3103	0.2896	0.3983	0.2598	0.2856	**0.1771**
ine ρ	0.7279	0.7419	0.6515	0.7655	0.7397	**0.7796**
ine						

Table 7. Results with different η values. Higher ρ and lower SI-MSE (Eigen et al., 2014) values are better.

ine	$\eta = 40$	$\eta = 50$	$\eta = 60$	$\eta = 70$
ine SI-MSE	0.2657	**0.1771**	0.2620	0.2405
ine ρ	0.7266	**0.7796**	0.7315	0.7635
ine				

5 DISCUSSIONS AND CONCLUSION

To further explore the potential of our model on depth prediction, we considered the work by Li *et al.* (Li et al., 2018) and prepared a more complex underwater image dataset including 4 different styles. In this experiment, we still consider the depth map as a conditional input to synthesize a corresponding underwater image. But we did not utilize the physical parameters (e.g., the water turbidity or any optical parameters) for the unpaired image-to-image translation. Instead, we roughly divide the images with different water turbidity into 4 groups and follow the manner of StarGAN Choi et al. (2018) to perform conditional image translation. Some synthetic examples of 4 different styles are shown in Figure 8. Due to the lack of ground truth of the depth map, we cannot quantitatively evaluate the effectiveness of our model for multi-style underwater depth map estimation. Instead, we prepared several qualitative evaluation results, as shown in Figure 9. Intuitively, we find that the depth estimation of a side-view underwater image is better than that from a vertical view. This result is caused by the lack of vertical view in-air images from the in-air D-Hazy dataset required to produce sufficient synthetic underwater vertical view images. We plan to improve the performance on this point by data augmentation in the future.

In this paper, we proposed an end-to-end system that can synthesize multi-style underwater images using one-hot encoding and estimate underwater depth maps. The system can convert the in-air RGB-D images into more realistic underwater images with multiple watercolor styles. Then we use the synthesized underwater RGB images to construct a semi-real underwater RGB-D dataset. With the synthetic underwater RGB-D dataset, our model can learn to estimate underwater depth maps using supervised learning. Finally, we compare our method with existing state-of-the-art methods to synthesize underwater images and estimate underwater depth maps, and we verify that our method outperforms these methods both qualitatively and quantitatively. Furthermore, our model can be fine-tuned on the untrained datasets to synthesize a similar underwater style. It effectively makes our model to be applied for depth map estimation on new underwater datasets.

(a) In-air images (b) Blue (c) Green (d) White (e) Yellow

Figure 8. Sample results for the synthesis of underwater images. (a) show in-air images. (b)–(e) represent blue style, green style, white style and yellow style, respectively.

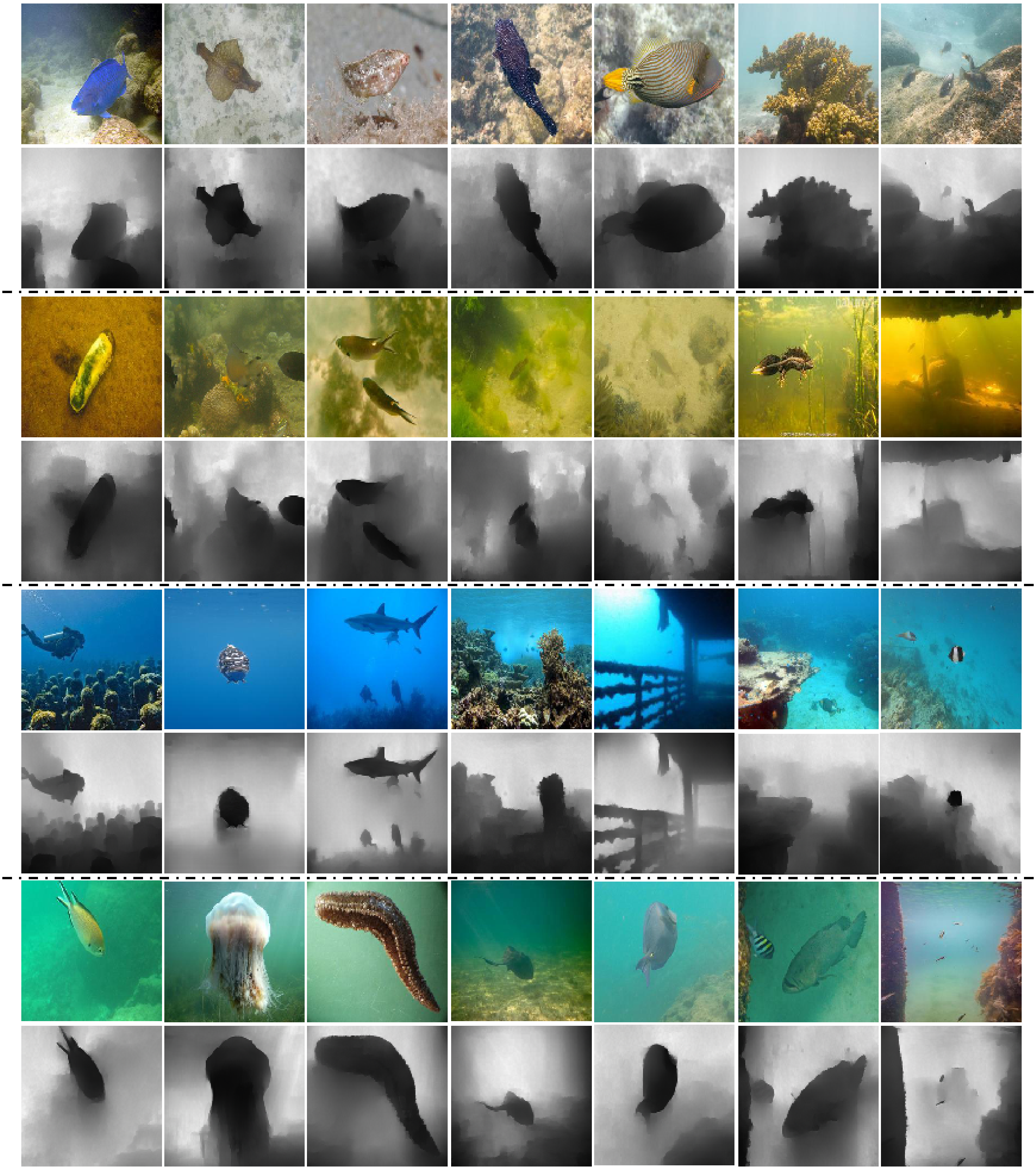


Figure 9. multi-style underwater depth map estimation. The rows from top to bottom are real underwater images with four different water types and the results of our model for depth map estimation. Every two rows are real underwater images and their predicted depth maps of our method.

ACKNOWLEDGEMENT

407 This work was supported by the finance science and technology project of 630 Hainan province of China
408 under Grant Number ZDKJ202017, National Natural Science Foundation of China under Grant Number
409 61701463.

REFERENCES

Abas, P. E., De Silva, L. C., et al. (2019). Review of underwater image restoration algorithms. *IET Image Processing* 13, 1587–1596

Akkaynak, D. and Treibitz, T. (2019). Sea-thru: A method for removing water from underwater images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 1682–1691

Ancuti, C., Ancuti, C. O., and De Vleeschouwer, C. (2016). D-hazy: A dataset to evaluate quantitatively dehazing algorithms. In *IEEE International Conference on Image Processing* (IEEE), 2226–2230

Berman, D., Levy, D., Avidan, S., and Treibitz, T. (2020). Underwater single image color restoration using haze-lines and a new quantitative dataset. *IEEE transactions on pattern analysis and machine intelligence*

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

Brock, A., Donahue, J., and Simonyan, K. (2018). Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*

Chen, R., Huang, W., Huang, B., Sun, F., and Fang, B. (2020). Reusing discriminators for encoding: Towards unsupervised image-to-image translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 8168–8177

Choi, Y., Choi, M., Kim, M., Ha, J.-W., Kim, S., and Choo, J. (2018). Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *IEEE Conference on Computer Vision and Pattern Recognition*. 8789–8797

Choi, Y., Uh, Y., Yoo, J., and Ha, J.-W. (2020). Stargan v2: Diverse image synthesis for multiple domains. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 8188–8197

Churnside, J. H., Marchbanks, R. D., Lembke, C., and Beckler, J. (2017). Optical backscattering measured by airborne lidar and underwater glider. *Remote Sensing* 9, 379

Dancu, A., Fourgeaud, M., Franjcic, Z., and Avetisyan, R. (2014). Underwater reconstruction using depth sensors. In *Special Interest Group Graph. Interact. Techn* (Association for Computing Machinery). 1–4

Deris, A., Trigonis, I., Aravanis, A., and Stathopoulou, E. (2017). Depth cameras on UAVs: A first approach. *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences* 42, 231

Drews, P. L., Nascimento, E. R., Botelho, S. S., and Campos, M. F. M. (2016). Underwater depth estimation and image restoration based on single images. *IEEE Computer Graphics and Applications* 36, 24–35

Eigen, D., Puhrsch, C., and Fergus, R. (2014). Depth map prediction from a single image using a multi-scale deep network. In *Advances in Neural Information Processing Systems*. 2366–2374

Gomez Chavez, A., Ranieri, A., Chiarella, D., Zereik, E., Babić, A., and Birk, A. (2019). Caddy underwater stereo-vision dataset for human–robot interaction (hri) in the context of diver activities. *Journal of Marine Science and Engineering* 7, 16

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., et al. (2014). Generative adversarial nets. In *Advances in Neural Information Processing Systems*. 2672–2680

Gupta, H. and Mitra, K. (2019). Unsupervised single image underwater depth estimation. In *IEEE International Conference on Image Processing* (IEEE), 624–628

He, K., Sun, J., and Tang, X. (2010). Single image haze removal using dark channel prior. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33, 2341–2353

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 770–778

Huang, X., Liu, M.-Y., Belongie, S., and Kautz, J. (2018). Multimodal unsupervised image-to-image translation. In *Proceedings of the European conference on computer vision (ECCV)*. 172–189

Islam, M. J., Xia, Y., and Sattar, J. (2020). Fast underwater image enhancement for improved visual perception. *IEEE Robotics and Automation Letters (RA-L)* 5, 3227–3234

Isola, P., Zhu, J.-Y., Zhou, T., and Efros, A. A. (2017). Image-to-image translation with conditional adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition*. 1125–1134

Jégou, S., Drozdzal, M., Vazquez, D., Romero, A., and Bengio, Y. (2017). The one hundred layers tiramisu: Fully convolutional densenets for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 11–19

Johnson, J., Alahi, A., and Fei-Fei, L. (2016). Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision*. 694–711

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*

Kupyn, O., Martyniuk, T., Wu, J., and Wang, Z. (2019). Deblurgan-v2: Deblurring (orders-of-magnitude) faster and better. In *Proceedings of the IEEE International Conference on Computer Vision*. 8878–8887

Lee, H.-Y., Tseng, H.-Y., Huang, J.-B., Singh, M., and Yang, M.-H. (2018). Diverse image-to-image translation via disentangled representations. In *Proceedings of the European conference on computer vision (ECCV)*. 35–51

Li, C., Guo, C., Ren, W., Cong, R., Hou, J., Kwong, S., et al. (2019). An underwater image enhancement benchmark dataset and beyond. *IEEE Transactions on Image Processing* 29, 4376–4389

Li, J., Skinner, K. A., Eustice, R., and Johnson-Roberson, M. (2017). Watergan: Unsupervised generative network to enable real-time color correction of monocular underwater images. *IEEE Robotics and Automation Letters*

Li, N., Zheng, Z., Zhang, S., Yu, Z., Zheng, H., and Zheng, B. (2018). The synthesis of unpaired underwater images using a multistyle generative adversarial network. *IEEE Access* 6, 54241–54257

Liu, M.-Y., Huang, X., Mallya, A., Karras, T., Aila, T., Lehtinen, J., et al. (2019). Few-shot unsupervised image-to-image translation. In *Proceedings of the IEEE International Conference on Computer Vision*. 10551–10560

Mao, X., Li, Q., Xie, H., Lau, R. Y., Wang, Z., and Paul Smolley, S. (2017). Least squares generative adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition*. 2794–2802

Massot-Campos, M. and Oliver-Codina, G. (2015). Optical sensors and methods for underwater 3d reconstruction. *Sensors* 15, 31525–31557

Mirza, M. and Osindero, S. (2014). Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*

Miyato, T., Kataoka, T., Koyama, M., and Yoshida, Y. (2018). Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957*

Odena, A., Olah, C., and Shlens, J. (2017). Conditional image synthesis with auxiliary classifier gans. In *International Conference on Machine Learning*. 2642–2651

Park, T., Liu, M.-Y., Wang, T.-C., and Zhu, J.-Y. (2019). Semantic image synthesis with spatially-adaptive normalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2337–2346

Peng, Y.-T., Zhao, X., and Cosman, P. C. (2015). Single underwater image enhancement using depth estimation based on blurriness. In *IEEE International Conference on Image Processing* (IEEE), 4952–4956

Pérez, J., Bryson, M., Williams, S. B., and Sanz, P. J. (2020). Recovering depth from still images for underwater dehazing using deep learning. *Sensors* 20, 4580

Ronneberger, O., Fischer, P., and Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer Assisted Intervention* (Springer), 234–241

Silberman, N., Hoiem, D., Kohli, P., and Fergus, R. (2012). Indoor segmentation and support inference from rgbd images. In *European conference on computer vision* (Springer), 746–760

Tan, M. and Le, Q. V. (2019). Efficientnet: Rethinking model scaling for convolutional neural networks

Ueda, T., Yamada, K., and Tanaka, Y. (2019). Underwater image synthesis from rgb-d images and its application to deep underwater image restoration. In *2019 IEEE International Conference on Image Processing (ICIP)* (IEEE), 2115–2119

Wang, C., Xu, C., Wang, C., and Tao, D. (2018a). Perceptual adversarial networks for image-to-image transformation. *IEEE Transactions on Image Processing* 27, 4066–4079

Wang, T.-C., Liu, M.-Y., Zhu, J.-Y., Liu, G., Tao, A., Kautz, J., et al. (2018b). Video-to-video synthesis. *arXiv preprint arXiv:1808.06601*

Wang, T.-C., Liu, M.-Y., Zhu, J.-Y., Tao, A., Kautz, J., and Catanzaro, B. (2018c). High-resolution image synthesis and semantic manipulation with conditional gans. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 8798–8807

Xiao, J., Hays, J., Ehinger, K. A., Oliva, A., and Torralba, A. (2010). Sun database: Large-scale scene recognition from abbey to zoo. In *IEEE Conference on Computer Vision and Pattern Recognition*. 3485–3492

Ye, X., Li, Z., Sun, B., Wang, Z., Xu, R., Li, H., et al. (2019). Deep joint depth estimation and color correction from monocular underwater images based on unsupervised adaptation networks. *IEEE Transactions on Circuits and Systems for Video Technology* 30, 3995–4008

Zhang, H., Goodfellow, I., Metaxas, D., and Odena, A. (2018). Self-attention generative adversarial networks. *arXiv preprint arXiv:1805.08318*

Zhang, Y. and Yang, Q. (2017). A survey on multi-task learning. *arXiv preprint arXiv:1707.08114*

Zheng, Z., Wu, Y., Han, X., and Shi, J. (2020). Forkgan: Seeing into the rainy night. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16* (Springer), 155–170

Zhu, J.-Y., Park, T., Isola, P., and Efros, A. A. (2017a). Unpaired image-to-image translation using cycle-consistent adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition*. 2223–2232

Zhu, J.-Y., Zhang, R., Pathak, D., Darrell, T., Efros, A. A., Wang, O., et al. (2017b). Toward multimodal image-to-image translation. In *Advances in neural information processing systems*. 465–476

APPENDIX

Generator architectures. In our experiments, the generator G_s from CycleGAN (Zhu et al., 2017a) and StarGAN (Choi et al., 2018) can be described as Figure 10. Here, Convolution denotes a 7×7 Convolution-InstanceNorm-ReLU layer with 64 filters and stride 1. Convolution/down denotes a 4×4 Convolution-InstanceNorm-ReLU layer and stride 2. Residual block denotes a residual block that contains two 3×3 Convolution-InstanceNorm-ReLU layers with the same number of filters on both layers. Deconvolution denotes a 4×4 fractional-strided-Convolution-InstanceNorm-ReLU layer and stride 2.

The generator G_d from Jégou *et al.* (Jégou et al., 2017) is based on dense-block (DB), as Figure 11. Convolution denotes a 3×3 Convolution-BatchNorm-ReLU layer with 32 filters and stride 1. Transition

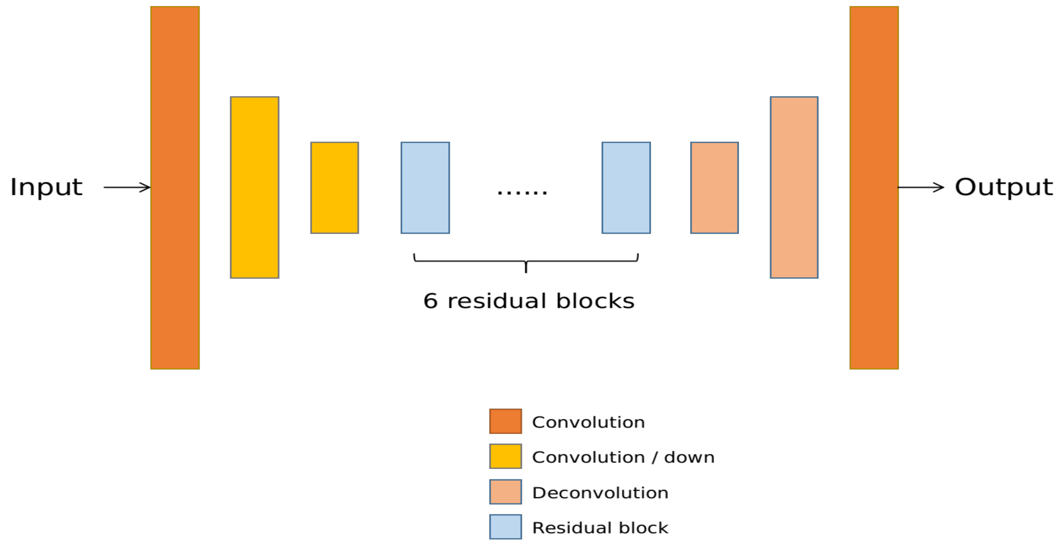


Figure 10. The network architecture of the generator G_s. It is a general ResNet (He et al., 2016) network for image-to-image translation .

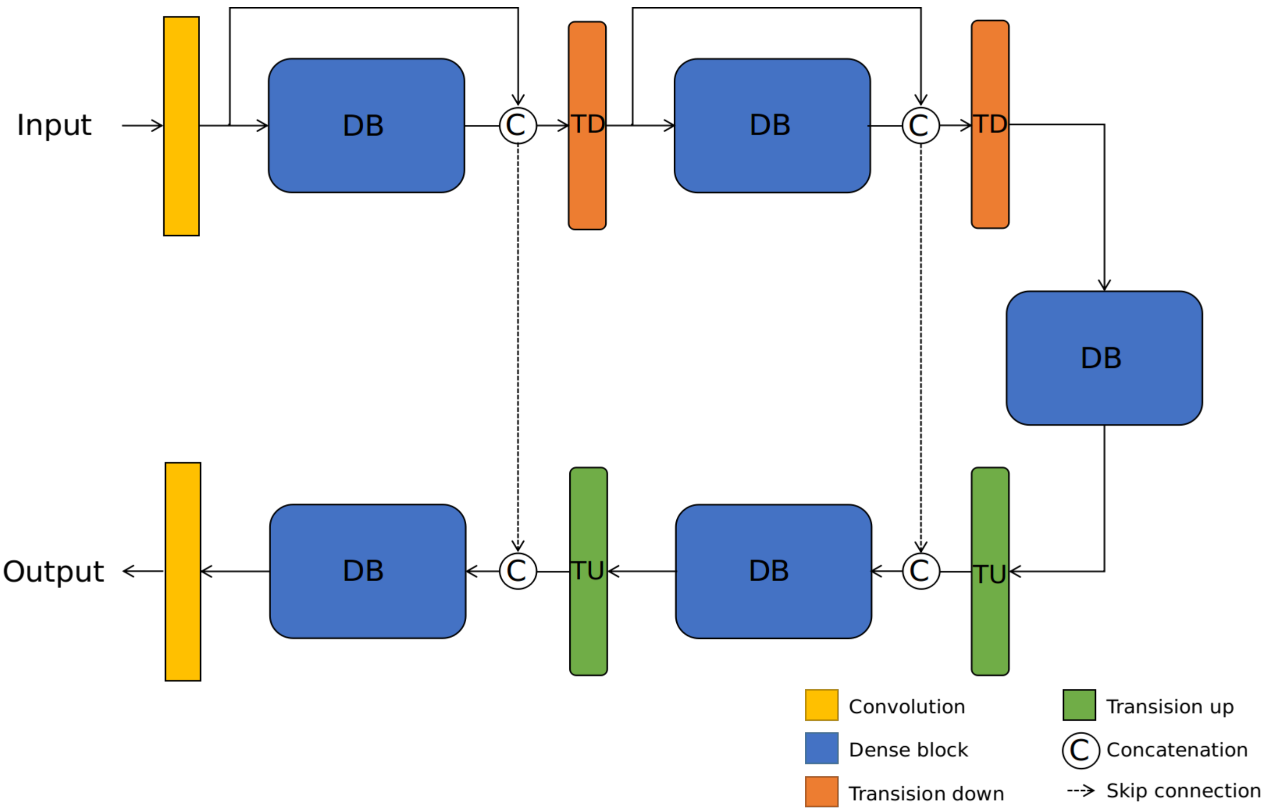


Figure 11. The network architecture of the generator G_d. Following the work of UW-Net (Gupta and Mitra, 2019), we choose DenseNet (Jégou et al., 2017) as the generator G_d.

down is a maxpool2d operation with the same number of filters and a 1×1 Convolution-BatchNorm-ReLU layer with the same number of filters and stride 1. Transition up denotes a 4×4 deconvolution layer with the same number of filters and stride 2. Dense block denotes four 3×3 BatchNorm-ReLU-Convolution layers with 12 filters and stride 1. The output channel number of the dense block is the concatenation from the output of four layers and the input. The encoder and the decoder concatenate with skip connection.

Discriminator architectures. For discriminator networks, we use 70×70 PatchGANs (Isola et al., 2017; Zhu et al., 2017a). Similarly, we do not use InstanceNorm or BatchNorm in any layer and use leaky ReLUs with a slope of 0.2. The discriminator D_s has two outputs from the discrimination branch and the classification branch. Differently, the discriminator D_d only has one discrimination output.