Deep Joint Depth Estimation and Color Correction from Monocular Underwater Images based on Unsupervised Adaptation Networks

Xinchen Ye, Zheng Li, Baoli Sun, Zhihui Wang, Rui Xu, Haojie Li, and Xin Fan

***Abstract*—Degraded visibility and geometrical distortion typically make the underwater vision more intractable than open air vision, which impedes the development of underwater-related machine vision and robotic perception. Therefore, this paper addresses the problem of joint underwater depth estimation and color correction from monocular underwater images, which aims at enjoying the mutual benefits between these two related tasks from a multi-task perspective. Our core ideas lie in our new deep learning architecture. Due to the lack of effective underwater training data, and the weak generalization to the real-world underwater images trained on synthetic data, we consider the problem from a novel perspective of style-level and feature-level adaptation, and propose an unsupervised adaptation network to deal with the joint learning problem. Specifically, a style adaptation network (SAN) is first proposed to learn a style-level transformation to adapt in-air images to the style of underwater domain. Then, we formulate a task network (TN) to jointly estimate the scene depth and correct the color from a single underwater image by learning domain-invariant representations. The whole framework can be trained end-to-end in an adversarial learning manner. Extensive experiments are conducted under air-to-water domain adaptation settings. We show that the proposed method performs favorably against state-of-the-art methods in both depth estimation and color correction tasks.**

***Index Terms*—Underwater Image, Depth estimation, color correction, unsupervised, domain adapatation.**

I. Introduction

NOWADAYS, underwater activities, e.g., underwater culture and monitoring [1], robotic operations [2], and underwater visual navigation [3], become prominent for the purpose of ocean science development. However, machine vision has been challenged by the underwater environment. Degraded visibility and geometrical distortion typically make the underwater vision more intractable than open air vision [4].

First, the discoloration effect in captured underwater images needs to be removed to provide a clear vision of the underwater scene. Many methods, including image-based [5], model-based [6], [7], and CNN-based [8]–[10] algorithms, have been proposed to correct the color-distorted underwater images. However, these methods under-perform as they use various assumptions and handcrafted constrains to model the complex underwater environment, or apply CNN trained on other unrelated datasets which does not always generalize to the real-world underwater images.

Besides, acquiring high quality 3D structure in the underwater environment [3], [11] is also essential. The depth map serves as a basic element for underwater 3D reconstruction and robot navigation. However, depth maps directly obtained by depth sensors [1], [12] or computed from stereo matching [13] are unsatisfactory due to the imaging limitation and optical distortion. Therefore, it is necessary to develop a method to estimate depth maps from monocular underwater images. Unfortunately, compared to the task of monocular depth estimation in the air, *almost no* research focuses on depth estimation in the underwater environment due to the lack of effective training data.

A. Motivation

As we observe, the tasks of depth estimation and color correction from a single underwater image have a strong connection essentially. First, the depth map is closely related to the formation of ambient light attenuation and backscattering in the water [14], [15]. For example, one common underwater imaging model derives from the Jaffe-McGlamery model [16] as:

$$I_c(x) = J_c(x)e^{-\beta_c d(x)} + A_c(1 - e^{-\beta_c d(x)}) \quad (1)$$

where I_c and J_c are the observed degraded image and clear image to be recovered respectively, c represents a color channel. A_c is global atmospheric light. $d(x)$ represents the scene depth, and β_c is the scattering coefficient. The wavelength-dependent attenuation causes color distortions that increase with an object's distance, while the scattering induces a distance-dependent additive component on the scene which reduces contrast. These phenomena cannot be globally corrected since the color degradation depends on the 3D scene structure. Meanwhile, a color image and the associated depth map are the photometric and geometrical representation of the same scene, and have strong structural similarity. Pixels with similar appearances have more chances of belonging to the same object, and should have close depth values [17], [18]. In summary, these clues indicate the strong correlations between both tasks, which motivates us to learn them together aiming at enjoying mutual benefits from each other.

Copyright © 2019 IEEE. Personal use of this material is permitted. However, permission to use this material for any other purposes must be obtained from the IEEE by sending a request to pubs-permissions@ieee.org.

This work was supported by National Natural Science Foundation of China (NSFC) under Grant 61702078, 61772108.

X. Ye, Z. Li, B. Sun, Z. Wang, R. Xu, H. Li, and X. Fan are with DUT-RU International School of Information Science & Engineering, Dalian University of Technology, Liaoning, and Key Laboratory for Ubiquitous Network and Service Software of Liaoning Province, China (Corresponding author: Z. Wang. E-mail: wangzhihui1017@gmail.com).

Besides, unlike in-air computer vision tasks, the training dataset with enough number of real underwater paired images is not easy to acquire. For example, building image-to-depth dataset for a given underwater scene type needs high-effort and high-cost for specialized depth-sensing equipment, while we even need to 'evacuate water from the sea' to capture the same scene without water to generate the color pairs. While available in-air datasets have been used to circumvent above problems, the learned distribution by CNN does not always generalize to the natural underwater scenes due to the inherent domain shift. Therefore, unsupervised domain adaptation would be desirable to address this challenge, which resorts to use available paired in-air images from the source domain and unpaired real underwater images in the target domain to reduce the prediction error on the target data. Therefore, We separately consider the problem from the perspective of style-level and feature-level adaptation. Since the underwater appearance encodes all the distance-dependent discoloration effect, the objective of style-level adaptation is to make the visual styles domain-invariant, i.e., learns a style-level transformation to adapt in-air images to the style of underwater domain based on the known in-air distance information. The motivation of feature-level adaptation is to guide the representation learning in both feature domains to mitigate the difference between source and target representation distributions through the domain discriminator. We focus on adversarial mechanism [19] to learn both in-air and underwater image representations that are invariant across domains.

B. Scope and Contributions

Following the above analysis, we present a deep learning framework for joint depth estimation and color correction from monocular underwater images based on unsupervised adaptation networks, as shown in Fig. 1. The framework consists of a style adaptation network (SAN) and a task network (TN). Specifically, SAN aims at rendering synthetic underwater images from in-air RGB-D images discriminated against real underwater images, i.e., combining texture content from in-air images and style information of the underwater domain in the pixel space. In TN, we propose a stacked network architecture to construct an unified depth estimation sub-network (DESN) and color correction sub-network (CCSN). Meanwhile, domain discriminator is employed to distinguish between source and target domain at the feature space for both sub-networks. Our contributions in this paper are as follows:

1) A joint learning framework for underwater depth estimation and color correction that exploits strong correlations on possible mutually improved benefits between each other.

2) A novel, end-to-end trainable unsupervised architecture that learns without paired underwater training images.

3) With both style-level and feature-level adaptations, our framework could better build invariance across domains and thus obtain high quality depth map and corrected color image.

Note that, our main purpose is to solve the more challenging underwater perception problem. We are the first to estimate depth map in the underwater domain without real paired data, and develop a multi-task learning framework to fully exploit mutual correlation between depth estimation and color correction. No work has yet considered using adaptation techniques to solve the lack of real underwater data, and jointly learning the two related tasks. Experiments also validate that the network not only performs well on synthetic data, but also achieves superior performance on real underwater data. We believe that our work would make contributions to underwater-related machine vision and robotic perception.

II. Related Work

In this section, we give a overview of related work in three aspects, i.e. underwater color correction, monocular depth estimation in open-air, and unsupervised domain adaptation.

A. Underwater Color Correction

Traditional underwater image enhancement methods can be divided into image-based and model-based algorithms. Image-based algorithms [5], [20]–[22] estimate the transmission map directly from captured underwater image and then use it for haze removal and color correction. Model-based algorithms [6], [7], [23], [24] are to model the underwater imaging process under the underwater optical properties to help to recover the image. Due to the powerful learning ability, CNN presents promising solutions to in-air dehazing [25], which also can be transferred to the application of underwater image enhancement [8]–[10]. The recent work all focus on employing CNN or GAN [19], [26], [27] to accomplish the target. Shin *et al.* [8] proposed a common convolutional architecture to estimate ambient light and transmission map. Wang *et al.* [9] proposed an end-to-end framework for jointly infer transmission map and correct color image. Li *et al.* [28] introduced an extra structural similarity index measure loss under the backbone of cycleGAN [27] to correct the color in an unsupervised manner. WaterGAN [10] generated underwater images from in-air image and depth pairs in an unsupervised pipeline, and trained a SegNet [29] on the synthetic dataset to output the corrected image.

B. Monocular Depth Estimation

Traditional methods on monocular depth estimation mainly focus on graphical models with hand-crafted geometry priors [30] or non-parametric depth transfer techniques [31], [32] based on the assumption that similar colors in a region from RGB images imply consistent depth values. Existing methods use CNN to model an implicit relation between image-depth pairs [33]–[37]. Eigen *et al.* [33] proposed a three-scale pyramid CNN architecture as a coarse-to-fine manner for regressing dense depth maps based on VGG. Laina *et al.* [36] used a deeper ResNet-50 architecture [38] to obtain a larger receptive field, and designed up-projection modules to progressively upsample the depth map. Lee *et al.* [37] used the fourier frequency domain analysis to enhance the dataset, and redesigned a depth balanced Euclidean loss function to estimate the depth better in closer areas.

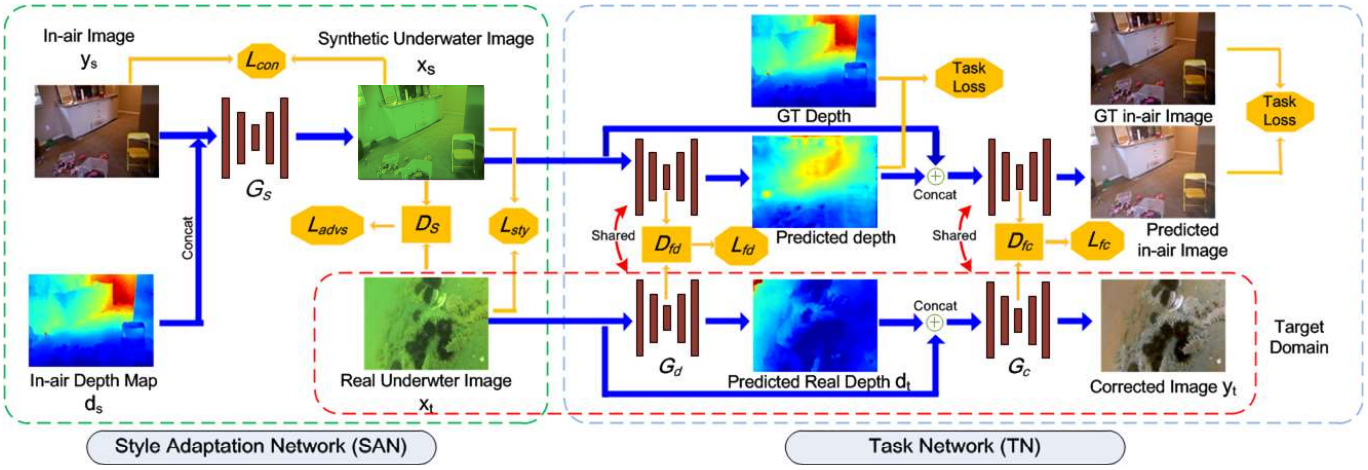


Fig. 1. Our whole network architecture for joint depth estimation and color correction. It consists of two networks, i.e., style adaptation network (SAN) and task network (TN). Domain adaptation is used in both networks to adapt for low-level appearance style and high-level feature representation, simultaneously. The whole framework can be trained end-to-end in an adversarial learning manner. The depth maps are colored with red for farther distance while blue for closer distance for easy observation.

C. Unsupervised Domain Adaptation

Domain adaptation methods have been developed to match the feature distributions between the source domain and the target domain. Csurka *et al.* [39] gives an overview of domain adaptation and transfer learning with a specific view on visual applications. Kundu *et al.* [40] proposed an unsupervised domain adaptation strategy, which uses adversarial learning and explicit imposition of content consistency on the adapted target representation for the task of monocular depth estimation. CyCADA *et al.* [41] transfered source domain images to the target domain with pixel alignment based on CycleGAN [27], thus generating extra training data combined with feature space adversarial learning. Tsai *et al.* [42] constructed a multi-level adversarial network to effectively perform output space domain adaptation at different feature levels.

In short, GANs [19] have achieved impressive results in various image-to-image translation problems, such as super-resolution [43], style transfer [27] and domain adaptation [39]. Our work mainly focuses on unsupervised adaptation for joint underwater depth estimation and color correction, which is seldom investigated. The most closely related work is WaterGAN [10], which only focuses on the tasks of color correction trained on generated dataset. Different from WaterGAN, our method addresses the problem of joint learning tasks, and solves the domain discrepancy from the perspectives of both visual style-level and feature-level domain adaptation, which bridges the domain gap in a more principled way and achieves better performances in both tasks.

III. Proposed Method

Fig. 1 illustrates the overview of our network architecture. It consists of two main components: the style adaptation network (SAN) and the task network (TN). Given that we have a collection of in-air RGB-D images, which contains color images $y_s \in \mathcal{Y}_s$ and the corresponding depth maps $d_s \in \mathcal{D}_s$. Meanwhile, the real underwater images $x_t \in \mathcal{X}_t$ are captured from underwater environment, but we do not have access to the corresponding real depth maps $d_t \in \mathcal{D}_t$ and corrected color images $y_t \in \mathcal{Y}_t$. Our main goal is to train the networks G_d, G_c, such that when presented with a single underwater RGB image x_t, they predict the corresponding depth map d_t and the corrected color image y_t simultaneously. Note that, we cannot directly train G_d, G_c due to the absence of paired underwater images $\{x_t, y_t\}$ and $\{x_t, d_t\}$. Therefore, we resort to use in-air image y_s to render synthetic underwater image x_s, i.e., learning a transformation G_s in SAN to adapt y_s to the appearance style of underwater domain but without changing the scene content. Then, we train G_d, G_c in TN with the synthetic data, and use two discriminators D_{f_d}, D_{f_c} to apply feature-level domain adaptation on G_d, G_c to guide the feature representation learning in both domains, making the source and target representations indistinguishable through the domain discriminator. Based on both visual style-level and feature-level domain adaptation, we accomplish the tough task in a more principled way.

A. Style Adaptation Network (SAN)

Our SAN is to learn a style-level transformation G_s to adapt in-air images $\mathcal{Y}_s$ to the style of underwater domain based on the distance knowledge $\mathcal{D}_s$. Since the underwater appearance encodes all the distance-dependent discoloration effect including attenuation and backscattering which are important to the task of color correction and depth estimation, the appearance features over the whole image in underwater domain should be separated and regarded as the low-level 'style' of underwater domain. In contrast, the high-level content of the scene structures in the in-air image should be extracted and recombined with the 'style' of underwater domain. Thus, we can obtain paired training images with the same content but different appearance to facilitate the following joint learning task.

We follow the GAN approach for this minimax game setting and consider the following adversarial loss that characterizes the interplay between the generator G_s and a discriminator

D_s:

$$\begin{aligned}\mathcal{L}_{adv_s}(G_s, D_s) =& \mathbb{E}_{x_t\sim\mathcal{X}_t}[\log D_s(x_t)]+ \\ & \mathbb{E}_{y_s\sim\mathcal{Y}_s}[\log(1-D_s(G_s(y_s,d_s)))]\end{aligned} \tag{2}$$

To achieve content preservation and style transfer simultaneously, we explicitly incorporate two perceptual loss components along with the adversarial loss, namely the style loss and the content loss. We follow the style transfer idea of [44], [45] to utilize the pretrained VGG-19 model to extract features from its multiple layers. For style loss, based on VGG-19 feature representation, the Gram matrix is considered to characterize pairwise style correlations between feature responses from a statistical measurement perspective. The Gram matrix $\mathbb{G}^l$ is defined as the inner product between the m-th and n-th vectorized feature maps in the l-th layer:

$$\mathbb{G}^l_{m,n} = \sum_k \widetilde{\phi}^l_{mk}\widetilde{\phi}^l_{nk} \tag{3}$$

where $\widetilde{\phi}$ is the vectorized version of feature map ϕ in VGG-19 model, and k is the number of channels. With the style representation $\mathbb{G}^l$ of the synthetic image x_s and the real underwater image x_t, the style loss can be defined as

$$\mathcal{L}_{sty}(G_s) = \sum_{l\in L_s}\omega^l_s||\mathbb{G}^l(x_t) - \mathbb{G}^l(G_s(y_s,d_s))||^2_2 \tag{4}$$

where ω^l_s depicts the weight of the l-th layer Gram matrix, L_s is the set of layers to be considered in the style loss. Besides, the responses in different convolutional layers characterize different sematic image content, and thus the content loss is also engaged to enforce the generated image retaining the scene structure, which is defined as

$$\mathcal{L}_{con}(G_s) = \sum_{l\in L_c}\omega^l_c||\phi^l(y_s) - \phi^l(G_s(y_s,d_s))||^2_2 \tag{5}$$

These two losses ensure the generated images align with the RGB-D input, such that the network does not alter the underlying structure of the scene itself, only its appearance. Combined with the above three losses, then we have the final loss for our SAN:

$$\mathcal{L}_{SAN} = \mathcal{L}_{adv_s} + \lambda_a\mathcal{L}_{sty} + \lambda_b\mathcal{L}_{con} \tag{6}$$

where λ_a and λ_b are weighting parameters.

Note that, one can use the traditional physical way [14] to simulate underwater effects. However, underwater image formation is subject to complex process of light propagation through water column. Even some methods [15] are still developing better physical models, but existing models depend on many parameters intrinsic to the water column in the scene, which makes recovery of these parameters difficult without simplifying assumptions. This motivates us to learn such a complex nonlinear network to address the problem.

B. Task Network (TN)

The main goal of TN is to jointly train the generators G_d, G_c based on the dataset $\{\mathcal{X}_s, \mathcal{Y}_s, \mathcal{D}_s\}$, where $\mathcal{X}_s$ is rendered from SAN. As shown in Fig. 2, our TN consists of two conditional GAN networks, i.e., depth estimation sub-network (DESN) and color correction sub-network (CCSN), where the second one is stacked on the first one. Each sub-network contains a generator and a discriminator, and we denote G_d,D_d and G_c,D_c for them, respectively. In DESN, G_d is conditioned on the input synthetic underwater image x_s, while both x_s and the generated depth map d_s are sent to D_d to distinguish from the corresponding ground-truth pairs. CCSN takes the pair of $\{x_s, d_s\}$ as G_c's input, and outputs the corrected RGB image y_s. A triplets that containing $\{x_s, y_s, d_s\}$ is fed to D_c to distinguish from the corresponding ground-truth triplets. The adversarial loss for both sub-networks can be expressed as follows:

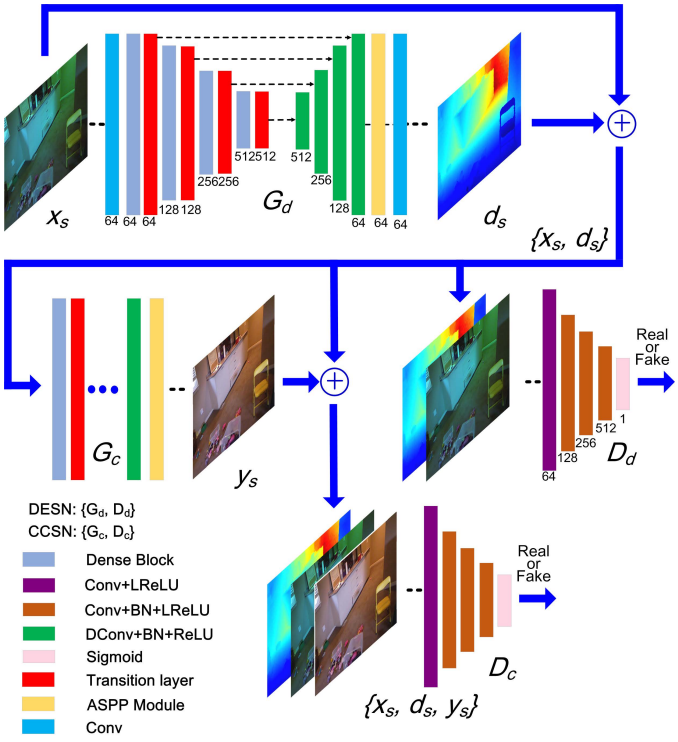


Fig. 2. Our stacked conditional GANs architecture for joint depth estimation and color correction. G_c is sketched out briefly and the domain adaptation modules on both generators are omitted for saving space.

$$\begin{aligned}\mathcal{L}_{adv_d}(G_d, D_d) =& \mathbb{E}_{x_s\sim\mathcal{X}_s, d_s\sim\mathcal{D}_s}[\log D_d(x_s,d_s)] \\ & + \mathbb{E}_{x_s\sim\mathcal{X}_s}[\log(1-D_d(x_s, G_d(x_s)))]\end{aligned} \tag{7}$$

$$\begin{aligned}&\mathcal{L}_{adv_c}(G_c, D_c|G_d) = \\ &\mathbb{E}_{x_s\sim\mathcal{X}_s, y_s\sim\mathcal{Y}_s, d_s\sim\mathcal{D}_s}[\log D_c(x_s,y_s,d_s)]+ \\ &\mathbb{E}_{x_s\sim\mathcal{X}_s}[\log(1-D_c(x_s, G_d(x_s), G_c(x_s, G_d(x_s))))]\end{aligned} \tag{8}$$

Besides, we directly measure per-pixel difference between the predicted images and its corresponding groudtruths as a task loss to encourage an accurate regression:

$$\begin{aligned}\mathcal{L}_t(G_d, G_c) =& ||d_s - G_d(x_s)||_1 + \\ & ||y_s - G_c(x_s, G_d(x_s))||_1\end{aligned} \tag{9}$$

Note that, the design of the proposed stacked sub-networks is beneficial to our multi-task learning. It can not only focus on one task once a time in different stages, but also shares mutual improvements through forward and backward information propagation.



Fig. 3. Training datasets: From left to right are real underwater images captured under three different light conditions ('bluish', 'greenish', and 'shallow'), and in-air NYU dataset, respectively.

Moreover, we want to bridge the domain gap of the internal feature representations between in-air and underwater images. Thus, we further define a representation-level loss to increase the domain discrimination capability in the encoder-decoder network G_d and G_c (shown in Fig. 1):

$$\begin{aligned}\mathcal{L}_{f_d}(G_d, D_d) = &\mathbb{E}_{f_{x_t}\sim f_{\mathcal{X}_t}}[\log D_{f_d}(f_{x_t})] \\ &+ \mathbb{E}_{f_{x_s}\sim f_{\mathcal{X}_s}}[\log(1 - D_{f_d}(f_{x_s}))]\end{aligned} \tag{10}$$

where f_{x_t} and f_{x_s} are features obtained from the last transition layer of encoder portion in G_d for real image x_t and synthetic image x_s respectively [1]. Then, the total loss of TN is the linear combination of the abovementioned losses with the weights $\lambda_t, \lambda_d, \lambda_c$ as follows:

$$\mathcal{L}_{TN} = \mathcal{L}_{adv_d} + \mathcal{L}_{adv_c} + \lambda_t\mathcal{L}_t + \lambda_d\mathcal{L}_{f_d} + \lambda_c\mathcal{L}_{f_c} \tag{11}$$

Taken together, our full objective for the proposed framework is $\mathcal{L} = \mathcal{L}_{SAN} + \mathcal{L}_{TN}$. In our end-to-end training, this objective function is used to solve for optimal G_d and G_c parameters.

Algorithm 1 Training Details

Input: Training datasets $\mathcal{Y}_s, \mathcal{D}_s, \mathcal{X}_t$; weighting parameters λ

–1) Training SAN separately—
Initialize G_s, D_s with random weights
for i = 1; i $\leq$ 30th epoch **do**
    Train G_s, D_s alternately by optimizing the GAN loss $\mathcal{L}_{adv_s}$ in Eq. (2);
end for
for i = 31; i $\leq$ max epoch **do**
    Train G_s, D_s alternately by optimizing the overall $\mathcal{L}_{SAN}$ in Eq. (6);
end for

Output: $x_s = G_s(y_s, d_s) \in \mathcal{X}_s$;

—2) Training TN separately—;
Initialize $G_d, G_c, D_d, D_c, D_{f_d}, D_{f_c}$ with random weights
for i = 1; i $\leq$ 100th epoch **do**
    Train D_d, D_c with G_d, G_c fixed by optimizing $\mathcal{L}_{advd} + \mathcal{L}_{advc}$ in Eq. (7) and Eq. (8);
    Train G_d, G_c with D_d, D_c fixed by jointly optimizing $\mathcal{L}_{advd} + \mathcal{L}_{advc} + \mathcal{L}_t$ in Eq. (7-9) using $\{\mathcal{Y}_s, \mathcal{D}_s, \mathcal{X}_s\}$;
end for
for i = 101; i $\leq$ max epoch **do**
    Fine-tune the task model aligned with D_{f_d}, D_{f_c} by optimizing the overall loss $\mathcal{L}_{TN}$ in Eq. (11);
end for
—3) Joint training SAN and TN—
for i = 1; i $\leq$ training epochs **do**
    Fine-tune the whole framework by optimizing $\mathcal{L} = \mathcal{L}_{SAN} + \mathcal{L}_{TN}$ via the pre-trained models in 1) and 2);
end for

Output: G_d, G_c;

IV. Network Architecture and Training

A. Network Architecture

In SAN, we have a generator G_s and a discriminator D_s for style-level domain adaption. In TN, we totally have two generators (G_d, G_c), two discriminators for tasks learning (D_d, D_c), and two discriminators (D_{f_d}, D_{f_c}) for feature-level domain adaptation. All the three generators inherit the same structure we design as follows, and so do the five discriminators.

The generator adopts the basic U-Net architecture [46]. The encoder is made up of four similar modules, each containing a dense block layer [47] and a transition layer. The former one has three 3-layer dense blocks, while the latter one consists of batch normalization (BN), ReLU, convolution (Conv), and average pooling sequentially. The decoder has corresponding four symmetric modules, each with a combination of deconvolution (DConv), BN, and ReLU. An extra atrous spatial pyramid pooling (ASPP) scheme [48] is employed at the end of the network to capture the multi-scale information. For the discriminator, one Conv + Leaky ReLU (LReLU) block is placed at first, followed by three Conv+BN+LReLU blocks and a sigmoid layer to output the real or fake label. The number of feature maps in each layer are shown in Fig. 2.

B. Training Details

As shown in Fig. 3, we choose three representative real underwater datasets, including two video datasets ('bluish', 'greenish') from underwater color cast set (UCCS) [49], and one image dataset ('shallow') in [50]. The first two datasets are established using approximately 500 frames extracted from videos, while the last one contains about 100 images. All the three datasets are captured under different light conditions. We use the indoor NYU RGB-D v2 dataset [51] as our in-air dataset. There are totally 1449 RGB and depth images in NYU dataset. Following the official splitting, We use 795 and 654 images pairs for training and testing separately. We augment the training data with rotation and flipping operations up to 14K images. To solve the distance scale problem between the indoor scene and underwater scene, we have normalized depth maps of in-air data to the maximum distance of underwater scene. For each kind of underwater dataset, we will train a separate network and predict scene depth maps and corrected color images from real underwater images to validate our method.

[1]Similar to the expression of $\mathcal{L}_{f_d}$, we also have a loss $\mathcal{L}_{f_c}$ applied on the same place in G_c, which does not present for saving space.

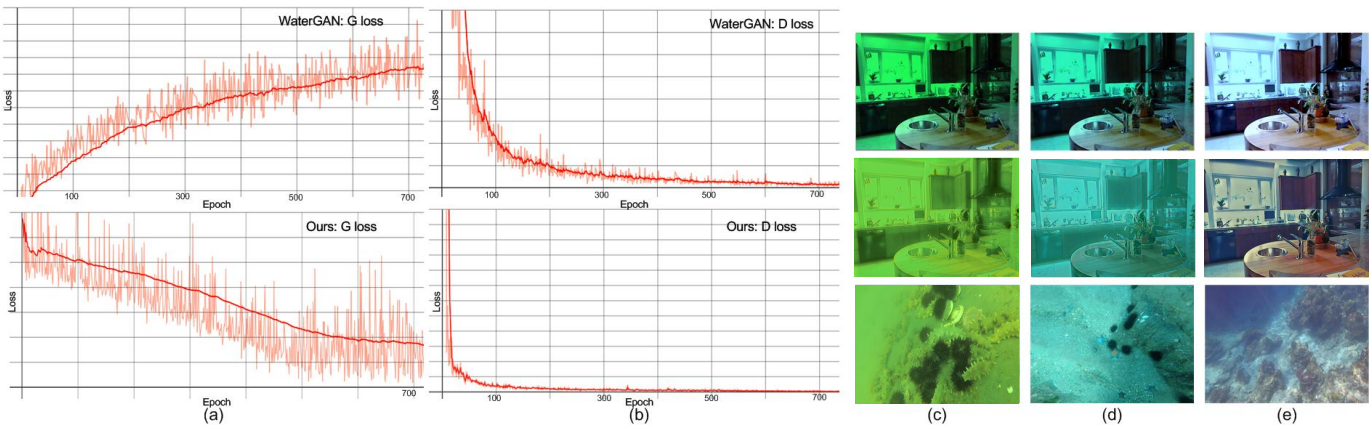


Fig. 4. Evaluation on SAN from the perspective of training details and rendering results. (a) and (b) show the loss curves of training process from WaterGAN [10] and ours. (c-e) present three visual examples for clearly observing the rendering results. From top to bottom are the results from WaterGAN, ours, and the real underwater images, respectively.

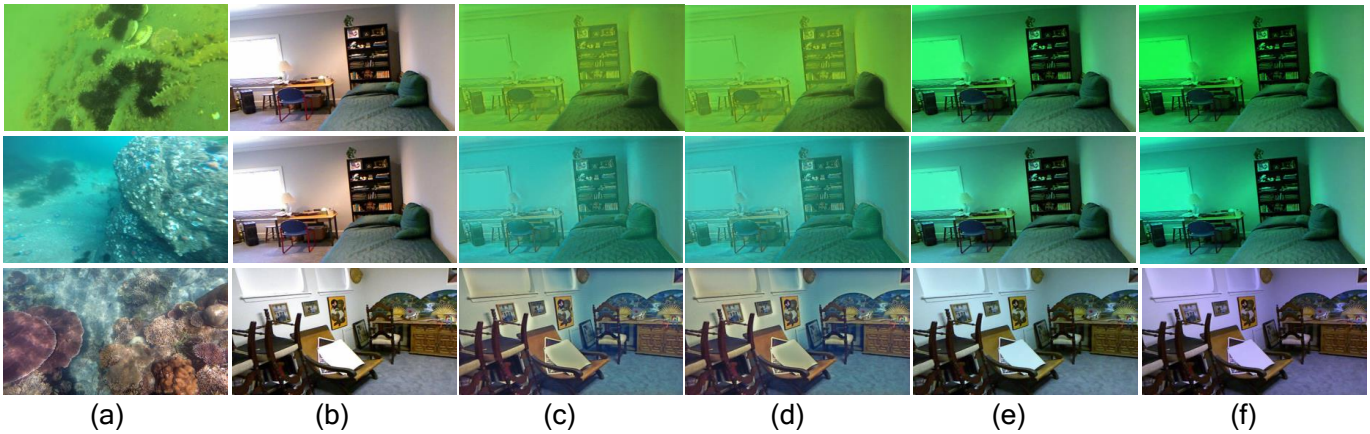


Fig. 5. Qualitative rendering results comparison between WaterGAN and ours. (a) Real underwater images; (b) Original in-air images; Rendering results generated from: (c) Ours at the 500th epoch; (d) Ours at the 700th epoch; (e) WaterGAN at the 500th epoch; (f) WaterGAN at the 700th epoch.

Initially, we train SAN and TN separately with respective loss functions. For SAN, we first train the network using the adversarial loss alone. Then, the style loss and content loss will be aligned with the adversarial loss after 30 epochs. In addition, we use {conv1-1, con2-1, con3-1, con4-1, conv5-1}, i.e., 1st, 3rd, 5th, 9th, 13th layers in VGG-19 to extract features, as the representations of these layers in general have the highest capability in each scale. It has been observed that features from lower layers such as conv1-1 and con2-1 preserve detailed pixel information, while the higher layer features tend to maintain high-level semantic patterns. The content loss relates to the scene structures. By including multiple layers, the model is able to capture the multi-scale structural information on the image and thus we consider a content loss function based on all the five layers. Meanwhile, the style loss can be regarded as a kind of statistical measurement or pattern, which is agnostic to spatial information, and thus we only use the single layer feature representation of con4-1 to reduce the impact of detailed spatial information. ω_s^l and ω_c^l are set to 1 and 10 respectively to lay slight importance on scene content preservation. For TN, we do not add domain adaptation at the beginning, but only train the stacked GANs with an alternating update scheme from scratch, i.e., we first update D_d, D_c with G_d, G_c fixed, then G_d, G_c are updated with D_d, D_c fixed. The domain adaptation modules will be aligned after about 100 epochs, when the stacked network has been well trained. Finally, both sub-networks are jointly fine-tuned using the overall loss function. The training details are summarized in Algorithm 1. We used the Adam solver, setting $\beta_1 = 0.5$, $\beta_2 = 0.9$. The learning rate is set to 2×10^{-4}. The weights of different loss components are set to $\lambda_a = 10, \lambda_b = 10, \lambda_t = 10, \lambda_d = 0.1, \lambda_c = 0.1$, respectively.

V. Experimental Results

For synthetic underwater dataset, we choose the public available PSNR and SSIM metrics to assess the color corrected results for all the compared methods, while the metrics of Error (RMSE, Log10, Rel) and Accuracy referred to [33] for depth estimation. The metrics of Log10, Rel and Accuracy all belong to the category of mean absolute difference (MAD), representing the overall recovery level. In contrast, RMSE measures the square dissimilarity between image pairs, and can amplify the errors when there appears obvious wrong estimation in regions with large variation in depth values. For the real underwater dataset, we assess the color corrected

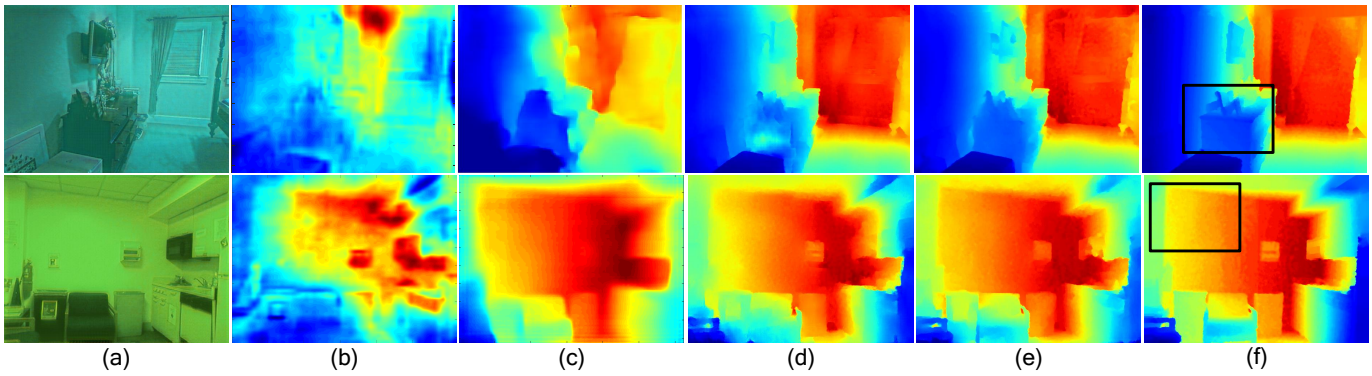


Fig. 6. Qualitative depth estimation comparison on synthetic underwater datasets: (a) Synthetic underwater images; Results recovered by (b) Li *et al.* [10], (c) Laina *et al.* [36], (d) Our DESN, (e) Our DESN with CCSN, (f) Groundtruth. We use black rectangles to direct readers to focus on those specific areas to compare the difference between our two cases.

TABLE I
OBJECTIVE COMPARISON BETWEEN THE PROPOSED DEPTH ESTIMATION MODULE AND OTHER METHODS ON ORIGINAL NYU V2 DATASET (TOP) AND SYNTHETIC UNDERWATER DATASET (BOTTOM).

Method	Error (Lower is better)			Accuracy (Higher is better)		
	RMSE	Log10	Rel	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Eigen *et al.* [33]	0.907	-	0.215	61.1%	88.7%	97.1%
Li *et al.* [35]	0.821	0.094	0.232	62.1%	88.6%	96.8%
Liu *et al.* [34]	0.824	0.095	0.230	61.4%	88.3%	97.1%
Laina *et al.* [36]	0.591	0.063	0.208	81.1%	95.3%	**98.8%**
DESN	**0.584**	**0.061**	**0.157**	**82.0%**	**95.6%**	**98.8%**
Li *et al.* [10]	1.573	0.202	1.994	40.8%	66.9%	80.0%
Laina *et al.* [36]	0.826	0.080	0.161	79.3%	93.0%	97.8%
DESN with DA	0.631	0.088	0.264	74.2%	88.1%	93.1%
DESN	0.456	0.069	0.124	80.3%	93.2%	96.2%
DESN + CCSN	**0.442**	**0.067**	**0.104**	**81.7%**	**94.8%**	**98.6%**

results using underwater color image quality evaluation (U-CIQE) metric [52], and underwater image quality measures (UICM, UISM, UIConM, UIQM) [53] [2]. UIQM consists of UICM, UISM, and UIConM, represents the overall quality of a corrected underwater image, and its sub-components represent color, sharpness, and contrast, respectively. UCIQE evaluates the image quality through the chrominance, saturation, and luminance.

A. Evaluation on SAN

To evaluate the effectiveness of our SAN, we compare our performance with another method, i.e., WaterGAN [10], which can generate synthetic underwater images based on a structured GAN that combines a simple physical underwater imaging model with several convolution blocks. We apply the comparison from both training process stability and rendering accuracy. The generator and discriminator loss curves are shown in Fig. 5(a) and (b), respectively. The training process for WaterGAN is not stable, i.e., the generator branch fluctuates a lot and can not converge. This is because using a simple underwater imaging model with a few convolutions to simulate the complex underwater appearances is not sufficient, which leads to the instable training process. For ours, G and D achieve dynamic equilibrium in the training and G loss is almost converged when approaching the 500_{th} epoch. Fig. 5(c-e) further show visual rendering performances. The results from WaterGAN have obvious color deviation from the true appearance of the real images, while our model obtains the right appearances, verifying that our perceptual style and content loss can help to learn the transformation G_s for domain adaptation to generate realistic synthetic images from the perspective of appearance.

Fig. 5 further shows more rendering results from different training epoches. As the epoches increase, our rendering results get better, and achieve the most similar appearances with those of real underwater images. However, the results from WaterGAN under both epoches have obvious color deviation from the true appearance of the real images. Instable training brings different rendering results in different epoches, which can not tell the best one from all the trained models from different epoches.

[2]There is no groundtruth for evaluation, and thus some no-reference quality assessment tools are employed for color correction. For depth estimation, we will show the visual comparison, and give our explanations.

B. Evaluation on TN

We first separately test the depth estimation module (DESN) and the color correction module (CCSN), and then evaluate the joint architecture. All the cases are tested on the synthetic test dataset. The following compared methods are all trained on the same training dataset for fairly comparison.

For depth estimation, we first evaluate our DESN on original NYU dataset compared with some state-of-the-art monocular depth estimation methods [33]–[36]. Table I (top part) shows the comparison results. Our U-net structure with dense blocks achieves better performance than other methods, which demonstrates the effectiveness of our model to infer the scene depth. Then, we evaluate on synthetic underwater dataset (bottom part in Table I). Our methods also achieve superior results to other methods, and has about 29% improvement (RMSE decreases from 0.826 to 0.442) compared to [36]. Compared between our three cases, we find that 1) The use of domain adaptation (DA for short) will degrade the performance, since the network need to spare some network capacities for the real underwater dataset, other than only

TABLE II
OBJECTIVE COMPARISON BETWEEN DIFFERENT COLOR CORRECTION METHODS ON SYNTHETIC UNDERWATER DATASET.

	NID [24]	FIP [55]	CBF [54]	WaterGAN [10]	CycleGAN [27]	R-Cycle [28]	Pix2pix [26]	CCSN	CCSN+DESN
PSNR	11.80	14.80	17.73	18.20	5.95	20.49	24.65	26.50	**26.62**
SSIM	0.50	0.77	0.86	0.70	0.06	0.78	0.89	0.91	**0.93**

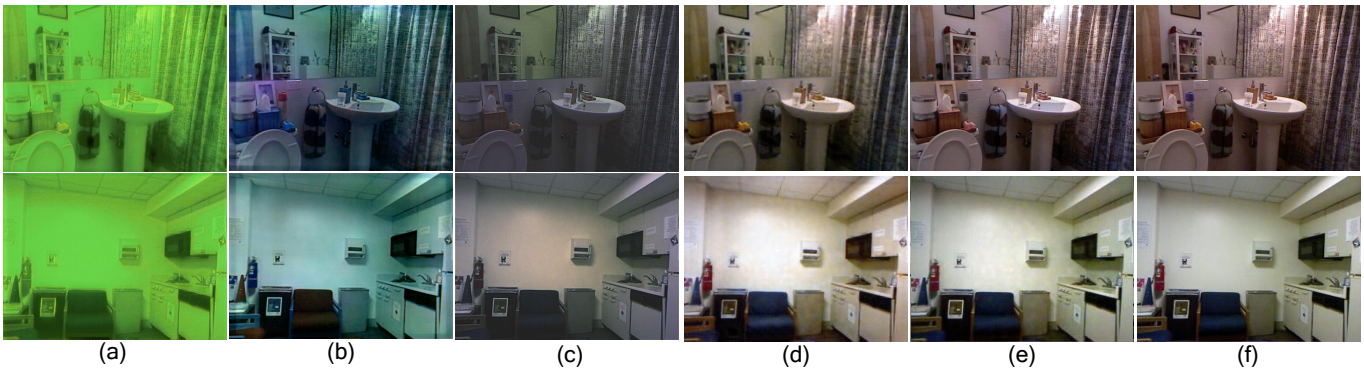


Fig. 7. Qualitative color correction comparison on synthetic underwater datasets: (a) Synthetic underwater images; Results recovered by (b) WaterGAN [10], (c) CBF [54], (d) R-Cycle [28], (e) Our CCSN, (f) Groundtruth.

the synthetic one; [3] 2) Jointly training depth estimation and color correction modules (both without DA) will improve the performance, because the stacked mode aims to learn a shared embedding across tasks by simply aggregating the supervisions from each individual task. Fig. 6 shows the comparison of depth estimation between Li *et al.* [10], Laina *et al.* [36], and our two cases (DESN, DESN + CCSN) on the synthetic test datasets. Li *et al.* cannot recover the correct scene geometry, while the results from Laina *et al.* are subjective to blurry artifacts. Our DESN achieves superior results to other two methods. Jointly training DESN and CCSN will improve the performance, because the stacked mode aims to learn a shared embedding across tasks by simply aggregating the supervisions from each individual task.

For color correction, we evaluate our results by comparing with the following methods: Non-local Image Dehazing [24] (NID), fast image processing algorithm (FIP) [55], color balance and fusion (CBF) [54], CycleGAN [27], the revised CycleGAN with a consistency loss (R-Cycle) [28], Pix2Pix [26] and WaterGAN [10]. Table II gives the quantitative results of color correction. CycleGAN cannot maintain the scene content, leading to bad performance. R-cycle is an improved version of CycleGAN with an extra SSIM loss to keep the scene content invariant, and thus obtains relatively better results. Our methods achieve the best performance than other methods. Training together with our depth estimation module also has some slight improvements. Fig. 7 shows the results of color correction by comparing WaterGAN [10], CBF [54], and R-Cycle [28]. WaterGAN cannot recover the right color, since their failure to obtain an accurate depth map leads to severe color deviation on the corrected image. The overall tone and brightness of CBF are dimmed compared to R-cycle and ours. Compared with other methods, our CCSN achieves the best visual performance, i.e., the most similar color with groundtruth.

[3]Even though DA makes the results on synthetic dataset worse, it will help improve the performance on real underwater dataset. See Sec. 5.3 for details.

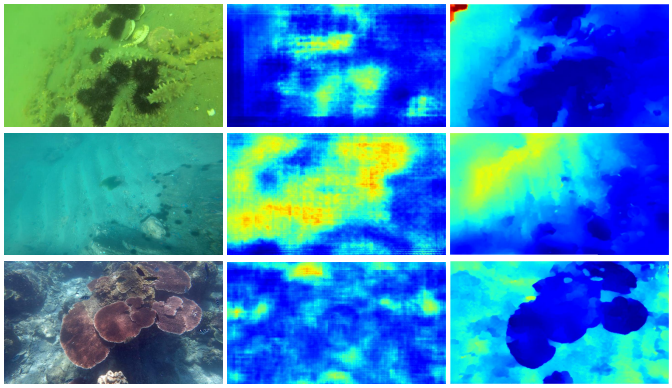


Fig. 8. Qualitative comparison of depth estimation under different training configurations. (a) underwater images; Results estimated from the model trained on (b) original NYU RGB-D dataset and (c) style-transferred NYU dataset, respectively.

C. Ablation Study on Real Underwater Images

Network architecture: To validate the effectiveness of our SAN on the task of depth estimation, we apply another experiment that extracts the learned model trained on original NYU dataset, and directly tests on underwater images. As shown in Fig. 8, the comparison of depth estimation results under different training configurations verifies that using the style-transferred dataset to train the task model brings significant improvement on depth estimation. Besides, for color correction, synthetic images are not available without SAN and thus we cannot train the color correction network, which demonstrates that SAN is indispensable for the whole framework.

To discover the vital element from TN, we conduct ablation study on real underwater dataset by gradually integrating each component into our framework. For depth estimation, we show

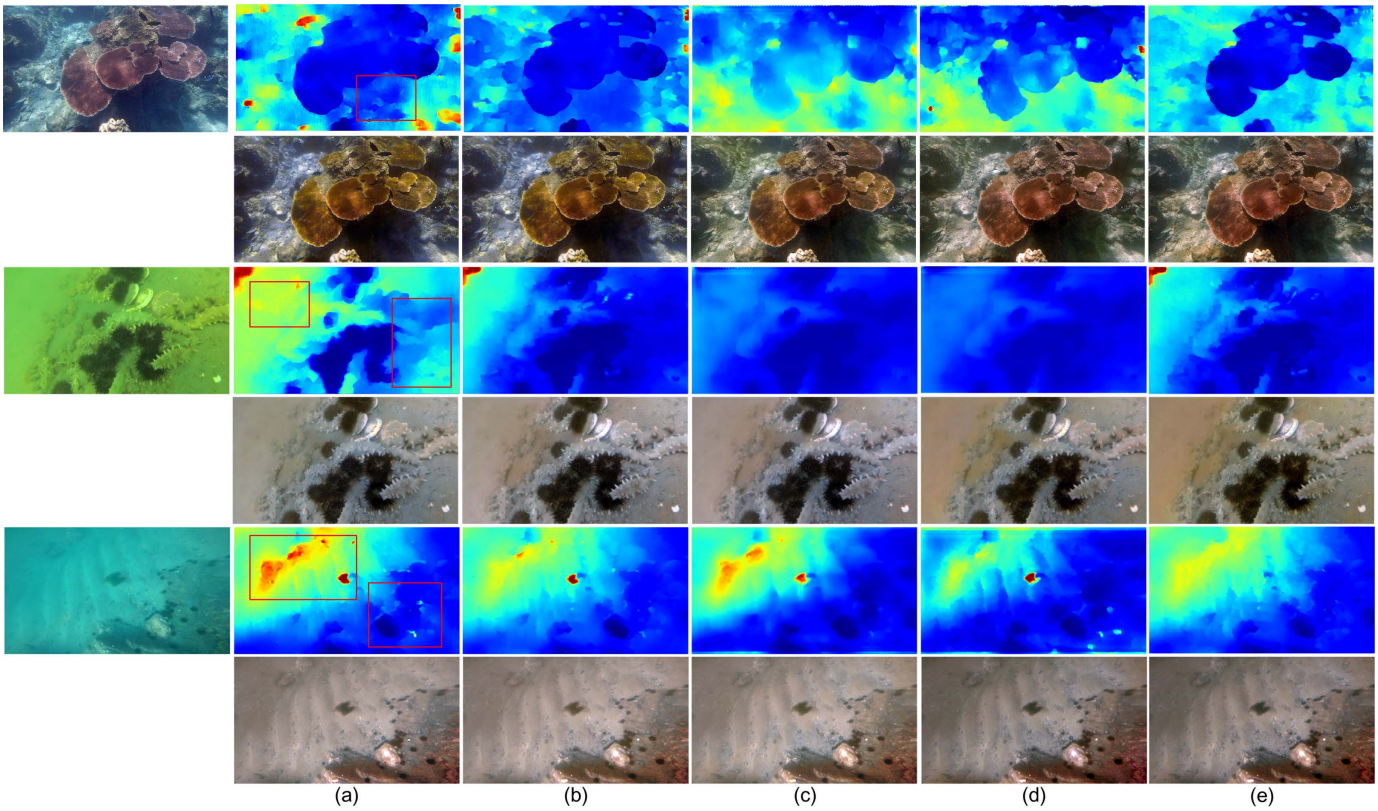


Fig. 9. Qualitative comparison on real underwater images under different module configurations: (a)DESN or CCSN separately; (b) DESN + CCSN, (c) DESN with DA + CCSN, (d) DESN with DA + CCSN with DA, (e) DESN + CCSN with DA. The depth maps are colored with red for farther distance, while blue for closer distance. We use red rectangles to direct readers to focus on those specific areas to compare the difference under different cases.

the visual comparison results in Fig. 9. Only reapplying a model learnt on synthetic in-air data (DESN only) may not generalize well to natural underwater scenes. The shadow regions in the first example are wrongly estimated, and some areas do not gradually change in the region of far distance in another two examples. Training with CCSN can improve the performance, which verifies the effectiveness of our joint learning framework. However, the results are still subjective to some depth outliers and blurring. The following three cases with DA applied on different parts present different results. Applying DA on DESN (Fig. 9(c)(d)) may cause a wrong estimation for far distance, and introduce some blurring artifacts. Therefore, only applying DA on CCSN is the best choice, and generates the right scene geometry and the sharp depth boundaries (Fig. 9(e)). For color correction, the objective results are shown in Table III. Applying DA in the framework can improve the quantitative performance than that without DA, which demonstrates the effectiveness of our DA module. Note that, the best results also appear in the one that applying DA only on CCSN. The similar conclusion can be obtained from the visual results in Fig. 9, i.e, the color is more realistic than other situations which are slightly bluish in the first example and whitish in other examples.

To conclude, our joint learning architecture and domain adaptation applied on CCSN module can help to achieve superior results in both tasks. Note that, although no DA module is applied on DESN, the stacked mode of joint learning can improve the performance on depth estimation via backward information flows.

TABLE III
OBJECTIVE COMPARISON ON REAL UNDERWATER IMAGES UNDER DIFFERENT CONFIGURATIONS. FOR ALL THE METRICS, HIGHER VALUES REPRESENT BETTER PERFORMANCE.

	UISM	UIConM	UICM	UIQM	UCIQE
CCSN	6.96	0.58	6.40	4.15	0.36
DESN + CCSN	6.96	0.56	6.45	4.22	0.41
DESN w/ DA + CCSN	6.96	0.54	6.49	4.18	0.42
DESN w/ DA + CCSN w/ DA	6.97	0.56	6.66	4.25	0.42
DESN + CCSN w/ DA	**6.98**	**0.59**	**10.75**	**4.47**	**0.44**

Loss terms: For SAN, the corresponding loss $\mathcal{L}_{SAN}$ can be divided into the basic GAN loss and the perceptual loss. The key point to generate satisfying style-transferred results is to use the perceptual loss that is composed by content loss and style loss to achieve content preservation and style transfer simultaneously. As shown in Fig. 10, the generated results without perceptual loss almost fail which demonstrates that the perceptual loss is essential, which can be regarded as a guidance for the process of style transfer. For TN, the overall loss $\mathcal{L}_{TN}$ can be divided into three aspects, i.e., the task loss $\mathcal{L}_t$, GAN loss $\mathcal{L}_{adv}$ and feature adaptation loss $\mathcal{L}_f$. $\mathcal{L}_t$ is a basic loss that supervises the training process of the whole task network. The stacked GAN losses $\mathcal{L}_{adv_d}$ and $\mathcal{L}_{adv_c}$ aim to make the predicted depth maps and corrected color images more realistic. As shown in Fig. 11, the result without stacked GAN loss presents unsmoothed depth values

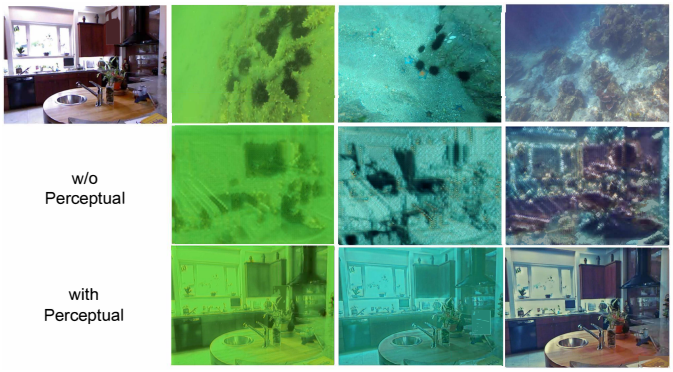


Fig. 10. Visual comparison of generated images without or with perceptual loss. Original in-air image and three underwater images with different styles are shown in the first row. The generated results without or with perceptual loss are shown in second and third rows, respectively.

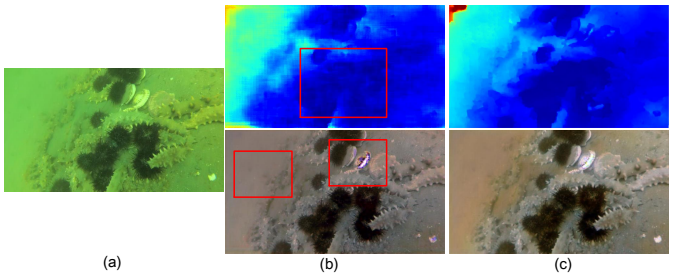


Fig. 11. Visual comparison of generated images without or with stacked GAN loss. (a) underwater image; (b) the result without GAN loss; (c) the result with GAN loss.

and unsatisfactory colors in the highlighted red rectangles. The remaining feature adaptation loss aligning with the feature adaptation module has been verified in Fig. 9 and Table III. Therefore, each component of the overall loss contributes to the final performance, and their combination achieves the optimal results.

D. Comparisons with State-of-the-Art

Fig. 12 shows the visual comparison of depth estimation between Laina *et al.* [36] and ours. For fairly comparison, we retrain the network of Laina *et al.* on our synthetic underwater dataset. The results of Laina *et al.* cannot generalize to real underwater images, and almost fail to infer the right scene geometry, while our method obtains the accurate depth map that reflects the real scene structures.

For color correction, Table IV shows the objective comparison between different methods. NID and WaterGAN fail to recover the right color of the scenes, leading to very low metric values. R-cycle and Pix2Pix obtain relatively good results, but are inferior to ours for most metrics. Fig. 13 shows the visual comparison of color correction between FIP [55], CBF [54], R-cycle [28], Pix2Pix [26], and Ours. FIP [55] cannot recover the right color for all the three underwater scenes, especially for the second two example (some checkerboard artifacts appear.). CBF still presents excessive dim results especially on the first four examples, and generates greenish results on the last two examples. R-cycle [28] and Pix2Pix [26] presents similar results: 1) The first two example presents excessively

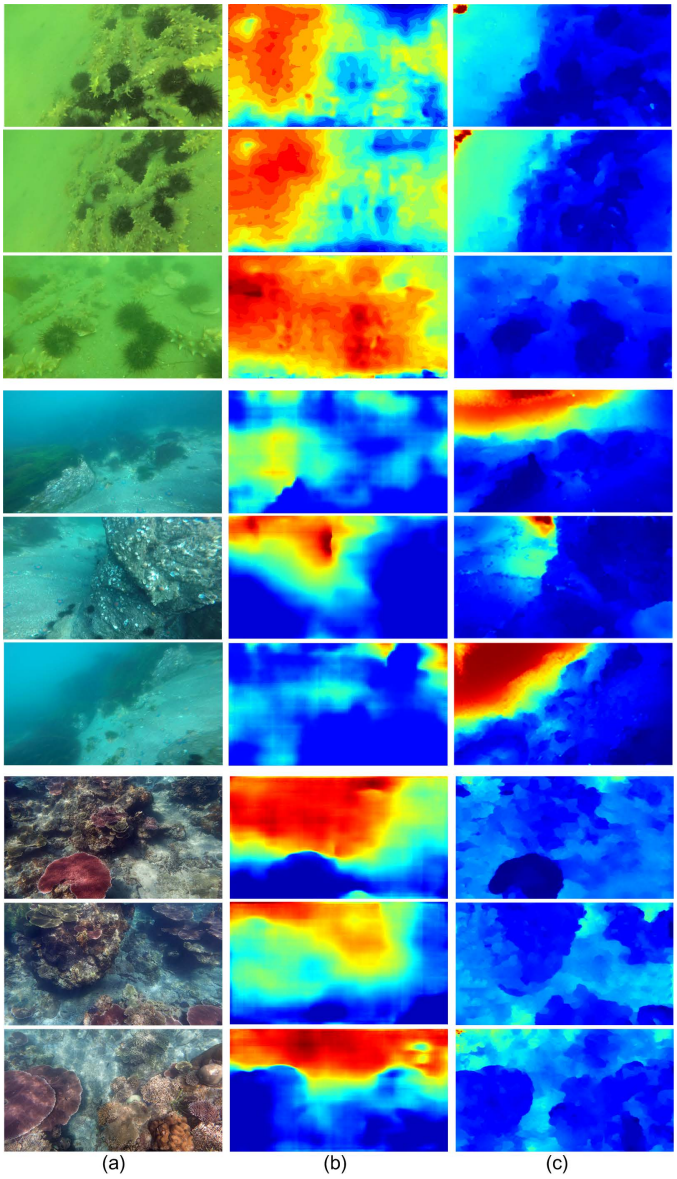


Fig. 12. Qualitative comparison of depth estimation on real underwater images: (a) underwater images; Results estimated by (b) Laina *et al.* [36], and (c) Ours.

yellowish for both methods, e.g., the color deviation of the sea cucumber; 2) The blue color of far distance in the fourth example cannot be removed; 3) The presentation of color saturation and image sharpness are not as good as our method. In summary, our method achieves the realistic vision, balanced color, and sharp recovered results.

Besides, runtime is measured on a workstation with an Intel Xeon CPU E5 2.40GHz and an Nvidia Titan X GPU. Our test images are of the size of 640×480 (about 0.29MP). We separately test our color correction subnetwork (CCSN) for fair comparison[4]. The objective computation runtime (ms) for different network architectures is shown in Fig. V. The parameterization of FIP is extremely compact, but its performance is significantly lower than other methods. R-cycle and

[4]Our DESN and CCSN have the same network architecture, and hence the same runtime and parameters.

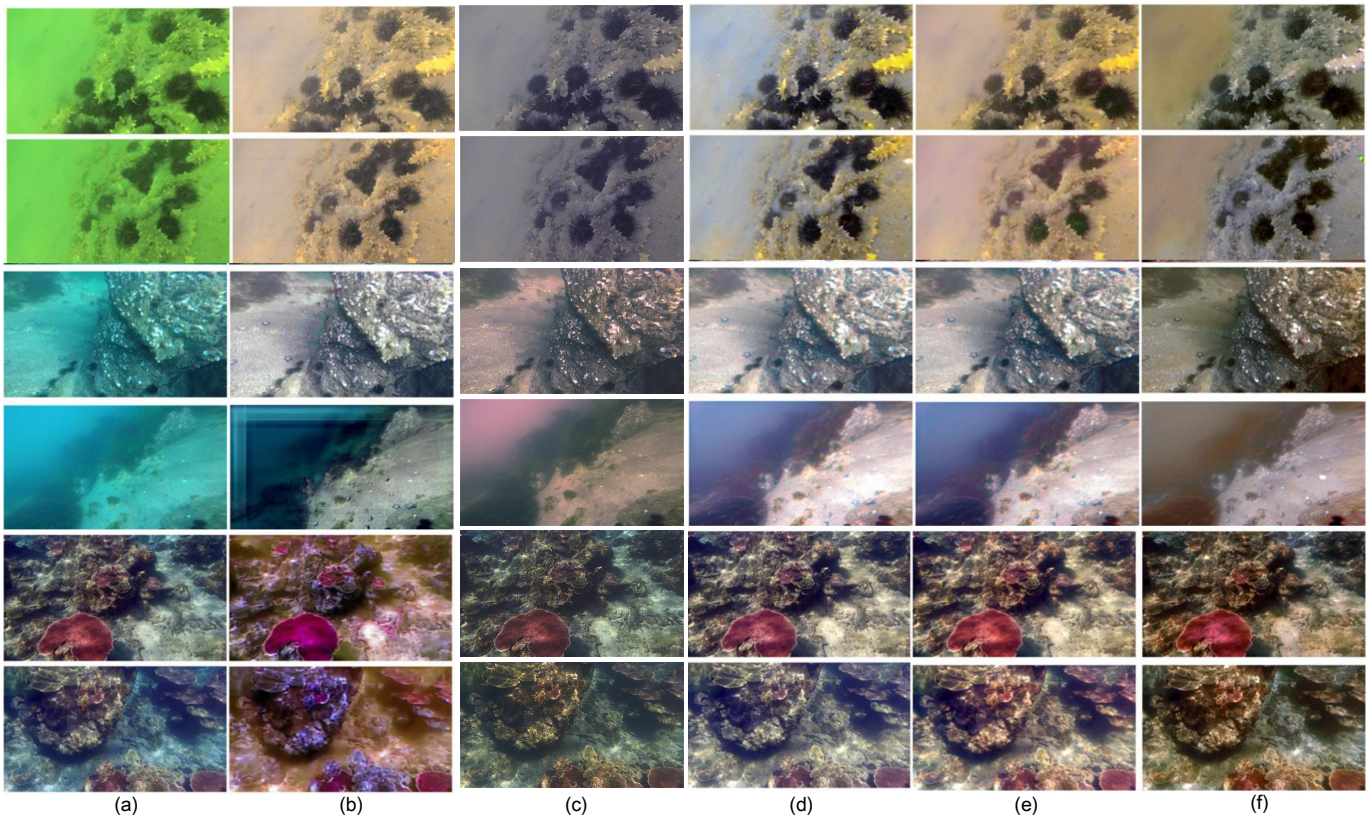


Fig. 13. Qualitative comparison of color correction on real underwater images: (a) underwater images. Results recovered by (b) FIP [55], (c) CBF [54], (d) R-cycle [28], (e) Pix2Pix [26], and (f) Ours.

TABLE IV
OBJECTIVE COMPARISON OF COLOR CORRECTION ON REAL UNDERWATER IMAGES. FOR ALL THE METRICS, HIGHER VALUES REPRESENT BETTER PERFORMANCE.

	UISM	UIConM	UICM	UIQM	UCIQE
NID [24]	2.70	0.49	-28.92	1.73	0.32
FIP [55]	6.96	0.46	25.21	4.41	0.28
CBF [54]	3.22	0.56	2.21	3.39	0.33
R-Cycle [28]	6.54	0.51	10.78	4.06	0.40
Pix2Pix [26]	6.75	0.44	**26.95**	4.33	0.34
WaterGAN [10]	6.88	0.54	-43.63	2.72	0.30
ours	**6.98**	**0.59**	10.75	**4.47**	**0.44**

TABLE V
OBJECTIVE COMPARISON OF RUNTIME AND TOTAL NUMBER OF PARAMETERS IN TESTING PHASE BETWEEN DIFFERENT NETWORK ARCHITECTURES. RUNNING TIME IS MEASURED ON IMAGES AT 480P RESOLUTION (ABOUT 0.29 MP).

Methods	FIP [55]	CycleGAN [27]	R-Cycle [28]
Runtime(ms)	58.9	103.3	103.3
# of param	75K	11,365K	11,365K
Methods	Pix2Pix [26]	WaterGAN [10]	Ours
Runtime(ms)	99.7	92.3	87.6
# of param	54,408K	29,469K	2768K

CycleGAN have the same architecture and hence the same runtime and parameters. Our method achieves lower runtime and fewer parameters but better performance than Pix2Pix and R-cycle, which demonstrates our effectiveness.

We further show a statistical analysis of all the compared methods in Table IV for better explaining the statistical significance of our results. We count the objective performance on each image in each underwater dataset under UCIQE metric statistically and present the box diagrams in Fig. 14. Obviously, our method is superior to other method on average, and achieves the most stable performance.

VI. LIMITATION

Our framework still presents some limitations and can be solved in the future work. First, the depth estimation accuracy still has much room for improvement. If the foreground objects, e.g., the blackish sea urchin in Fig. 11, present obvious incongruous color with the overall underwater tone, the estimated depth values for these objects are also affected. For example, although the relative depth of the sea urchin to the ground can be estimated, the accuracy is not high, since the absolute distance from the object to the ground is not as far as estimated. Therefore, we need to use extra multi-modal information, e.g., contour prediction and semantic parsing, as the guidance to facilitate the depth estimation task. Besides, the performance of depth estimation on monocular images alone is somewhat limited. When considering spatial and temporal correlation together on available underwater videos, the performance can be further improved.

Besides, the ability of SAN is not enough to mimic all the negative underwater effects. The key point we address is the content preservation and the general appearance colors

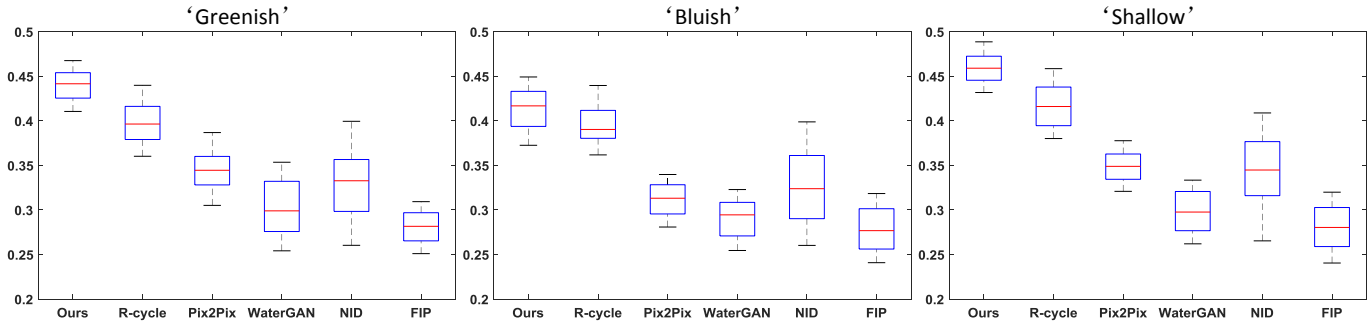


Fig. 14. Box diagrams to present the statistical significance of our methods. We count the objective performance on each image in each underwater dataset under UCIQE metric statistically.

transfer. Other effects, e.g., blurring or non-smooth lighting which are also very important degradations in the underwater domain, are ignored. For example, some blurring artifacts appear in the color correction results (the 1st, 2rd and 4th example in Fig.13); the ripple artifacts caused by refraction of non-smooth lighting (the last three examples in Fig. 12) interfere with the smoothness of local region in the predicted depth map, which appears some stratified cotton-shaped artifacts especially in the farther regions. In future work, we will fully address these problems by designing more sophisticated style transfer network to mimic the effects and introducing some pro-processing modules that can be trained together with existing network to further enhance the results.

Finally, our method mainly focuses on the improvement of algorithm performance, but current processing frame rate (about 12 fps) cannot meet the requirements of real-time underwater depth estimation and color correction. Therefore, resorting to network pruning and recent distillation techniques, we will focus on reducing the computation delay of the model without compromising the performance in future work.

VII. Conclusion

We propose an unsupervised adaptation network to jointly estimate scene depth and correct the color from a single underwater image. We consider the problem from a novel perspective of style-level and feature-level adaptation to address the lack of effective underwater training data, and the weak generalization to the real-world underwater images trained on synthetic data. Extensive experiments show that the proposed method performs favorably against state-of-the-art methods in both depth estimation and color correction tasks.

References

[1] S. T. Digumarti, G. Chaurasia, A. Taneja, R. Siegwart, A. Thomas, and P. Beardsley, Underwater 3d capture using a low-cost commercial depth camera, in Applications of Computer Vision, 2016, pp. 1-9.

[2] J. Sattar and G. Dudek, Where is your dive buddy: tracking humans underwater using spatio-temporal features, in IEEE/RSJ International Conference on Intelligent Robots and Systems, 2007, pp. 3654-3659.

[3] P. L. Negre, F. Bonin-Font, and G. Oliver, Cluster-based loop closing detection for underwater slam in feature-poor regions, in IEEE ICRA, 2016, pp. 2589-2595.

[4] J. S. Jaffe, Underwater optical imaging: The past, the present, and the prospects, IEEE Journal of Oceanic Engineering, vol. 40, no. 3, pp. 683-700, 2015.

[5] C. Ancuti, C. O. Ancuti, T. Haber, and P. Bekaert, Enhancing underwater images and videos by fusion, in IEEE CVPR, 2012, pp. 81-88.

[6] J. Y. Chiang and Y. C. Chen, Underwater image enhancement by wavelength compensation and dehazing. IEEE TIP, vol. 21, no. 4, pp. 1756-1769, 2012.

[7] S. Zhang, J. Zhang, S. Fang, and Y. Cao, Underwater stereo image enhancement using a new physical model, in IEEE ICIP, 2015, pp. 5422-5426.

[8] Y. S. Shin, Y. Cho, G. Pandey, and A. Kim, Estimation of ambient light and transmission map with common convolutional architecture, in Oceans, 2016, pp. 1-7.

[9] Y. Wang, J. Zhang, Y. Cao, and Z. Wang, A deep cnn method for underwater image enhancement, in IEEE ICIP, 2017, pp. 1382- 1386.

[10] J. Li, K. A. Skinner, R. M. Eustice, and M. Johnson-Roberson, Watergan: Unsupervised generative network to enable real-time color correction of monocular underwater images, IEEE Robotics and Automation Letters, vol. 3, no. 1, pp. 387-394, 2018.

[11] L. Mazzei, L. Corgnati, S. Marini, and E. Ottaviani, Low cost stereo system for imaging and 3d reconstruction of underwater organisms, in Oceans, 2015, pp. 1-4.

[12] H. Lu, Y. Zhang, Y. Li, Q. Zhou, R. Tadoh, T. Uemura, H. Kim, and S. Serikawa, Depth map reconstruction for underwater kinect camera using inpainting and local image mode filtering, IEEE Access, vol. PP, no. 99, pp. 1-1, 2017.

[13] F. Oleari, F. Kallasi, D. L. Rizzini, and J. Aleotti, An underwater stereo vision system: From design to deployment and dataset acquisition, in Oceans, 2015, pp. 1-6.

[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 IEEE CVPR, 2017, pp. 568-577.

[15] D. Akkaynak and T. Treibitz, A revised underwater image formation model, in IEEE CVPR, 2018.

[16] J. S. Jaffe, Computer modeling and the design of optimal underwater imaging systems, IEEE Journal of Oceanic Engineering, vol. 15, no. 2, pp. 101-111, 1990.

[17] J. Yang and et.al, Color-guided depth recovery from RGB-D data using an adaptive autoregressive model, IEEE TIP, vol. 23, no. 8, pp. 3443-3458, 2015.

[18] J. Yang, X. Ye, and P. Frossard, Global auto-regressive depth recovery via iterative non-local filtering, IEEE Trans. Broadcasting, vol. PP, no. 99, pp. 1-15, 2018.

[19] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, Generative adversarial nets, in NIPS, 2014, pp. 2672-2680.

[20] K. Iqbal, Underwater image enhancement using integrated contrast correction and white balance methods. 2009.

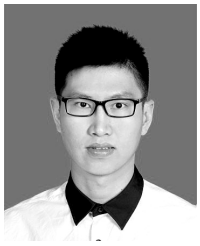
[21] E. Provenzi, C. Gatta, M. Fierro, and A. Rizzi, A spatially variant whitepatch and gray-world method for color image enhancement driven by local contrast, IEEE TPAMI, vol. 30, no. 10, pp. 1757-1770, 2008.

[22] C. Y. Li, J. C. Guo, R. M. Cong, Y. W. Pang, and B. Wang, Underwater image enhancement by dehazing with minimum information loss and histogram distribution prior, IEEE TIP, vol. 25, no. 12, pp. 5664-5677, 2016.

[23] Y. Y. Schechner and Y. Averbuch, Regularized image recovery in scattering media, IEEE TPAMI, vol. 29, no. 9, pp. 1655-1660, 2007.

[24] D. Berman, T. Treibitz, and S. Avidan, Non-local image dehazing, in IEEE CVPR, 2016, pp. 1674-1682.

[25] H. Zhang and V. M. Patel, Densely connected pyramid dehazing network, in IEEE CVPR, 2018.
[26] P. Isola, J. Zhu, T. Zhou, and A. A. Efros, Image-to-image translation with conditional adversarial networks, in IEEE CVPR, July 2017, pp. 5967-5976.
[27] J. Zhu, T. Park, P. Isola, and A. A. Efros, Unpaired image-to-image translation using cycle-consistent adversarial networks, in IEEE ICCV, Oct 2017, pp. 2242-2251.
[28] C. Li, J. Guo, and C. Guo, Emerging from water: Underwater image color correction based on weakly supervised color transfer, IEEE Signal Processing Letters, vol. 25, no. 3, pp. 323-327, 2018.
[29] V. Badrinarayanan, A. Kendall, and R. Cipolla, Segnet: A deep convolutional encoder-decoder architecture for scene segmentation, IEEE TPAMI, vol. PP, no. 99, pp. 1-1, 2015.
[30] A. Saxena, S. H. Chung, and A. Y. Ng, Learning depth from single monocular images, in NIPS, 2005, pp. 1161-1168.
[31] K. Karsch, C. Liu, and S. B. Kang, Depth transfer: Depth extraction from video using non-parametric sampling, IEEE TPAMI, vol. 36, no. 11, p. 2144, 2014.
[32] S. Choi, D. Min, B. Ham, Y. Kim, C. Oh, and K. Sohn, Depth analogy: Data-driven approach for single image depth estimation using gradient samples, IEEE TIP, vol. 24, no. 12, p. 5953, 2015.
[33] D. Eigen and R. Fergus, Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture, in IEEE ICCV, 2015, pp. 2650-2658.
[34] F. Liu, C. Shen, G. Lin, and I. Reid, Learning depth from single monocular images using deep convolutional neural fields, IEEE TPAMI, vol. 38, no. 10, pp. 2024-2039, 2016.
[35] B. Li, C. Shen, Y. Dai, A. van den Hengel, and M. He, Depth and surface normal estimation from monocular images using regression on deep features and hierarchical crfs, in IEEE CVPR, 2015, pp. 1119- 1127.
[36] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab, Deeper depth prediction with fully convolutional residual networks, in Fourth International Conference on 3d Vision, 2016, pp. 239-248.
[37] J.-H. Lee, M. Heo, K.-R. Kim, and C.-S. Kim, Single-image depth estimation based on fourier domain analysis, in IEEE CVPR, 2018.
[38] K. He, X. Zhang, S. Ren, and J. Sun, Deep residual learning for image recognition, in IEEE CVPR, 2016, pp. 770-778.
[39] G. Csurka, Domain adaptation for visual applications: A comprehensive survey, arXiv preprint arXiv:1702.05374, 2017.
[40] J. N. Kundu, P. K. Uppala, A. Pahuja, and R. V. Babu, Adadepth: Unsupervised content congruent adaptation for depth estimation, in IEEE CVPR, 2018.
[41] J. Hoffman, E. Tzeng, T. Park, J.-Y. Zhu, P. Isola, K. Saenko, A. A. Efros, and T. Darrell, Cycada: Cycle-consistent adversarial domain adaptation, arXiv preprint arXiv:1711.03213, 2017.
[42] Y. H. Tsai, W. C. Hung, S. Schulter, K. Sohn, M. H. Yang, and M. Chandraker, Learning to adapt structured output space for semantic segmentation, in IEEE CVPR, 2018.
[43] C. Ledig, Z. Wang, W. Shi, L. Theis, F. Huszar, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, and A. Tejani, Photo-realistic single image super-resolution using a generative adversarial network, in IEEE CVPR, 2017, pp. 105-114.
[44] L. A. Gatys, A. S. Ecker, and M. Bethge, Image style transfer using convolutional neural networks, in IEEE CVPR, 2016.
[45] D. Ulyanov, V. Lebedev, A. Vedaldi, and V. Lempitsky, Texture networks: Feed-forward synthesis of textures and stylized images, in ICML, 2016.
[46] O. Ronneberger, P. Fischer, and T. Brox, U-net: Convolutional networks for biomedical image segmentation, in MICCAI, 2015, pp. 234-241.
[47] G. Huang, Z. Liu, L. V. D. Maaten, and K. Q. Weinberger, Densely connected convolutional networks, in IEEE CVPR, 2017, pp. 2261- 2269.
[48] L. C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs, IEEE TPAMI, vol. PP, no. 99, pp. 1-1, 2017.
[49] R. Liu, X. Fan, M. Zhu, M. Hou, and Z. Luo, Real-world underwater enhancement: Challenges, benchmarks, and solutions, arXiv preprint arXiv: 1901.05320, 2019.
[50] C. Li, C. Guo, W. Ren, R. Cong, J. Hou, S. Kwong, and D. Tao, An underwater image enhancement benchmark dataset and beyond, arXiv preprint arXiv: 1901.05495, 2019.
[51] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, Indoor segmentation and support inference from rgbd images, in ECCV, 2012, pp. 746-760.
[52] M. Yang and A. Sowmya, An underwater colour image quality evaluation metric. IEEE TIP, vol. 24, no. 12, pp. 6062-6071, 2015.
[53] K. Panetta, C. Gao, and S. Agaian, Human-visual-system-inspired underwater image quality measures, IEEE Journal of Oceanic Engineering, vol. 41, no. 3, pp. 541-551, 2016.
[54] C. O. Ancuti, C. Ancuti, V. C. De, and P. Bekaert, Color balance and fusion for underwater image enhancement, IEEE TIP, vol. 27, no. 1, pp. 379-393, 2018.
[55] Q. Chen, J. Xu, and V. Koltun, Fast image processing with fully-convolutional networks in IEEE ICCV, Oct 2017, pp. 2516-2525



Xinchen Ye (M'17) received the B.E. degree and Ph.D. degree from the Tianjin University, Tianjin, China, in 2012 and 2016, respectively. He was with the Signal Processing Laboratory, EPFL, Lausanne, Switzerland in 2015 under the Grant of the Swiss federal government. He has been a Faculty Member of Dalian University of Technology, Dalian, Liaoning, China, since 2016, where he is currently a Assistant Professor with the DUT-RU International School of Information Science and Engineering. His current research interests include image/video processing and 3D imaging. As a co-author, he received the Platinum Best Paper Award in the IEEE ICME 2017. He won the Rising Star Award in 2018 ACM Turing Celebration Conference-China (ACM TURC 2018).



Zheng Li is currently a undergraduate at the School of Software in Dalian University of Technology. His research interests include computer vision and deep learning.



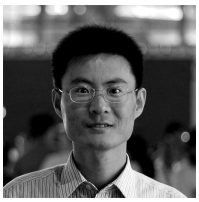
Baoli Sun received the B.S degree in microelectronics science and engineering in 2018 from the Hefei University of Technology , Anhui , China. He is currently a graduate student at the school of Software in Dalian University of Technology in Liaoning, China. His research interests include image processing, computer vision and deep learning.



Zhihui Wang received the B.S. degree in software engineering in 2004 from the North Eastern University, Shenyang, China. She received her M.S. degree in software engineering in 2007 and the Ph.D degree in software and theory of computer in 2010, both from the Dalian University of Technology, Dalian, China. Since November 2011, she has been a visiting scholar of University of Washington. Her current research interests include information hiding and image compression.



Rui Xu received the B.S. and M.S. degrees in 2001 and 2004, respectively from the school of electronic and information, South China University of Technology. He received the Ph.D. degree in 2007 from the graduate school of science and engineering, Ritsumeikan University, Japan. He worked in the digital technology research center, Sanyo Electric Co., Ltd., Japan, from 2008 to 2010. He worked as a senior researcher successively in Yamaguchi University and Ritsumeikan University from 2010 to 2015. Since December 2015, he moved back to China and served as an associate professor at Dalian University of Technology. His research fields include intelligent computing in medical images and computer vision.



Haojie Li is a Professor in the School of Software, Dalian University of Technology. His research interests include social media computing and multimedia information retrieval. Dr. Li received the B.E. and the Ph. D. degrees from Nankai University, Tianjin and the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, in 1996 and 2007 respectively. From 2007 to 2009, he was a Research Fellow in the School of Computing, National University of Singapore.



Xin Fan received the B.E. and Ph.D. degrees in information and communication engineering from Xian Jiaotong University, Xian, China, in 1998 and 2004, respectively. He was with Oklahoma State University, Stillwater, from 2006 to 2007, as a postdoctoral research Fellow. He joined the School of Software, Dalian University of Technology, Dalian, China, in 2009. His current research interests include computational geometry and machine learning.