

# Underwater Image Enhancement with Lightweight Cascaded Network

Nanfeng Jiang, Weiling Chen, *Member, IEEE*, Yuting Lin, Tiesong Zhao, *Senior Member, IEEE*, and Chia-Wen Lin, *Fellow, IEEE*

**Abstract**—Due to light scatter and absorption in waterbody, underwater imaging can be easily impaired with low contrast and visual distortion. The resulting images are often unable to meet the quality requirements of human perception and computer processing. Therefore, Underwater Image Enhancement (UIE) has been attracting extensive research efforts. Although deep learning has demonstrated its great success in many vision tasks, its huge amounts of parameters and computations are not conducive to UIE in resource-limited scenarios. In this paper, we address this issue by proposing a Lightweight Cascaded Network (LCNet) based on Laplacian image pyramids. At each pyramid level, we implement cascaded blocks upon a residual network. Specifically, high quality residuals can be progressively predicted with significantly reduced complexity in a coarse-to-fine fashion. Furthermore, these sub-networks are recursively nested to build our LCNet, thereby reducing the overall computational complexity with reused parameters. Extensive experiments demonstrate that the proposed method performs favorably against the state-of-the-arts in terms of visual quality, model parameters and complexity.

**Index Terms**—Underwater images, Image enhancement, Oceanic image processing.

## I. INTRODUCTION

Underwater Image Enhancement (UIE) is an important low-level image manipulation technique in oceanic tasks. Owing to light scattering, absorption effects and color variations, underwater images often suffer from quality degeneration, as shown in Fig. 1-(a). These distorted or blurred images might become an obstacle of computer vision tasks, such as autonomous navigation [1], vehicle tracking [2] and marine environmental surveillance [3]. In such case, visual enhancement of distorted images are essential to research in ocean environment. UIE techniques are thereby required to facilitate these tasks. It can be observed from Fig. 1-(b) that the images after UIE are with enhanced visual quality, which may help succeeding image processing or vision tasks.

This work is supported by the National Natural Science Foundation of China (Grant 61901119). (\*Corresponding author: Tiesong Zhao, t.zhao@fzu.edu.cn)

N. Jiang, W. Chen and Y. Lin are with the Fujian Key Laboratory for Intelligent Processing and Wireless Transmission of Media Information, College of Physics and Information Engineering, Fuzhou University, Fuzhou 350108, China. (emails: {201110003, weiling.chen, N181120063}@fzu.edu.cn)

T. Zhao is with the Fujian Key Laboratory for Intelligent Processing and Wireless Transmission of Media Information, College of Physics and Information Engineering, Fuzhou University, Fuzhou 350108, China, and also with the Peng Cheng Laboratory, Shenzhen 518055, China (e-mail: t.zhao@fzu.edu.cn)

C.-W. Lin is with the Department of Electrical Engineering, National Tsing Hua University, Hsinchu 30013, Taiwan, and also with the Institute of Communications Engineering, National Tsing Hua University, Hsinchu 30013, Taiwan (email: cwlin@ee.nthu.edu.tw)

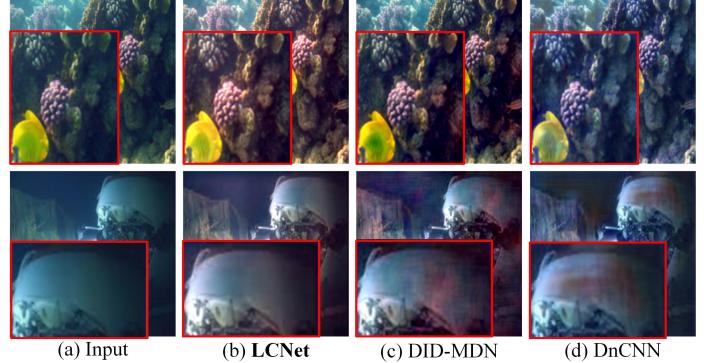


Fig. 1: Examples of UIE tasks. (a) The raw underwater images, (b)-(d) enhanced results by our model, the popular deraining method DID-MDN and denoising method DnCNN.

In the past decade, visual restoration and enhancement algorithms have been benefited from large-scale training data and deep Convolutional Neural Networks (CNNs). These methods include deraining [4]–[6], dehazing [7]–[9], denoising [10], etc. However, UIE is different from these popular low-level tasks such as denoising/deraining in the following aspects. First, underwater images present different types of visual appearances. In popular denoising problems, distortions are generated by additional noises or unwanted objects; while in UIE, distortions are generated by underwater imaging under unsatisfactory lighting conditions. As a result, we can still observe original scenes covered by additional noises; however, this does not work well for underwater scenes with visual range and contrast loss. Second, assessing UIE images require different quality evaluation metrics. It is well known that in visual quality assessment, collecting the original references of the restored/enhanced images is a critical issue. With the high difficulty to recover an original reference for an underwater image, the quality assessment of UIE images differs from that of conventional images. Therefore, popular UIE quality metrics, such as Underwater Color Image Quality Evaluation (UCIQE) [11] and Underwater Image Quality Measure (UIQM) [12], are specially designed for underwater images. Third, due to the above differences, existing denoising/deraining methods show inferior performances for the UIE task. An example is shown in Fig. 1, where the denoising method DnCNN [10] generates color cast artifacts while the deraining method DID-MDN [4] generates over-saturation. As a result, there is still a need to develop effective UIE

algorithms.

Until now, typical UIE algorithms have been developed based on statistical models. The methods proposed in [13]–[15] improve visual quality or calibrate color casts by combining traditional enhancement models with latent parameter estimation. Recently, deep CNNs have shown their strengths in image restoration. In these algorithms, large-sized parameters and memory footprints are usually required to construct complicated network layers and stages. However, self-guiding and self-powered marine vehicles, such as Autonomous Underwater Vehicles (AUVs) and Remotely Operated Vehicles (ROVs), are usually with limited computing power and storage that are not sufficient to support comprehensive computing locally. On the other hand, an Underwater Acoustic Channel (UAC) is also known to offer too poor-quality communication to support cloud-based image processing. In this case, existing methods are often not applicable owing to their heavy computational requirements.

To bridge the gap between existing UIE approaches and real-world applications, we exploit the potential of lightweight network that benefits both effectiveness and efficiency for the UIE task. Recently, many low-level tasks (*e.g.*, LapSRN [16] for single image super-resolution, LPNet [17] for single image rain removal) use Laplacian pyramid to build their lightweight architecture and achieve promising results. Motivated by them, we inherit the advantages of Laplacian Pyramid and propose a Lightweight Cascaded Network (LCNet) to enhance underwater images effectively and efficiently. LCNet is an enhanced version of LPNet [17] with the following improvements. First, LCNet takes all-rounded advantages of the correlated information across different scales. Second, at each pyramid level, we build a cascaded sub-network based on several recursive blocks, which can make full use of the hierarchical features via global connections. Third, in each block, we conduct residual learning in a recursive manner to improve performance with reused model parameters.

In summary, the major contributions of this paper are as follows:

- We propose an LCNet to alleviate undesired underwater color degradation and reduce the network parameters through image pyramids and parameters reuse. Besides, this model is able to learn complicate feature maps at different scales and improve model robustness.
- At each pyramid level, we design a cascaded sub-network, which effectively handles different types of underwater images with a recursive strategy and a coarse-to-fine image residual estimation. We also introduce perceptual loss to constrain the network training.
- The proposed model achieves an excellent balance among visual quality, inference speed, model parameters and complexity. As demonstrated by extensive experiments, it could benefit the object detection in underwater environment.

## II. RELATED WORK

Existing UIE methods can be classified into three categories: non-physics model-based, physics model-based and deep learning-based methods.

### A. Non-physics Model-Based Methods

Non-physics model-based methods adjust pixel values to improve image quality without modeling mathematical formulations [18]–[23]. In [18], Iqbal *et al.* designed a contrast correction model to remove the bluish and red color. Ahmad *et al.* [19] followed the Rayleigh distribution within certain range to remove extra artifacts. Ancuti *et al.* [20] proposed a UIE solution built on the multi-scale fusion principle. After that, Ancuti *et al.* [24] also introduced a novel approach to remove the haze in underwater, improving the visual performance. Ancuti *et al.* [25] designed a UIE approach that estimates the back-scattered light locally. Recently, Ancuti *et al.* [26] proposed a Color Channel Compensation (3C) approach, which effectively enhances in terms of color appearance. Fu *et al.* [21] proposed a two-step UIE procedure, including a color correction and a contrast enhancement. Moreover, several works [22], [23] of UIE were based on retinex theory, Fu *et al.* [22] proposed a variational framework to decompose the reflectance and illumination, which are utilized in further enhancement. Zhang *et al.* [23] proposed an extended multi-scale retinex-based UIE method, which processed the underwater images in CIELAB color space. These methods perform well, but suffer from low contrast, detail loss and color deviations due to the complexity of underwater environment and limited observations [27].

### B. Physics Model-Based Methods

Instead of pixel-to-pixel mapping, the physics model-based methods explicitly characterize the physical imaging process, and estimate the imaging parameters from observations and various prior assumptions [14], [28]–[33]. Chiang *et al.* [14] proposed to enhance underwater images by combining the Dark Channel Prior (DCP) [28] with wavelength-dependent compensation algorithm. In [29], an Underwater DCP (UDCP) was proposed based on the fact that the information of red channel is undependable in an underwater image. Peng *et al.* [30] proposed a Generalized DCP (GDCP) that estimated ambient light using the depth-dependent color change. Recently, Berman *et al.* [34] proposed a method to recover the distance maps and object colors in scenes taken under water and under ambient illumination. In addition, Li *et al.* [31] proposed to combine an underwater image dehazing model with contrast enhancement. More recently, Akkaynak *et al.* [32] designed an underwater image color correction method based on a physically accurate model. In summary, the non-physics and physics model-based methods depended on various observations and prior assumptions. Therefore, these algorithms cannot be well generalized to some specific scenes [30].

### C. Deep Learning-Based Methods

Very recently, deep CNNs have promoted the development of computer vision and image processing [7], [35]–[42]. As for the UIE, deep learning-based methods also led to a fast development and achieved state-of-the-art performance [43]–[48]. In [45], Li *et al.* employed the Generative Adversarial Network (GAN) to build a WaterGAN model, which incorporated the process of underwater image formation to generate

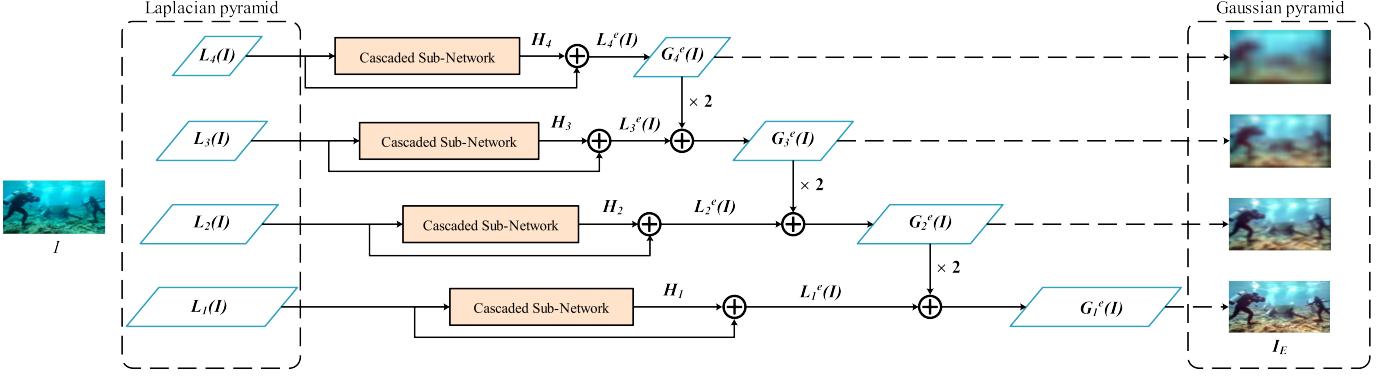


Fig. 2: Architecture of the proposed LCNet. It achieves multi-scale image enhancement with three steps: pyramid decomposition, multi-scale enhancement and image reconstruction.

high resolution outputting images. Guo *et al.* [44] proposed a multi-scale Dense-GAN, which improves UIE by learning the distribution of ground truth images in the feature domain. Recently, Li *et al.* [49] proposed a medium transmission-guided decoder network to enhance the response of our network towards quality-degraded regions. For the diversity of underwater images, Li *et al.* [43] also proposed to train an end-to-end UWCNN network, which directly reconstructs the clear latent image based on 10 types of underwater images. Hou *et al.* proposed an underwater residual network to jointly optimize transmission and correct color casts [50]. The authors of [47] proposed a Cycle-consistent Generative Adversarial Network (Cycle-GAN) for UIE. In [46], Li *et al.* developed a real-world underwater image database of Underwater Image Enhancement Benchmark (UIEB), and proposed a CNN model (called Water-Net) trained by this dataset for UIE. Although most deep learning-based methods have delivered excellent performances, they are at the expense of high complexity. To address this issue, some lightweight methods have been applied in other low-level tasks. For example, Fu *et al.* [17] introduced a Lightweight Pyramid Network (LPNet) for single image deraining. This network uses several sub-networks to predict the clean Gaussian pyramid, which simplifies the learning problem. LPNet requires fewer than 8K parameters while still achieving promising performance.

### III. PROPOSED METHOD

As discussed above, a practical UIE method should be capable of effectively processing underwater images with low complexity. In this work, it is achieved by multi-scale processing and recursive networks. Detailed illustration of our model is presented as follows.

#### A. Philosophy of Model Design

Underwater images have their particular features which impede the further improvement of UIE performance. Due to various light conditions and water current, the color details of images might be blended with object edges and backgrounds, which makes the pixel-domain mapping into a difficult task.

Moreover, the superior performance of a deep network usually requires an inevitable increase of model parameters, which make it costly in real-world application. To address these problems, we attempt to build a compact network by making full use of multi-scale features.

The introduction of multi-scale processing is also motivated by image quality assessment and other low-level tasks. In view of quality assessment, a user measures the quality of an image based on its multi-scale fidelities and textures [51], which inspires us to enhance images at different scales. Moreover, the multi-scale processing has also been widely employed in low-level processing for its ability to decompose a difficult problem to several easier sub-problems. In this work, we achieve the multi-scale processing with three steps: pyramid decomposition, multi-scale enhancement and image reconstruction.

In multi-scale enhancement, we introduce a compact network with lightweight sub-networks to enhance images in a coarse-to-fine manner. After pyramid decomposition, we are able to design simple sub-networks to handle each pyramid level in a divide-and-conquer way, resulting in compact sub-networks and fast computation. In addition, we also design each sub-network with recursive residual networks, that allow parameter reuse to further benefit the lightweighting of our model.

#### B. The Framework of LCNet

Based on the above discussions, we propose LCNet, that consists of pyramid decomposition, multi-scale enhancement and enhanced image reconstruction. LCNet is a pyramid network that incorporates a series of lightweight sub-networks to progressively refine the image details for output. The general architecture of LCNet is illustrated in Fig. 2. We first decompose an underwater image  $I$  into a multi-level Laplacian pyramid. For each pyramid level, we build a sub-network to enhance image residuals. All sub-networks share the same architecture, but they are trained separately (*i.e.*, no weight sharing). Finally, we enhance all Laplacian residuals at different levels and then reconstruct a Gaussian pyramid from coarse to fine to obtain the output enhanced image. Let

$I$  and  $I_E$  respectively denote an input underwater image and its enhanced version, the overall model can be expressed as

$$I_E = \text{LCN}(I), \quad (1)$$

where  $\text{LCN}(\cdot)$  denotes the operation of LCNet, which is elaborated as follows.

### C. Pyramid Decomposition

In this work, we employ Laplacian pyramid decomposition to decompose an image into multi-scale sub-bands. The Laplacian image pyramid is a mature algorithm with low computational cost [16]. Thanks to its multi-scale decomposition ability, the enhancement problem can be separately performed at different scales, thereby simplifying the learning process. In particular, the most time-consuming part of Laplacian decomposition can be realized by convolutions with GPU acceleration, thereby achieving significant speed-up. The multi-scale decomposition also facilitates the network to take advantage of sparsity at each level, thereby removing color artifacts [52].

As shown in Fig. 2, we adopt Laplacian pyramid decomposition to generate a set of images  $L_n$  with  $N$  levels:

$$L_n(I) = G_n(I) - u(G_{n+1}(I)), \quad (2)$$

where  $G_n(I)$  and  $u(\cdot)$  are the  $n$ -th Gaussian pyramid image and an upsampling operator, respectively.  $L_n(I)$  is the  $n$ -th Lapcian pyramid image, which is constructed by taking the difference between  $G_n(I)$  and  $u(G_{n+1}(I))$  ( $n = 1, 2, \dots, N-1$ ).  $L_N(I) = G_N(I)$  and  $G_1(I) = I$ .

### D. Multi-scale Enhancement

It is often desirable in practical applications to apply a lightweight network to trade model performance for efficiency. In this work, we realize it by an efficient model and parameter reuse across different pyramid levels.

With Laplacian pyramid decomposition, we are able to design simple sub-networks to handle each pyramid level in a divide-and-conquer way, resulting in compact sub-networks and fast computation. In addition, we also design each sub-network in a recursive processing manner, which allows parameter reuse to further benefit the lightweighting of our model. The illustrations of multi-level processing are as follows:

After decomposing  $I$  into different pyramid levels, we build a set of sub-networks for enhancement. As shown in Fig. 3, each sub-network consists of two  $3 \times 3$  convolutional layers and four blocks. At the front and end of sub-network, convolutional layers with Leaky Rectified Linear Units (LReLUs) [53] are added to adapt to the feature dimensions. All sub-networks share the same architecture but their parameters are separately trained. Thus, at each level, the sub-network only needs to consider the input's own physical characteristics (e.g., low or high frequencies). The basic formulation of sub-network at the  $n$ -th level can be expressed as

$$H_n = f_C(L_n(I)), \quad (3)$$

where  $f_C(\cdot)$  denotes the operation of sub-network,  $n = 1, 2, \dots, N$ .

1) *Cascaded Sub-Network*: The deep residual networks (ResNet) achieve fast and stable convergence that addresses the vanishing gradient problem in back propagation. By learning a residual mapping for a few stacked layers, deep neural networks can be easily trained with impressive performance. In contrast, Tong *et al.* [54] proposed to concatenate feature maps densely from shallower to deeper layers to alleviate the vanishing gradient problem and reduce the number of model parameters. It can be interpreted as there is no need to learn redundant features. Inspired by this, we perform a global residual learning with a shortcut in each block to facilitate the learning process.

As shown in Fig. 3, Block 1 takes an input  $r_0$ , which represents the features extracted from  $L_n(I)$ , and generates an output  $r_1$  to its succeeding blocks. The combination of  $r_1$  and  $L_n(I)$  forms  $y_1$  as another input to the next block. Block 2 receives  $y_1$  and  $r_1$  and outputs  $r_2$ , which serves as an input to all its succeeding blocks as well as  $y_2$ . Sequentially, Blocks 3 and 4 also receive features from their preceding blocks and output features. In particular, the output of Block 4, denoted as  $r_4$ , is utilized to generate  $y_4$  only, that is further processed by the convolutional layer to generate the enhanced residual  $H_n$ . The above process can be expressed as

$$\begin{aligned} r_0 &= c_{3 \times 3}(L_n(I)), \\ r_1 &= f_{B_1}(r_0), \\ y_1 &= r_1 + L_n(I), \\ r_2 &= f_{B_2}(y_1, r_1), \\ y_2 &= r_2 + L_n(I), \\ r_3 &= f_{B_3}(y_2, r_2, r_1), \\ y_3 &= r_3 + L_n(I), \\ r_4 &= f_{B_4}(y_3, r_3, r_2, r_1), \\ y_4 &= r_4 + L_n(I), \\ H_n &= c_{3 \times 3}(y_4), \end{aligned} \quad (4)$$

where  $c_{3 \times 3}(\cdot)$  denotes the operation of convolutional layer with  $3 \times 3$  kernel,  $f_{B_b}(\cdot)$  denotes the operation of each block and  $r_b$  denotes the output of the  $b$ -th block within a sub-network,  $b = 1, 2, 3, 4$ . The output of Fig. 3 is then cascaded to fuse all information of sub-network. Through this procedure, the LCNet quickly predicts residuals from the shallower to deeper layers.

2) *Recursive Strategy*: In each pyramid level, we adopt a lightweight cascaded sub-network consisting of several blocks, where each block adopts convolutional layers and LReLUs to formulate a nonlinear mapping of features. As a result, the cascaded networks achieve feature extraction with a high model complexity. To develop a lightweight model, we adopt the recursive scheme in each block, in which two recursive units are deployed with shared weights that can be trained simultaneously. Through weight sharing, the model parameters are significantly reduced. As shown in Fig. 4, each block has two convolutional layers and two recursive units. The

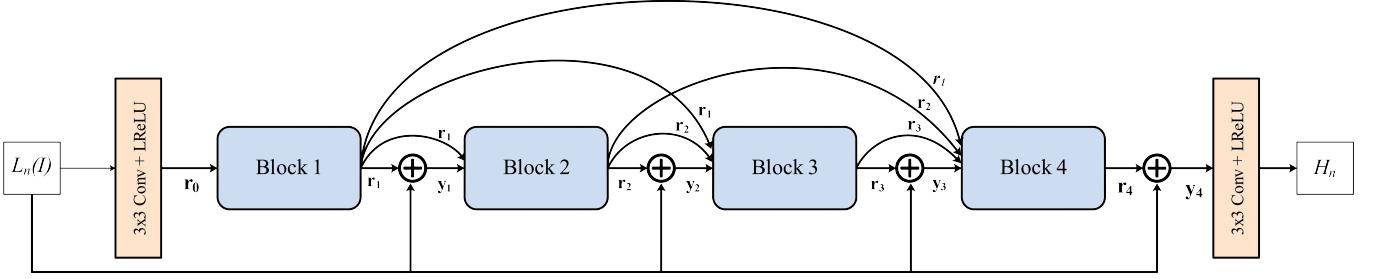


Fig. 3: Architecture of cascaded sub-network, which is composed of four recursive blocks

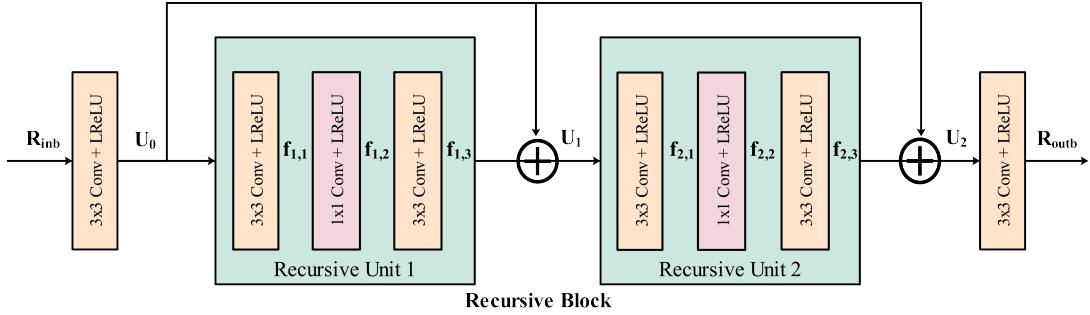


Fig. 4: Architecture of recursive block, which is composed of two recursive units.

processing within a block can be expressed as

$$\begin{aligned} U_0 &= c_{3 \times 3 \times b}(R_{inb}), \\ U_1 &= R(U_0) + U_0, \\ U_2 &= R(U_1) + U_0, \\ R_{outb} &= c_{3 \times 3}(U_2), \end{aligned} \quad (5)$$

where  $R_{inb}$ ,  $R_{outb}$  are the input and output of the  $b$ -th block ( $b = 1, 2, 3, 4$ ).  $R(\cdot)$  denotes the operation of the recursive unit,  $U_{t-1}$  and  $U_t$  are the input and output of the  $t$ -th unit ( $t = 1, 2$ ). In particular, the  $c_{3 \times 3 \times b}$  is used for adapting to the different dimensions of  $R_{inb}$ . The input-output relations within blocks can be expressed as

$$\begin{aligned} R_{in1} &= r_0, \\ R_{out1} &= r_1, \\ R_{in2} &= (r_1, y_1), \\ R_{out2} &= r_2, \\ R_{in3} &= (r_1, r_2, y_2), \\ R_{out3} &= r_3, \\ R_{in4} &= (r_1, r_2, r_3, y_3), \\ R_{out4} &= r_4. \end{aligned} \quad (6)$$

Each unit can be formulated as

$$\begin{aligned} f_{t,1} &= c_{3 \times 3}(W_{t,1}; U_{t-1}), \\ f_{t,2} &= c_{1 \times 1}(W_{t,2}; f_{t,1}), \\ f_{t,3} &= c_{3 \times 3}(W_{t,3}; f_{t,2}), \end{aligned} \quad (7)$$

where  $f_{t,1}$ ,  $f_{t,2}$  and  $f_{t,3}$  are intermediate features in corresponding  $t$ -th recursive unit, respectively.  $W_{t,1}$ ,  $W_{t,2}$  and  $W_{t,3}$

are weights which are shared across the two recursive units. The architecture of recursive unit 2 is the same as that of unit 1.

#### E. Enhanced Image Reconstruction

In the last stage, we use element-wise summation to combine Laplacian pyramid images  $L_n$  with enhanced residual  $H_n$ :

$$L_n^e(I) = H_n + L_n(I), \quad (8)$$

where  $n = 1, 2, \dots, N$ ,  $L_n^e(I)$  is the output of the Laplacian image at the  $n$ -th level.

Consequently, the corresponding Gaussian pyramid of the enhanced image can be reconstructed as

$$G_n^e(I) = L_n^e(I) + u(G_{n+1}^e(I)), \quad (9)$$

where  $n = 1, 2, \dots, N-1$ ,  $L_N^e(I) = G_N^e(I)$ . The final enhanced image  $I(E)$  is obtained at the bottom level of the Gaussian pyramid  $G_1^e(I)$ .

#### F. Loss Function

To achieve both good subjective visual quality and high fidelity, we combine Mean Square Error (MSE) loss and Perceptual (PER) loss to guide the training of our network. We initialize training image pairs  $\{I^i, I_{gt}^i\}_{i=1}^K$ , where  $I_{gt}^i$  is the ground-truth, and  $K$  denotes the number of the training images.

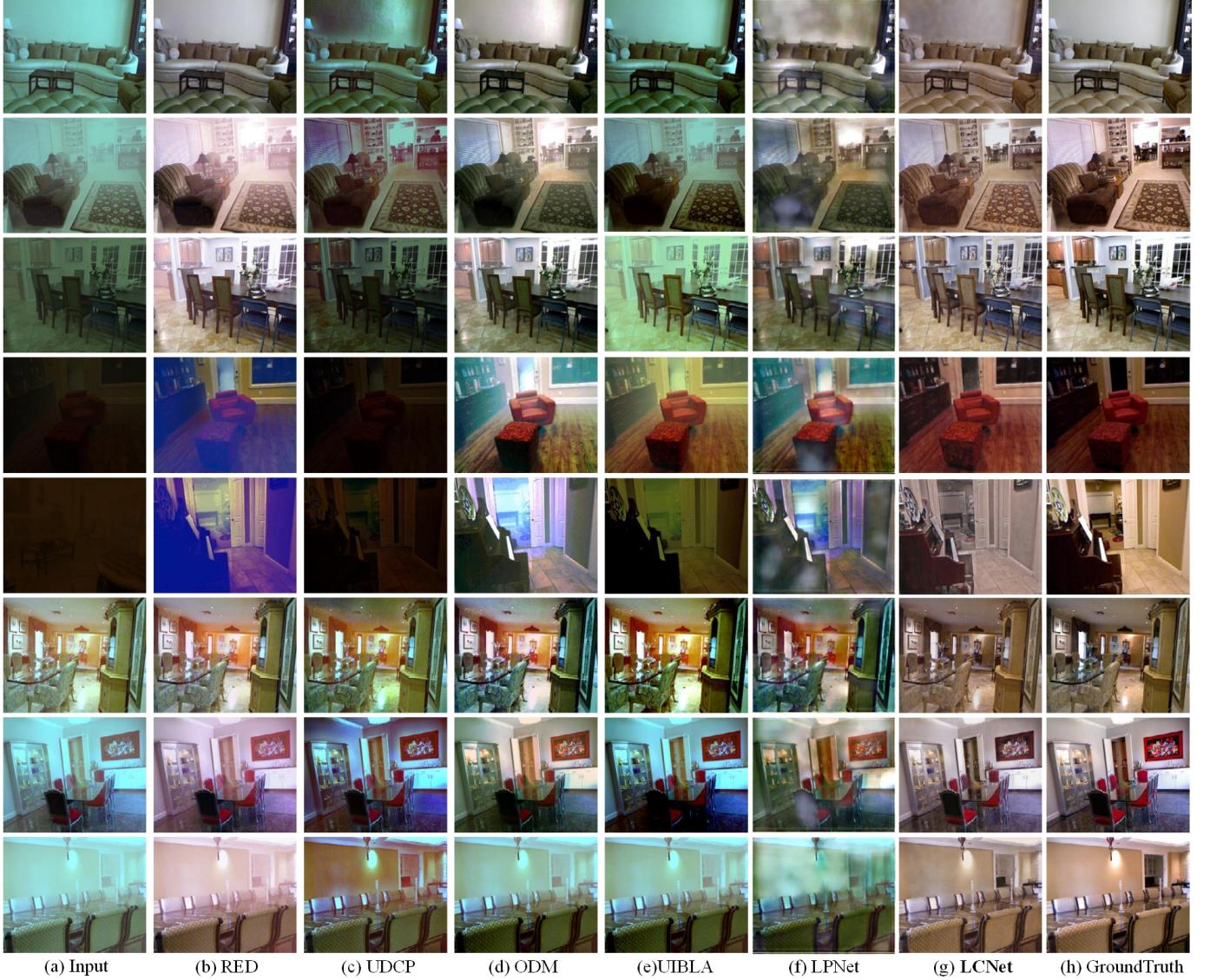


Fig. 5: Full-reference visual comparisons of six UIE methods performed on eight synthetic images.

The MSE metric is a widely used loss function in many low-level tasks (*e.g.*, super-resolution, image deraining). It can be expressed as

$$L_{mse} = \frac{1}{K} \sum_{i=1}^K \|LCN(I^i) - I_{gt}^i\|_2^2. \quad (10)$$

However, training with the MSE loss only might lead to over-smoothed results. To avoid this problem, we combine it with a perceptual loss [55], which measures the feature difference between our predicted output and ground-truth. Similar to [46], [55], we input the two images into a pre-trained VGG16 [56] network and calculate their feature difference:

$$L_{per} = \frac{1}{K} \sum_{i=1}^K \frac{1}{C_j H_j W_j} \|V_j(LCN(I^i)) - V_j(I_{gt}^i)\|_2^2, \quad (11)$$

where  $C_j H_j W_j$  represents the dimension of the feature map at  $j$ -th convolution layer within the VGG16 network, and  $V_j(X)$  represents the feature of the  $j$ -th convolutional layer.

In particular, we focus on the last convolutional layer of VGG16 only. It is noted that the VGG16 model accepts multi-channel inputs and thus the perceptual loss can be utilized for color image processing. Finally, the above two functions are summed up to obtain the overall loss:

$$L_{loss} = L_{mse} + \lambda * L_{per}, \quad (12)$$

where  $\lambda = 0.02$  is set empirically, according to [7]. The effects of  $\lambda$  will be demonstrated in Table IX of Section IV-E.

#### G. Parameter Settings

The parameters involved in our model include the model and training parameters. In the model parameters, we set the number of pyramid levels  $N = 4$ , the number of recursive blocks  $b = 4$  and the weight for loss function  $\lambda = 0.02$ . Ablation studies on these parameters are provided in Section IV-E. As for activation function, we use LReLUs with a small negative slope of 0.1. For training, We randomly generate one million

$80 \times 80$  patch pairs. We use the Adam optimizer [57] with default settings: alpha = 0.001, beta1 = 0.9, beta2 = 0.999 and epsilon =  $10^{-8}$ . The initial learning rate and epochs are set to 0.001 and 3, respectively. The whole model is trained in an end-to-end manner.

#### IV. EXPERIMENTAL RESULTS

In this section, we evaluate the proposed method and compare it with several state-of-the-art UIE methods on both synthetic and real-world underwater images.

##### A. Compared Algorithms

We choose 15 popular algorithms for comparison, including Fusion-Based [20], RED [58], UDCP [29], ODM [31], UIBLA [59], LPNet [17], UWCNN [43], Retinex-Based [22], GDPC [30], Ancuti *et al.* (2016) [25], Ancuti *et al.* (2017) [24], Berman *et al.* (2020) [34], Water-CycleGAN [47], Dense-GAN [44] and Water-Net [46]. Among them, [20], [24], [25] are non-physics model-based methods, [22], [25], [29]–[31], [34], [59] are physics model-based methods and [17], [43], [44], [46], [47] are deep learning-based methods. All experiments (Training/Testing) are performed on a server equipped with two NVIDIA GeForce GTX 2080Ti with 16GB GPU memory. The size of images are set to  $512 \times 512$ .

##### B. Datasets

**Synthetic Underwater Image Dataset.** We train the proposed network on synthetic images from [43] which were generated using the attenuation coefficients described in [60]. This dataset contains 10 different water types of oceanic and coastal classes (*i.e.*, I, IA, IB, II, and III for open ocean waters, and 1, 3, 5, 7, and 9 for coastal waters). Each type consists of 1449 images, in which we randomly select 1000 images as training set and the remaining 449 images as testing set.

**Real-World Underwater Image Dataset.** We also use the UIEB real-world dataset built in [46] to evaluate different methods. These images are collected from Google and YouTube, and their corresponding reference images are generated by 15 image enhancement methods. The final versions of images are provided according to well-designed pairwise comparisons. This dataset consists of 890 underwater images with their corresponding high-quality reference images, in which a random set of 800 pairs of the images are used for training and the rest 90 images are used for testing. Besides, it also contains 60 challenging underwater images for non-reference quality evaluation. Recently, a new underwater dataset, named Stereo Quantitative Underwater Image Dataset (SQUID), was proposed in [34]. The dataset contains 57 stereo pairs from four different sites in Israel and takes into account multiple spectral profiles of different water types. In addition, this dataset is only used for non-reference evaluation.

##### C. Results on Quantitative Evaluation

To compare the performances of all methods, we perform both quantitative and qualitative evaluations on each dataset.

First of all, we compare all methods on the synthetic dataset and the UIEB dataset (except its challenging set), where the reference images are available for full reference quality evaluation. Four full-reference quality metrics are utilized: Peak Signal to Noise Ratio (PSNR), Structural SIMilarity (SSIM), Mean Squared Error (MSE) and Learned Perceptual Image Patch Similarity (LPIPS) [61]. Among them, the PSNR and SSIM evaluate the image similarity, where a higher score indicates a better performance, and the other two metrics evaluate the discrepancy, which favors lower scores. All comparison results are summarized in Tables I, II, III. Due to the space limit, we provide the average scores of all methods only. The values colored in red and blue indicate the best and second-best results, respectively. As can be observed, the proposed LCNet outperforms the compared methods in most cases. Even in some cases for which our models are not the best or second-best, its performances are still close to the best results in most of quality metrics.

We also compare all methods in the challenging set of UIEB and the SQUID dataset, where the reference images are unavailable. In such case, we utilize the non-reference quality metrics including Underwater Color Image Quality Evaluation (UCIQE) [11], Underwater Image Quality Measure (UIQM) [12], Cross Correlation Function (CCF) [62] and Integrated Local Natural Image Quality Evaluator (ILNIQE) [63]. All of them give higher scores for higher quality images. The comparison results in UIEB challenging set and SQUID are shown in Tables IV and V, respectively. They show that our model achieves the best or second-best scores in three of the four quality metrics. To the best of our knowledge, there is no metric with overwhelming superiority in the state-of-the-art UIE quality assessment. Therefore, we can still conclude that our LCNet outperforms the other models for its superior performances in most quality metrics.

In addition, we conduct the runtime analysis to examine the time efficiency of our lightweight model. As shown in Table VI, LCNet achieves the fastest computation. Compared with the fastest state-of-the-art, our model achieves a significant run-time reduction of 55.3% on average. This fact indicates that our model achieves both quality improvement and complexity reduction. Moreover, Table VI also shows that, compared with deep learning-based state-of-the-arts, our model significantly reduces the number of model parameters and the number of FLOPs (Floating Point Operations), which also demonstrates the efficiency of our model.

##### D. Results on Qualitative Evaluation

Fig. 5 shows the results of some synthetic samples. The compared algorithm might generate images with unnatural colors or low contrasts. For example, UDCP [29] produces distinctly darkish results, ODM [31] causes over-enhancement, UIBLA [59] introduces color deviations and LPNet [17] tends to generate obvious blurring artifacts. In contrast, our method generates visually pleasant results across all degradation types, showing more consistent color, with fewer effects of vignetting and attenuation compared to the other methods.

The visual comparisons of various methods on the UIEB are also illustrated Figs. 6, 7. To better represent the diversity

TABLE I: Quantitative results evaluated on synthetic underwater images (PSNR  $\uparrow$  and SSIM  $\uparrow$ ).

Water Type	1		3		5		7		9		I		II		III	
	PSNR	SSIM														
RAW	15.53	0.765	14.69	0.629	12.14	0.502	10.17	0.519	9.50	0.302	17.36	0.886	20.59	0.868	16.55	0.788
Fusion-Based [20]	15.79	0.706	15.01	0.578	13.19	0.421	11.02	0.279	10.76	0.179	18.38	0.862	20.88	0.871	17.12	0.752
RED [58]	15.90	0.741	12.79	0.663	11.12	0.593	9.99	0.508	11.62	0.319	19.54	0.881	20.79	0.883	16.69	0.791
UDCP [29]	15.76	0.763	14.47	0.661	10.86	0.426	9.47	0.262	9.32	0.162	18.82	0.826	17.20	0.838	14.92	0.758
ODM [31]	16.09	0.724	14.28	0.676	14.12	0.644	12.27	0.563	9.30	0.417	18.09	0.817	17.61	0.825	16.71	0.754
GDCP [30]	15.89	0.779	17.45	0.706	12.89	0.687	12.47	0.613	13.08	0.462	21.42	0.826	23.20	0.832	18.92	0.828
Retinex-Based [22]	16.02	0.735	15.81	0.688	13.53	0.625	12.02	0.679	12.16	0.455	20.58	0.862	22.19	0.892	17.75	0.791
UIBLA [59]	15.88	0.695	13.44	0.576	12.61	0.474	10.75	0.305	10.09	0.220	17.49	0.744	18.06	0.801	17.10	0.765
LPNet [17]	18.38	0.755	13.71	0.663	13.01	0.558	10.37	0.466	10.33	0.373	17.87	0.812	18.35	0.857	17.49	0.807
Ancuti <i>et al.</i> (2016) [25]	16.22	0.651	14.12	0.593	12.98	0.532	11.31	0.499	11.45	0.391	17.12	0.801	18.12	0.831	17.11	0.801
Ancuti <i>et al.</i> (2017) [24]	16.41	0.677	14.33	0.599	13.31	0.542	11.66	0.512	11.55	0.401	17.35	0.803	18.22	0.840	17.21	0.808
Berman <i>et al.</i> (2020) [34]	17.01	0.702	14.82	0.613	13.62	0.581	12.01	0.532	11.86	0.489	17.66	0.811	18.63	0.861	17.65	0.812
Water-CycleGAN [47]	17.19	0.789	19.18	0.776	17.32	0.758	14.32	0.683	12.15	0.517	22.89	0.817	24.61	0.837	19.66	0.889
Dense-GAN [44]	19.88	0.815	14.15	0.793	17.51	0.819	14.13	0.712	13.09	0.620	23.99	0.882	23.06	0.821	20.15	0.862
Water-Net [46]	19.15	0.832	14.39	0.805	17.11	0.822	14.16	0.785	13.21	0.613	24.17	0.898	24.35	0.862	20.29	0.833
UWCNN [43]	21.79	0.850	20.25	0.795	17.52	0.726	14.22	0.607	13.23	0.492	25.93	0.937	24.82	0.923	22.63	0.879
LCNet	22.14	0.876	23.38	0.832	22.35	0.816	21.55	0.797	21.46	0.651	26.34	0.895	25.28	0.907	23.18	0.916

TABLE II: Quantitative results evaluated on synthetic underwater images (MSE  $\downarrow$  and LPIPS  $\downarrow$ ).

Water Type	1		3		5		7		9		I		II		III	
	MSE	LPIPS	MSE	LPIPS	MSE	LPIPS	MSE	LPIPS	MSE	LPIPS	MSE	LPIPS	MSE	LPIPS	MSE	LPIPS
RAW	2367.3	0.246	2676.5	0.453	4851.2	0.393	7381.1	0.502	9060.6	0.713	1449.0	0.242	941.9	0.253	1851.0	0.421
Fusion-Based [20]	2258.3	0.227	2836.5	0.442	3218.5	0.312	5835.8	0.425	7895.4	0.632	1239.2	0.208	747.2	0.192	1293.4	0.386
RED [58]	3489.7	0.126	4953.2	0.281	8385.8	0.233	9809.8	0.423	5952.3	0.427	936.9	0.241	851.3	0.177	2240.0	0.241
UDCP [29]	2062.3	0.304	3380.6	0.277	6708.9	0.319	8591.6	0.414	9500.1	0.458	1020.7	0.234	1466.9	0.229	2337.6	0.245
ODM [31]	2508.6	0.117	3130.1	0.268	3488.9	0.142	5337.1	0.410	10634.0	0.230	1272.0	0.239	1401.9	0.171	1701.1	0.326
GDCP [30]	1877.4	0.152	2542.2	0.252	2799.5	0.289	3578.9	0.394	3528.1	0.458	738.9	0.202	811.2	0.194	1788.4	0.227
Retinex-Based [22]	2149.2	0.193	1677.9	0.423	3009.2	0.232	4766.2	0.395	4312.7	0.659	1128.2	0.218	658.3	0.181	1031.2	0.312
UIBLA [59]	2812.6	0.207	3490.1	0.368	4563.7	0.330	6737.9	0.299	8433.1	0.265	1492.2	0.237	1141.4	0.225	1697.8	0.319
LPNet [17]	2502.2	0.254	3217.8	0.373	4105.2	0.219	5832.7	0.408	8722.7	0.252	1099.6	0.278	1022.5	0.264	1487.2	0.397
Ancuti <i>et al.</i> (2016) [25]	1365.8	0.354	2917.2	0.311	4615.2	0.232	3812.5	0.412	7739.7	0.253	1126.3	0.277	1121.5	0.292	1827.1	0.401
Ancuti <i>et al.</i> (2017) [24]	1347.2	0.313	2527.2	0.302	4533.2	0.212	3202.3	0.396	7255.9	0.241	1109.2	0.262	1002.4	0.263	1617.1	0.381
Berman <i>et al.</i> (2020) [34]	1042.3	0.301	2217.2	0.282	3833.6	0.201	2811.7	0.312	6651.4	0.223	899.2	0.249	898.1	0.254	1482.1	0.361
Water-CycleGAN [47]	655.3	0.168	698.2	0.168	867.5	0.141	1423.7	0.380	1247.9	0.370	876.2	0.142	1232.9	0.162	338.9	0.233
Dense-GAN [44]	609.2	0.113	536.7	0.149	409.5	0.165	1028.3	0.199	865.3	0.365	213.6	0.138	203.8	0.125	411.2	0.222
Water-Net [46]	586.2	0.112	402.3	0.176	616.5	0.171	785.6	0.130	799.3	0.434	455.9	0.112	223.4	0.155	452.9	0.134
UWCNN [43]	587.7	0.200	747.5	0.192	1295.1	0.114	2974.1	0.123	4121.5	0.218	209.7	0.123	251.6	0.176	456.4	0.114
LCNet	467.7	0.107	398.2	0.146	427.9	0.135	613.7	0.128	598.3	0.215	137.5	0.097	205.7	0.102	376.9	0.397

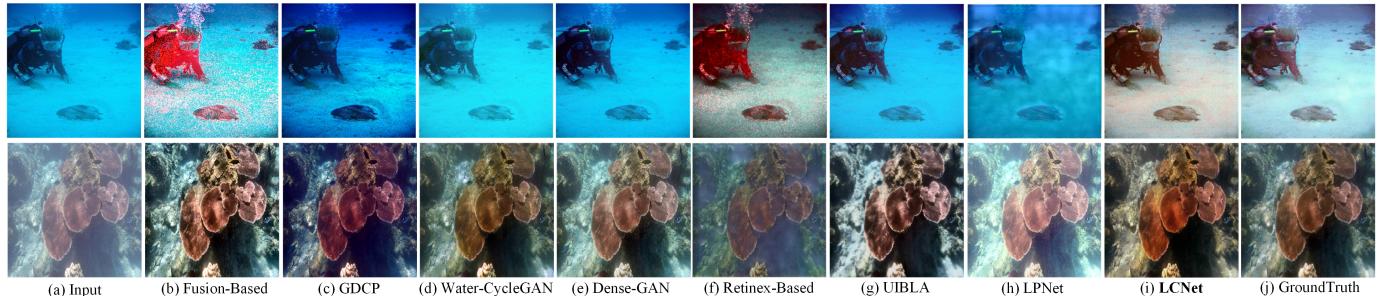


Fig. 6: Visual qualitative comparisons on real-world images.

of underwater scenarios, we intentionally select images with different types. From the results, Fusion-Based [20] and GDCP [30] introduce reddish or greenish colors, LPNet [17] still tends to leave some blurry parts of UIE images, UIBLA [59] overly sharpens the images, and Dense-GAN [44] causes over-enhancement and over-saturation. Water-CycleGAN [47] produces color casts because it tends to translate content and structure of images to the underwater images. As for evaluations on SQUID [34], Fig. 8 shows that Ancuti *et al.* (2017) [24] produces distinctly darkish results, while UIBLA [59] and Berman *et al.* [34] introduce artificial colors or color deviations. In contrast, our method leads to more consistent

colors across various views. It can also reduce vignetting and attenuation effects while better preserving the structures of individual objects in underwater images.

The proposed LCNet also benefits downstream object recognition under underwater environment, which can be demonstrated by a comparison study. First, we generally collect 150 UIEB images with detected objects: 65 with humans (*e.g.* divers) and 85 with fishes, and then enhance them with our LCNet. Second, we pretrain a YOLOv4 [64] model with UIEB database [46] for underwater object recognition. Third, we compare the performances of object detection with the original and enhanced images. Fig. 9 shows the different

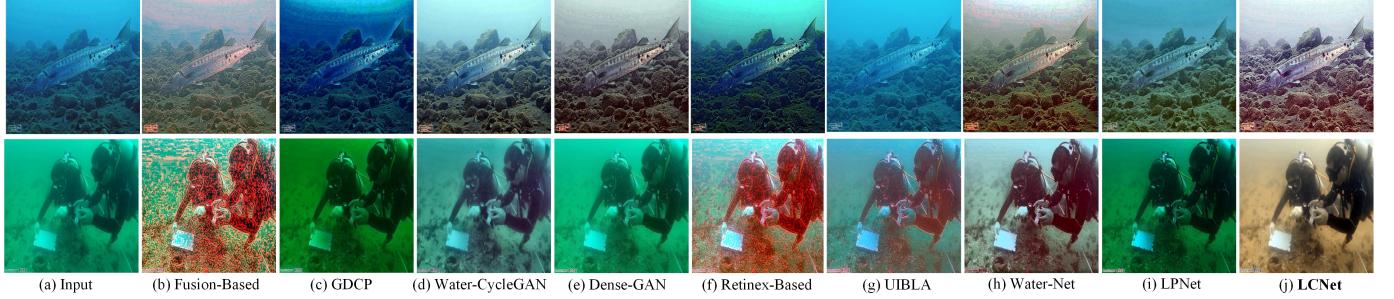


Fig. 7: Visual qualitative comparisons on challenging set.

TABLE III: Quantitative results evaluated on real-world dataset UIEB.

Methods	PSNR↑	SSIM↑	MSE↓	LPIPS↓
Fusion-Based [20]	17.61	0.772	1128.0	0.344
RED [58]	17.89	0.793	933.2	0.298
UDCP [29]	18.79	0.811	897.2	0.295
ODM [31]	15.82	0.539	1701.9	0.382
UIBLA [59]	15.32	0.603	1911.1	0.311
LPNet [17]	15.65	0.714	1671.3	0.396
UWCNN [43]	19.08	0.812	766.1	0.344
Retinex-Based [22]	17.02	0.607	1292.4	0.537
GDCP [30]	12.09	0.512	1010.0	0.770
Ancuti <i>et al.</i> (2016) [25]	15.12	0.602	1324.2	0.612
Ancuti <i>et al.</i> (2017) [24]	15.58	0.724	1466.2	0.649
Berman <i>et al.</i> (2020) [34]	16.11	0.711	1320.9	0.596
Water-CycleGAN [47]	15.75	0.521	1729.8	0.348
Dense-GAN [44]	17.28	0.443	1215.2	0.318
Water-Net [46]	19.11	0.797	797.6	0.222
<b>LCNet</b>	<b>23.22</b>	<b>0.923</b>	<b>573.6</b>	<b>0.287</b>

TABLE IV: Non-reference image quality evaluations on challenge set of UIEB [46].

Methods	UCIQE↑	UIQM↑	CCF↑	ILNIQE↑
Fusion-Based [20]	0.6414	1.5310	15.0811	0.4695
RED [58]	0.6355	1.489	14.1922	0.5733
UDCP [29]	0.6216	1.435	14.2943	0.5812
ODM [31]	0.6778	1.5440	6.0821	0.5768
UIBLA [59]	0.6001	1.3757	15.7701	0.4619
LPNet [17]	0.6221	1.2782	9.1002	0.4724
UWCNN [43]	0.6328	1.5082	12.3402	0.6579
Retinex-Based [22]	0.6062	1.4338	13.4510	0.4911
GDCP [30]	0.5993	1.4301	12.0734	0.4979
Ancuti <i>et al.</i> (2016) [25]	0.6001	1.3908	13.2439	0.5100
Ancuti <i>et al.</i> (2017) [24]	0.5872	1.4871	12.9912	0.5495
Berman <i>et al.</i> (2020) [34]	0.6231	1.4118	13.1212	0.5766
Water-CycleGAN [47]	0.5656	1.4376	12.1455	0.6386
Dense-GAN [44]	0.6419	1.5467	13.4732	0.6051
Water-Net [46]	0.6110	1.4640	10.3101	0.6566
<b>LCNet</b>	<b>0.6449</b>	<b>1.4096</b>	<b>16.2102</b>	<b>0.6919</b>

recognition confidence distributions of the two groups of images. It can be easily concluded that our enhancement leads to an improvement of object recognition. Such improvement can also be demonstrated by the results in Fig. 10, where all recognition confidences are improved. In particular, an extra diver is successfully detected after LCNet-based UIE.

### E. Ablation Study

To further examine the effectiveness of our framework, we investigate the different configurations and study their impacts on performance.

TABLE V: Non-reference image quality evaluations on SQUID [34].

Methods	UCIQE↑	UIQM↑	CCF↑	ILNIQE↑
Fusion-Based [20]	0.6237	1.6110	15.2312	0.5012
RED [58]	0.6122	1.308	13.2317	0.4723
UDCP [29]	0.5916	1.277	10.3243	0.3812
ODM [31]	0.5782	1.3480	7.2831	0.5668
UIBLA [59]	0.4988	1.4297	15.6901	0.3939
LPNet [17]	0.6119	1.2213	9.1002	0.4724
UWCNN [43]	0.6738	1.5342	11.5422	0.6012
Retinex-Based [22]	0.6179	1.4125	12.1599	0.4232
GDCP [30]	0.5803	1.2931	13.1924	0.4979
Ancuti <i>et al.</i> (2016) [25]	0.6003	1.3915	13.2509	0.5103
Ancuti <i>et al.</i> (2017) [24]	0.5885	1.4871	12.9912	0.5512
Berman <i>et al.</i> (2020) [34]	0.6252	1.4115	13.1223	0.5779
Water-CycleGAN [47]	0.5752	1.4188	12.1915	0.6192
Dense-GAN [44]	0.6529	1.5437	13.4285	0.6122
Water-Net [46]	0.6019	1.5123	11.4123	0.6896
<b>LCNet</b>	<b>0.6949</b>	<b>1.4286</b>	<b>15.7102</b>	<b>0.6989</b>

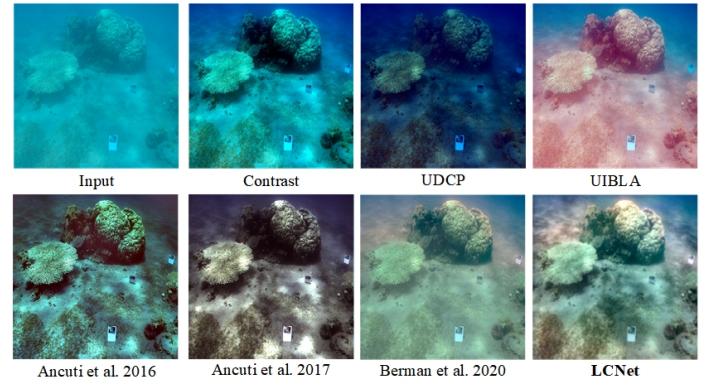


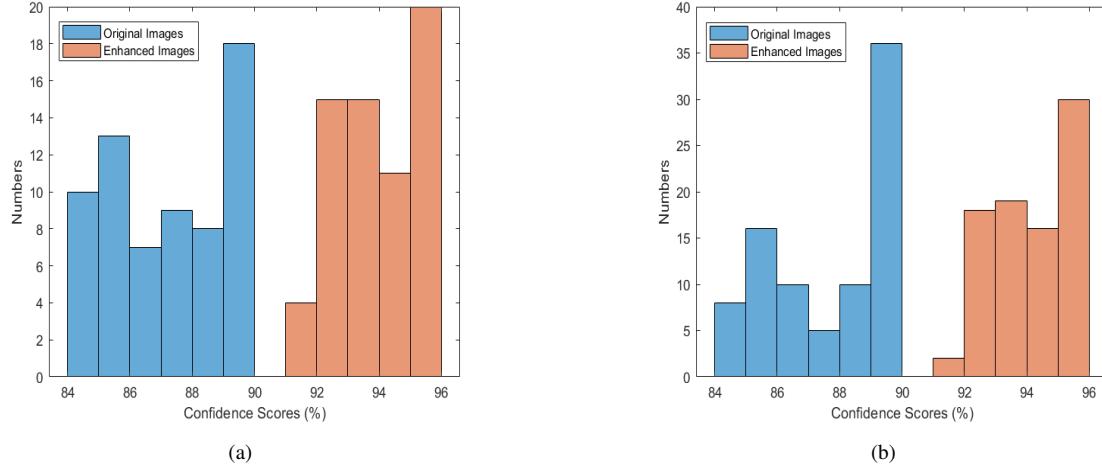
Fig. 8: Visual qualitative results on SQUID.

1) *Number of Pyramid Levels*: In our model, we utilize  $N$  to denote the number of pyramid levels and cascaded sub-networks. Generally speaking, a larger number will benefit model performance while leading to an inevitable increase of parameters. According to [16], a typical number of  $N$  should range from 3 to 5. In Table VII, we present the performances of our model with different  $N$  values. It can be seen that  $N = 4$  and  $N = 5$  achieve similar performance in terms of PSNR and SSIM, while  $N = 4$  leads to greatly reduced parameters. Therefore, we select  $N = 4$  in our work.

2) *Number of Recursive Blocks*: In each pyramid level, we design a set of cascaded sub-networks. The UIEB test results in Fig. 11 show performance improvement with an

TABLE VI: Average runtime of some selected methods. Red and blue indicate the best and second-best results.

Methods	Fusion-Based [20]	Retinex-Based [22]	RED [58]	UDCP [29]	ODM [31]	Water-CycleGAN [47]	Water-Net [46]	UWCNN [43]	<b>LCNet</b>
Time(s) 500×500	0.604	0.697	2.752	2.268	4.628	0.225	<b>0.213</b>	0.241	<b>0.121</b>
Time(s) 640×480	0.679	0.882	3.250	3.319	5.829	0.263	<b>0.228</b>	0.271	<b>0.128</b>
Time(s) 1280× 720	1.843	2.109	9.745	9.902	16.923	0.325	<b>0.282</b>	0.402	<b>0.157</b>
Parameters	-	-	-	-	-	57,169,668	1,090,668	<b>90,646</b>	<b>23,325</b>
FLOPs(M)	-	-	-	-	-	3,887,403	<b>1,937,278</b>	3,931,010	<b>139,055</b>

Fig. 9: The distributions of confidence scores of original and enhanced images. (a) Humans ( $87.4 \pm 4.5\%$  vs  $93.8 \pm 1.8\%$ ). (b) Fishes ( $86.7 \pm 4.2\%$  vs  $93.5 \pm 2.1\%$ ).

increased number of blocks. It also demonstrates the superiority of LReLUs as the activate function in Fig. 11-(a). If the application scenario changes, one can detach a portion of our model to meet varying computational requirements at test. For instance, we can simplify our model by using two cascade blocks (namely LCNet-2), which are still able to achieve an average PSNR of 22.85dB on UIEB. This performance still outperforms some popular algorithms. Meanwhile, as shown in Table VIII, its number of parameters is greatly reduced by 54.6% (from 23,325 to 10,593). This demonstrates the flexibility of our model to offer different trade-offs between quality and complexity different scenarios.

TABLE VII: PSNR and SSIM comparisons on UIEB for different numbers of pyramid levels.

Pyramid levels	$N = 3$	$N = 4$	$N = 5$
PSNR	22.69	<b>23.22</b>	23.43
SSIM	0.897	<b>0.923</b>	0.926
Parameters	10,140	<b>23,325</b>	60,390

TABLE VIII: Comparison of complexity in terms of model parameters.

Methods	Water-Net [46]	UWCNN [43]	LCNet-2	LCNet-3	LCNet-4(ours)
Parameters	1,090,668	90,646	<b>10,593</b>	16,392	23,325

3) *Loss Function*: To validate the effectiveness of our loss function, we train the LCNet model with MSE and perceptual loss only and compare their performances with our complete model. Fig. 11-(b) indicates that each component has its own

performance contribution, which justifies the overall design. Moreover, we also conduct experiments to evaluate the impact of  $\lambda$ . Although  $\lambda = 0.02$  is a popular setting, some works also use  $\lambda$  values of 0.01 and 0.04 [7], [65]. In Table IX, the experiment demonstrates the superior performance of  $\lambda = 0.02$  in our model.

TABLE IX: PSNR and SSIM comparisons on UIEB for different values of  $\lambda$ .

Values	$\lambda = 0.01$	$\lambda = 0.02$	$\lambda = 0.04$
PSNR	23.02	<b>23.22</b>	23.15
SSIM	0.901	<b>0.923</b>	0.912

TABLE X: PSNR and SSIM comparisons on UIEB for different numbers of feature maps.

Number of feature maps	16	12	8	4
PSNR	<b>23.22</b>	23.14	22.81	21.99
SSIM	<b>0.923</b>	0.923	0.895	0.854

4) *Number of Learned Feature Maps*: We also conduct an experiment on UIEB with an increased number of feature maps (out of total 16 feature maps for all convolutional layers of each sub-network). Fig. 12 visualizes four learned feature maps generated by a sub-network, where each learned feature map enhances a certain aspect of a given image. For instance, the learned representation with 16 feature maps highlights specific textures, which is not evidently shown in the input image. Besides, the quantitative results in Table X show the performance gain achieved by the diversity of learned features. These experiments demonstrate that effectiveness of feature

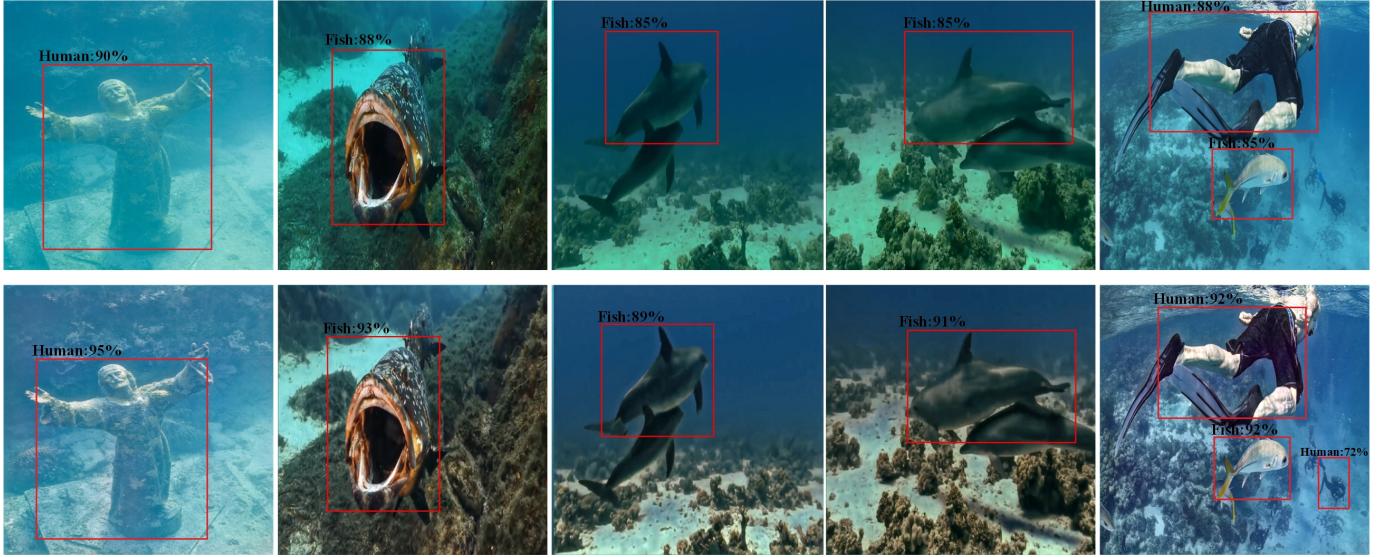


Fig. 10: The improvement of object recognition before (the top row) and after (the bottom row) LCNet-based UIE.

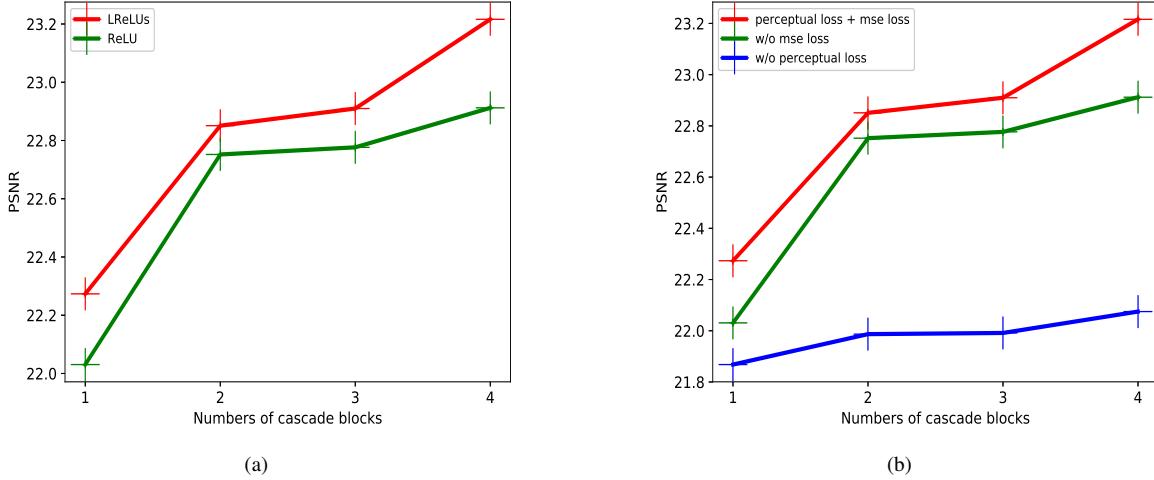


Fig. 11: Comparisons of our method using different (a) activation functions and (b) loss functions.

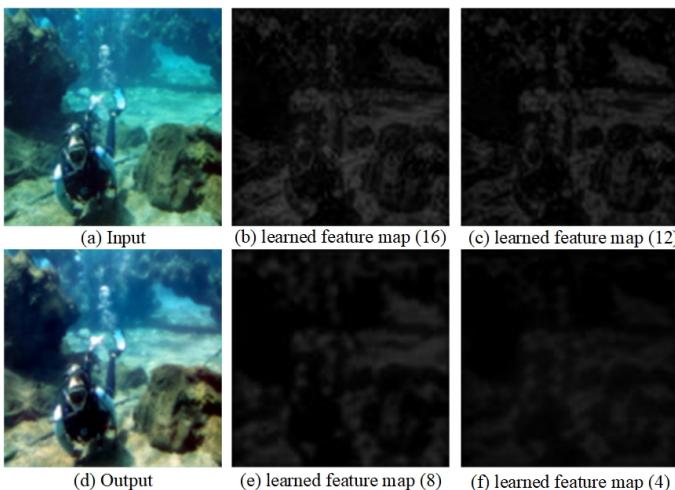


Fig. 12: Comparisons of different numbers of feature maps.

maps that more feature maps improve the performance of UIE modeling.

## V. CONCLUSIONS

In this paper, we proposed a lightweight deep network for UIE. The proposed LCNet consists of three steps: Laplacian pyramid-based image decomposition, sub-network-based multi-scale enhancement and reconstruction of enhanced images. Specifically, we employ cascaded sub-networks to predict sub-band residuals in a coarse-to-fine fashion to extract a diversity of features. In addition, to develop a lightweight model, we introduce the recursive strategy for model and parameter reuse. Experimental results demonstrate the effectiveness, robustness and low complexity of our UIE model.

## REFERENCES

- [1] D. Lee, G. Kim, D. Kim, H. Myung, and H.-T. Choi, “Vision-based object detection and tracking for autonomous navigation of underwater robots,” *Ocean Eng.*, vol. 48, pp. 59–68, 2012.

- [2] Y.-C. Liu, S.-Y. Liu, and N. Wang, "Fully-tuned fuzzy neural network based robust adaptive tracking control of unmanned underwater vehicle with thruster dynamics," *Neurocomputing*, vol. 196, no. 6, pp. 1–13, 2016.
- [3] Y. Li, H. Lu, J. Li, X. Li, Y. Li, and S. Serikawa, "Underwater image de-scattering and classification by deep neural network," *Comput. Electr. Eng.*, vol. 54, pp. 68–77, 2016.
- [4] H. Zhang and V. M. Patel, "Density-aware single image de-raining using a multi-stream dense network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 695–704.
- [5] H. Zhang, V. Sindagi, and V. M. Patel, "Image de-raining using a conditional generative adversarial network," *IEEE Trans. Image Process.*, vol. 30, no. 11, pp. 3943–3956, 2019.
- [6] Y. Wang, D. Gong, J. Yang, Q. Shi, A. van den Hengel, D. Xie, and B. Zeng, "Deep single image deraining via modeling haze-like effect (early access)," *IEEE Trans. Multimedia*, 2020.
- [7] R. Li, J. Pan, Z. Li, and J. Tang, "Single image dehazing via conditional generative adversarial network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 8202–8211.
- [8] X. Liu, Y. Ma, Z. Shi, and J. Chen, "Griddehazenet: Attention-based multi-scale network for image dehazing," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 7314–7323.
- [9] X. Yang, H. Li, Y.-L. Fan, and R. Chen, "Single image haze removal via region detection network," *IEEE Trans. Multimedia*, vol. 21, no. 10, pp. 2545–2560, 2019.
- [10] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising," *IEEE Trans. Image Process.*, vol. 26, no. 7, pp. 3142–3155, 2017.
- [11] M. Yang and A. Sowmya, "An underwater color image quality evaluation metric," *IEEE Trans. Image Process.*, vol. 24, no. 12, pp. 6062–6071, 2015.
- [12] K. Panetta, C. Gao, and S. Agaian, "Human-visual-system-inspired underwater image quality measures," *IEEE J. Ocean. Eng.*, vol. 41, no. 3, pp. 541–551, 2015.
- [13] N. Carlevaris-Bianco, A. Mohan, and R. M. Eustice, "Initial results in underwater single image dehazing," in *MTS/IEEE Seattle, OCEANS*, 2010, pp. 1–8.
- [14] J. Y. Chiang and Y.-C. Chen, "Underwater image enhancement by wavelength compensation and dehazing," *IEEE Trans. Image Process.*, vol. 21, no. 4, pp. 1756–1769, 2011.
- [15] H. Wen, Y. Tian, T. Huang, and G. Wen, "Single underwater image enhancement with a new optical model," in *Proc. IEEE Int. Symp. Circuits Syst.*, 2013, pp. 753–756.
- [16] W.-S. Lai, J.-B. Huang, N. Ahuja, and M.-H. Yang, "Fast and accurate image super-resolution with deep laplacian pyramid networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 11, pp. 2599–2613, 2018.
- [17] X. Fu, B. Liang, Y. Huang, X. Ding, and J. Paisley, "Lightweight pyramid networks for image deraining," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 6, pp. 1794–1807, 2020.
- [18] K. Iqbal, M. Odetayo, A. James, R. A. Salam, and A. Z. H. Talib, "Enhancing the low quality images using unsupervised colour correction method," in *Proc. IEEE Int. Conf. Syst. Man Cybern.*, 2010, pp. 1703–1709.
- [19] A. S. A. Ghani and N. A. M. Isa, "Underwater image quality enhancement through integrated color model with rayleigh distribution," *Appl. Soft Comput.*, vol. 27, pp. 219–230, 2015.
- [20] C. Ancuti, C. O. Ancuti, T. Haber, and P. Bekaert, "Enhancing underwater images and videos by fusion," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 81–88.
- [21] X. Fu, Z. Fan, M. Ling, Y. Huang, and X. Ding, "Two-step approach for single underwater image enhancement," in *Proc. Int. Symp. Intell. Signal Process. Commun. Syst.*, 2017, pp. 789–794.
- [22] X. Fu, P. Zhuang, Y. Huang, Y. Liao, X.-P. Zhang, and X. Ding, "A retinex-based enhancing approach for single underwater image," in *Proc. IEEE Int. Conf. Image Process.*, 2014, pp. 4572–4576.
- [23] S. Zhang, T. Wang, J. Dong, and H. Yu, "Underwater image enhancement via extended multi-scale retinex," *Neurocomputing*, vol. 245, pp. 1–9, 2017.
- [24] C. O. Ancuti, D. Ancuti, C. Vleeschouwer, and P. Bekaert, "Color balance and fusion for underwater image enhancement," *IEEE Trans. Image Process.*, vol. 27, no. 1, pp. 379–393, 2019.
- [25] C. Ancuti, D. Ancuti, C. Vleeschouwer, R. Garcia, and A. C. Bovik, "Multi-scale underwater de-scattering," in *Proc. IEEE Int. Conf. Pattern Recognit.*, 2016, pp. 4202–4207.
- [26] C. O. Ancuti, C. Ancuti, C. De Vleeschouwer, and M. Sbert, "Color channel compensation (3c): A fundamental pre-processing step for image enhancement," *IEEE Trans. Image Process.*, vol. 29, pp. 2653–2665, 2020.
- [27] R. Liu, X. Fan, M. Zhu, M. Hou, and Z. Luo, "Real-world underwater enhancement: Challenges, benchmarks, and solutions under natural light," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 12, pp. 4861–4875, 2020.
- [28] K. He, J. Sun, and X. Tang, "Single image haze removal using dark channel prior," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 12, pp. 2341–2353, 2010.
- [29] P. L. Drews, E. R. Nascimento, S. S. Botelho, and M. F. M. Campos, "Underwater depth estimation and image restoration based on single images," *IEEE Comput. Graph. Appl.*, vol. 36, no. 2, pp. 24–35, 2016.
- [30] Y.-T. Peng, K. Cao, and P. C. Cosman, "Generalization of the dark channel prior for single image restoration," *IEEE Trans. Image Process.*, vol. 27, no. 6, pp. 2856–2868, 2018.
- [31] C. 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 Trans. Image Process.*, vol. 25, no. 12, pp. 5664–5677, 2016.
- [32] D. Akkaynak and T. Treibitz, "Sea-thru: A method for removing water from underwater images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 1682–1691.
- [33] S. Serikawa and H. Lu, "Underwater image dehazing using joint trilateral filter," *Comput. Electr. Eng.*, vol. 40, no. 1, pp. 41–50, 2014.
- [34] D. Berman, D. Levy, S. Avidan, and T. Treibitz, "Underwater single image color restoration using haze-lines and a new quantitative dataset (early access)," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2020.
- [35] Y. Song, J. Li, X. Wang, and X. Chen, "Single image dehazing using ranking convolutional neural network," *IEEE Trans. Multimedia*, vol. 20, no. 6, pp. 1548–1560, 2017.
- [36] Z. Jin, M. Z. Iqbal, D. Bobkov, W. Zou, X. Li, and E. Steinbach, "A flexible deep cnn framework for image restoration," *IEEE Trans. Multimedia*, vol. 22, no. 4, pp. 1055–1068, 2019.
- [37] C. Li, R. Cong, S. Kwong, J. Hou, H. Fu, G. Zhu, D. Zhang, and Q. Huang, "Asif-net: Attention steered interweave fusion network for rgb-d salient object detection," *IEEE Trans. Cybern.*, vol. 51, no. 1, pp. 88–100, 2020.
- [38] Y. Zheng, Y. Xie, Y. Qu, X. Yang, C. Li, and Y. Zhang, "Scale robust deep oriented-text detection network," *Pattern Recognit.*, vol. 102, p. 107180, 2020.
- [39] P. Li, D. Wang, L. Wang, and H. Lu, "Deep visual tracking: Review and experimental comparison," *Pattern Recognit.*, vol. 76, no. 7, pp. 323–338, 2018.
- [40] R. Liu, Y. Zhang, S. Cheng, X. Fan, and Z. Luo, "A theoretically guaranteed deep optimization framework for robust compressive sensing mri," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, 2019, pp. 4368–4375.
- [41] X. Fu, J. Huang, D. Zeng, Y. Huang, X. Ding, and J. Paisley, "Removing rain from single images via a deep detail network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 3855–3863.
- [42] C. Li, C. Guo, and C. C. Loy, "Learning to enhance low-light image via zero-reference deep curve estimation (early access)," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2021.
- [43] C. Li, S. Anwar, and F. Porikli, "Underwater scene prior inspired deep underwater image and video enhancement," *Pattern Recognit.*, vol. 98, p. 107038, 2020.
- [44] Y. Guo, H. Li, and P. Zhuang, "Underwater image enhancement using a multiscale dense generative adversarial network," *IEEE J. Ocean. Eng.*, vol. 45, no. 3, pp. 862–870, 2020.
- [45] 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 Robot. Autom. Lett.*, vol. 3, no. 1, pp. 387–394, 2017.
- [46] C. Li, C. Guo, W. Ren, R. Cong, J. Hou, S. Kwong, and D. Tao, "An underwater image enhancement benchmark dataset and beyond," *IEEE Trans. Image Process.*, vol. 29, pp. 4376–4389, 2019.
- [47] C. Fabbri, M. J. Islam, and J. Sattar, "Enhancing underwater imagery using generative adversarial networks," in *Proc. IEEE Int. Conf. Rob. Autom.*, 2018, pp. 7159–7165.
- [48] H. Lu, Y. Li, T. Uemura, H. Kim, and S. Serikawa, "Low illumination underwater light field images reconstruction using deep convolutional neural networks," *Future Gener. Comput. Syst.*, vol. 82, pp. 142–148, 2018.
- [49] C. Li, S. Anwar, J. Hou, R. Cong, C. Guo, and W. Ren, "Underwater image enhancement via medium transmission-guided multi-color space embedding," *IEEE Trans. Image Process.*, vol. 30, pp. 4985–5000, 2021.

- [50] M. Hou, R. Liu, X. Fan, and Z. Luo, "Joint residual learning for underwater image enhancement," in *Proc. IEEE Int. Conf. Image Process.*, 2018, pp. 4043–4047.
- [51] Z. Wang, E. P. Simoncelli, and A. C. Bovik, "Multiscale structural similarity for image quality assessment," in *IEEE Asilomar Conference on Signals, Systems and Computers*, vol. 2, 2003, pp. 1398–1402.
- [52] M. Iqbal, M. M. Riaz, S. S. Ali, A. Ghafoor, and A. Ahmad, "Underwater image enhancement using laplace decomposition (early access)," *IEEE Geosci. Remote. Sens. Lett.*, 2020.
- [53] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *Proc. Int. Conf. Mach. Learn.*, vol. 30, no. 1, 2013, p. 3.
- [54] T. Tong, G. Li, X. Liu, and Q. Gao, "Image super-resolution using dense skip connections," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 4799–4807.
- [55] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *Proc. Eur. Conf. Comput. Vis.* Springer, 2016, pp. 694–711.
- [56] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Int. Conf. Learn. Represent.*, 2015.
- [57] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Represent.*, 2015.
- [58] A. Galdran, D. Pardo, A. Picón, and A. Alvarez-Gila, "Automatic red-channel underwater image restoration," *J Vis Commun Image Represent.*, vol. 26, pp. 132–145, 2015.
- [59] Y.-T. Peng and P. C. Cosman, "Underwater image restoration based on image blurriness and light absorption," *IEEE Trans. Image Process.*, vol. 26, no. 4, pp. 1579–1594, 2017.
- [60] D. Berman, T. Treibitz, and S. Avidan, "Diving into haze-lines: Color restoration of underwater images," in *Proc. Br. Mach. Vis. Conf.*, vol. 1, no. 2, 2017.
- [61] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, June 2019.
- [62] Y. Wang, N. Li, Z. Li, Z. Gu, H. Zheng, B. Zheng, and M. Sun, "An imaging-inspired no-reference underwater color image quality assessment metric," *Comput. Electr. Eng.*, vol. 70, pp. 904–913, 2018.
- [63] L. Zhang, L. Zhang, and A. C. Bovik, "A feature-enriched completely blind image quality evaluator," *IEEE Trans. Image Process.*, vol. 24, no. 8, pp. 2579–2591, 2015.
- [64] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "Yolov4: Optimal speed and accuracy of object detection," *arXiv preprint arXiv:2004.10934*, 2020.
- [65] S. Zhou, J. Zhang, J. Pan, H. Xie, W. Zuo, and J. Ren, "Spatio-temporal filter adaptive network for video deblurring," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 2482–2491.