IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 12 March 2024, accepted 23 May 2024, date of publication 28 May 2024, date of current version 5 June 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3406944

RESEARCH ARTICLE

Efficient Image Enhancement via Representative Color Transform

YEJI JEON, (Student Member, IEEE), AND HANUL KIM, (Member, IEEE)
Department of Applied Artificial Intelligence, Seoul National University of Science and Technology, Seoul 01811, South Korea

Corresponding author: Hanul Kim (hukim@seoultech.ac.kr)

This work was supported by Seoul National University of Science and Technology.

ABSTRACT We propose an improved representative color transformation (RCT++), which is an effective framework to describe complex color transformations between low- and high-quality images. We identify the representative colors and features of the input image. For each representative color, we estimate a transformed color that represents its enhanced version. Then, we enhance all input colors by interpolation, taking into account the similarity between input pixels and representative features. We further improve the original RCT framework by introducing the reconstruction term, which clarifies the representative colors, and the entropy term, which diversifies the representative features. Finally, we develop the enhancement network to achieve fast and lightweight image enhancement. Comprehensive experiments on various image enhancement tasks validate our superiority in both effectiveness and efficiency. Our method exceeds recent state-of-the-art methods in efficient image enhancement on MIT-Adobe 5K, Low Light, and Underwater Image Enhancement Benchmark datasets, with comparable computational and memory costs.

INDEX TERMS Image enhancement, efficient image enhancement.

I. INTRODUCTION

Image quality is one of the fundamental concept in computer vision and image processing. People want to capture their daily lives in visually appealing photographs. The performance of computer vision systems such as object detection [1] and segmentation [2], [3] relies heavily on the quality of input images. Regrettably, the visual quality of images can be easily degraded due to inadequate shooting environments like low-light conditions and camera sensor limitations. Therefore, image enhancement techniques become more popular, which automatically retouch input images to improve their visual quality.

Recently, image enhancement studies have primarily focused on the learning-based approach that involves training models using pairs of low-quality and high-quality images, aiming to learn the mapping between them. Especially, with advances in datasets [4], [5], [6] and deep learning [7], most state-of-the-art techniques [8], [9], [10], [11], [12], [13], [14], [15] train deep neural networks to learn a complex pixel-wise mapping between low-quality and high-quality images and yield promising enhancement results. However, their networks come with substantial computational costs. Considering that image enhancement algorithms often serve as a pre-processing step in various computer vision systems, it is crucial to develop them to be fast and lightweight.

To address this issue, many attempts [16], [17], [18], [19], [20], [21] have been made to learn a color transformation, which defines a color mapping between input and output images, rather than pixel-wise mapping between them. This approach models a color transformation controlled by only a few parameters. Then, instead of outputting enhanced images directly, neural networks predict these parameters, resulting in a significant improvement in the computational complexity of learning-based image enhancement algorithms. However, despite the advantages of this approach, developing an effective color transformation model remains a challenge. For instance, some methods [18], [19], [20], [21] design a color mapping as a weighted combination of predefined 3-dimensional lookup tables (LUTs) [18], which are data structures to describe a color mapping. Consequently, their networks only need to

The associate editor coordinating the review of this manuscript and approving it for publication was Davide Patti.

predict a small number of LUT weights, enabling real-time processing. Nevertheless, these LUT-based methods may encounter difficulties in accurately emulating complex color mappings due to the limitations imposed by the less flexible predefined LUTs.

In this paper, we propose an efficient image enhancement algorithm called the improved representative color transformation (RCT++), which is an extension of our previous work [22]. The RCT++ algorithm estimates the representative features associated with the representative colors found in the input image, as well as the transformed colors that indicate the enhanced representative colors in the output image. Subsequently, it interpolates the output image from transformed colors based on the similarities between the representative features and per-pixel features. Compared to the original RCT [22], our method has three major improvements: First, we clarify the concept of representative colors by incorporating the reconstruction loss into the RCT framework. This encourages the representative features to correspond to the significant color of the input image. Second, we introduce the entropy term to measure the diversity of representative features. By maximizing this entropy term, our scheme can estimate more diverse representative features, which are useful to enhance minority colors in the input image. Third, we design an efficient enhancement network that performs the RCT++ algorithm for fast and lightweight image enhancement. Specifically, we implement RCT++ algorithm based on depth-wise separable convolution [23] to lighten our network. Experimental results on three datasets [4], [5], [6] with different characteristics demonstrate that our RCT++ outperforms existing algorithms in efficient image enhancement with comparable computational costs and parameters. The main contributions are summarized as follows:

- We propose the novel color transform, RCT++, for image enhancement. The RCT++ improves the original RCT [22] by incorporating the reconstruction term and the entropy term.
- We develop an efficient image enhancement network. Our network contains about 131K parameters and takes 3ms to process an image of 480 × 720 resolution.
- RCT++ outperforms state-of-the-art methods in efficient image enhancement on three datasets collected in different shooting environments.

The rest of this paper is organized as follows. Section II reviews related works. Section III describes the proposed algorithm, and Section IV discusses experimental results. Finally, Section V draw a conclusion.

II. RELATED WORK

Image enhancement is a long-standing problem with wide applications. Therefore, lots of attempts have been made to improve the image enhancement performance. In this section, we briefly review relevant studies on learning-based image enhancement and efficient image enhancement which are closely related to our work.

A. LEARNING-BASED IMAGE ENHANCEMENT

Early learning-based image enhancement methods [24], [25], [26] mainly depend on hand-crafted features, such as intensity, brightness, and the amount of highlight, or pre-fixed mappings. Dale et al. [24] introduced visual context based on scale-invariant feature transform (SIFT) [27]. Wang et al. [25] defined tone and color adjustments given a set of examples. Hwang et al. [26] searched contextually similar images from examples and adjusted the input image via corresponding transformation functions. However, due to the limited representation of hand-crafted features, it is difficult for these methods to reliably enhance various input images.

Deep neural networks allow image enhancement methods to learn complex mapping between low-quality and high-quality images. Therefore, over the past decade, many deep learning-based image enhancement methods [8], [10], [11], [14], [15], [28], [29], [30], [31] have been proposed. Yan et al. [8] learned the feature descriptor for each input image pixel to consider semantic information in retouching. Lore et al. [10] stacked a sparse denoising autoencoder, which can learn to adaptively enhance and denoise from synthetically darkened and noise-added training examples. However, these methods [8], [10] often fail to exploit high-level context for image enhancement due to the small receptive field of their networks.

Chen et al. [28] employed the encoder-decoder structure [32] for image enhancement, in which the encoder gradually performs down-sampling to increase the size of receptive fields, and the decoder restores the original resolution while enhancing images. Yang et al. [11] developed two encoder-decoder structures for low-light image enhancement. Based on the retinex theory [33], Wang et al. [29] enhanced an input image by decomposing it into reflectance and illumination, and then improving the illumination. Xu et al. [30] proposed the frequency-based decomposition to enhance low-light images. Kim et al. [31] adopted the encoder-decoder structure to perform a personalized image enhancement. Tu et al. [14] proposed a general image processing network through multi-axis multilayer perceptron (MLP). Cai et al. [15] designed a transformer-based network to leverage the large receptive field of the attention mechanism [34]. These methods [11], [14], [15], [28], [29], [30], [31] yields the promising enhanced results. However, it requires many parameters and computational costs, making it difficult to apply to various applications.

B. EFFICIENT IMAGE ENHANCEMENT

Efficient image enhancement aims to minimize high computational burdens while maintaining the high visual quality of output images. There are two streamlines for efficient image enhancement. The first focuses on efficient color transformation modeling, defining color mappings between input and output images rather than pixel-wise mappings between them. Deng et al. [16] defined the piece-wise intensity curve controlled by only a few parameters predicted by the neural network. And the small size of the output space reduces the complexity of learning-based image enhancement algorithms. Similarly, Kim et al. [17] presented the learnable non-monotonic intensity transformation for both paired and unpaired image enhancement.

Look-up tables (LUTs) are another popular way to model color transformation, which are efficient data structures enabling real-time enhancement [18], [19], [20], [21]. Zeng et al. [18] stored non-linear color transformation in a 3D lattice, which can be loaded through simple indexing and leveraged by affine transforms. Wang et al. [20] integrated 1D-LUTs and 3D-LUTs, considering image-level and pixel-wise transforms, respectively. Yang et al. [21] designed an effective sampling strategy to improve the quality of the output image Despite their efforts to improve enhancement quality, they demonstrate limited capability to estimate highly non-linear retouching mappings due to predefined transformations on LUTs. In contrast, the proposed method estimates adaptive representative colors according to the input image and predicts color transformation for each representative color.

The second streamline attempts to design a lightweight architecture. Gharbi et al. [35] used a low-resolution image to predict bilateral coefficients. It then applied an affine transform to the original resolution. Ma et al. [36] relieved the computational burden of cascaded blocks via sharing weights. In this line, we compose the network using depth-wise convolution layers resulting in the small-sized model.

III. PROPOSED ALGORITHM

In this section, we first describe the original RCT algorithm [22] and highlight its limitations. We then propose the improved RCT (RCT++) algorithm and develop its network architecture for efficient image enhancement. Finally, we present the loss functions used to train our network.

A. REPRESENTATIVE COLOR TRANSFORM

Given an RGB input image $X \in \mathbb{R}^{h\times w\times 3}$, where h and w are the height and width of the image, we extract its feature map $Z \in \mathbb{R}^{h\times w\times c}$ to embed high-level context for image enhancement:

$$Z = f_\theta(X) \tag{1}$$

Here, c is the dimension of the feature space, and $f_\theta(\cdot)$ is an embedding function parameterized by θ. In practice, $f_\theta(\cdot)$ is a deep neural network.

RCT estimates n representative features $F \in \mathbb{R}^{n\times c}$ corresponding to representative colors of the input image. Also, it predicts n transformed colors $C_t \in \mathbb{R}^{n\times 3}$ that are enhanced representative colors. These are given by

$$F = g_\phi(Z) \tag{2}$$

$$C_t = g_\psi(Z) \tag{3}$$

where $g_\phi(\cdot)$ and $g_\psi(\cdot)$ are mapping functions with parameters ϕ and ψ, respectively.

Note that the transformed colors only describe the color mapping for N representative colors. To determine the enhanced colors for all input colors, we interpolate them through the scaled-dot product attention mechanism [34]. Specifically, we consider these enhanced colors as the weighted sum of transformed colors, in which the weights are proportional to the feature similarities between the input colors and representative colors. So, the weight matrix $A \in \mathbb{R}^{hw\times n}$ between the input features and representative features is given by

$$A = \mathrm{softmax}\left(\frac{ZF^T}{\tau}\right) \tag{4}$$

where τ is the temperature parameter to control the confidence of the resulting distribution, and $a_{ij} \in A$ represents the similarity between the ith pixel in the input image feature and the jth representative feature. Finally, the enhanced image $\widehat{Y}$ is given by

$$\widehat{Y} = AC_t \tag{5}$$

Since the RCT is fully described by the representative features F, transformed colors C_t, and the image feature map Z, it is crucial to carefully design these components to achieve successful image enhancement. However, the original RCT [22] has a drawback as it lacks an explicit mechanism to guarantee that representative colors become the actual important colors of the input image. Moreover, the original RCT does not encourage diversity in representative features. As a result, it may face challenges when enhancing some colors appearing rarely in the input image, using a weighted sum of transformed colors.

B. IMPROVED REPRESENTATIVE COLOR TRANSFORM

To overcome the first limitation, we explicitly define the representative colors of the input image. We introduce a function denoted as $g_\omega(\cdot)$ with learnable parameters ω. The function takes the image feature map Z and estimates n representative colors $C_r \in \mathbb{R}^{n\times 3}$:

$$C_r = g_\omega(Z) \tag{6}$$

Subsequently, we restore the input colors by computing the weighted sum of the representative colors, using the same weights as in (4). Thus, the reconstructed image $\widehat{X}$ is given by

$$\widehat{X} = AC_r \tag{7}$$

We train the parameters ω to minimize the reconstructed error between the input image and the reconstructed image. Unlike the original RCT, our scheme explicitly learns that the estimated representative colors serve as actual representatives of input colors. Consequently, the representative feature is also guaranteed to be a feature of the representative color.

Next, we address the second problem of the original RCT by introducing an entropy term that quantifies the diversity

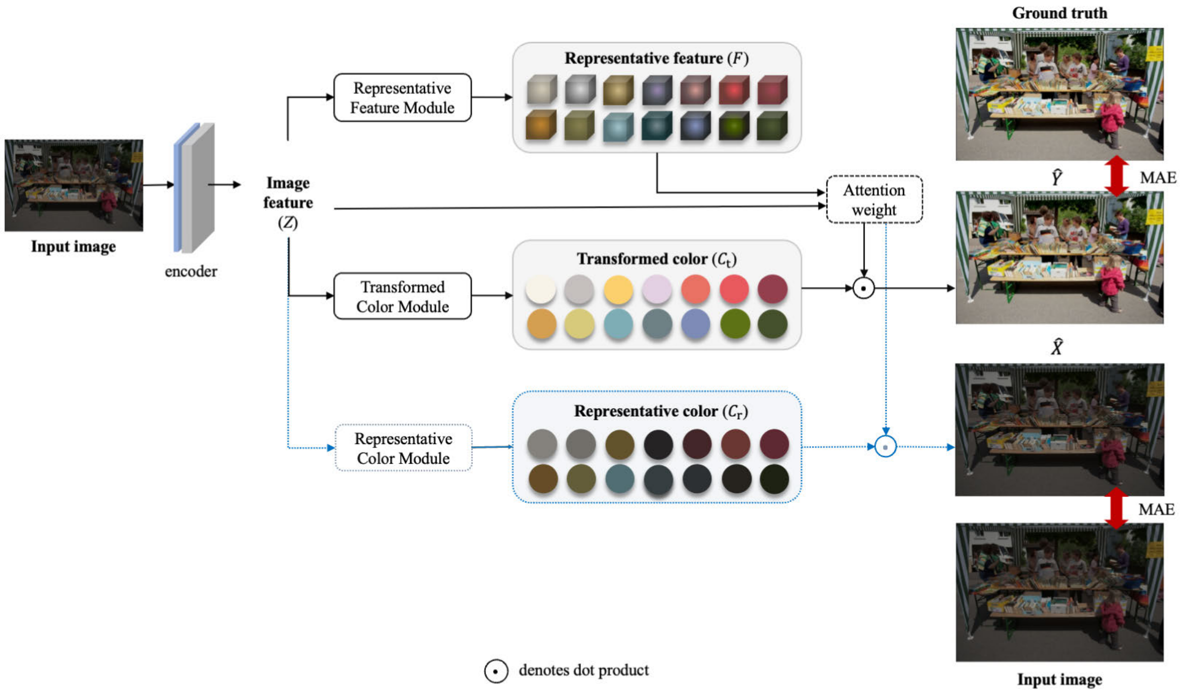


FIGURE 1. An overview of the proposed RCT++ process. The blue dotted line, which reconstructs the input image, is only used in the training.

among representative features. To this end, we compute a similarity matrix S whose each element s_{ij} indicates the cosine similarity between the ith representative feature $\mathbf{f}_i$ and jth representative features $\mathbf{f}_j$:

$$s_{ij} = \frac{\mathbf{f}_i \mathbf{f}_j^T}{||\mathbf{f}_i||||\mathbf{f}_j||} \tag{8}$$

We set the diagonal elements of the similarity matrix to zero using a masking operation to constrain the contribution of self-similarity. Then, we apply a softmax function to the masked similarity matrix $\tilde{S}$ to obtain the distribution matrix P, given by

$$P = \text{softmax}(\tilde{S}) \tag{9}$$

Since each row of the matrix is a probability distribution, we can define the entropy of each row. Finally, the entropy term of representative colors is given by the average entropy of distributions:

$$\text{entropy} = -\frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{n} p_{ij} \log p_{ij} \tag{10}$$

where $\log p_{ij}$ indicates the element at the ith row and jth column of the distribution matrix. Algorithm 1 summarizes this procedure to obtain the entropy term.

Algorithm 1 An Entropy of Representative Features

```
Input: $F \in \mathbb{R}^{c \times n}$
Output: entropy $\in \mathbb{R}$
$S$ ← Compute the similarity matrix ;            // (8)
$\tilde{S}$ ← Set the diagonal elements of $S$ to 0 ;
$P$ ← Compute the distribution matrix ;          // (9)
entropy ← Compute the entropy term ;             // (10)
```

Note that the entropy term is maximized when each representative feature is dissimilar to each other. By incorporating this entropy term into the RCT method, we effectively promote the diversity of representative features, enabling the RCT method to handle colors that exist in the input image as a minority. In Section IV, we will provide experimental results to demonstrate the effectiveness of the entropy term.

C. ENHANCEMENT NETWORK

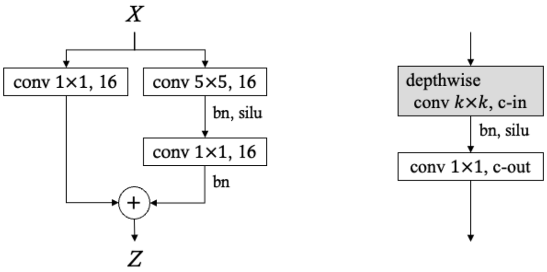
We develop an enhancement network to implement the proposed RCT++ method, aiming for efficient image enhancement. Figure 1 illustrates the overall architecture of our enhancement network, which comprises four modules: an encoder, a representative feature module, a transformed color module, and a representative color module. These modules correspond to f_θ, g_ϕ, g_ψ, and g_ω in our RCT++ framework, respectively. Table 1 summarizes the detailed specification of each module.

1) ENCODER

The encoder embeds the input image X into the image feature Z for the RCT++ process. As shown in Figure 2, the encoder is a residual block [37] with two branches. The first branch is a single convolution layer with 1 × 1 filters, employed to transform input pixel colors into feature vectors.

TABLE 1. Specification for the proposed enhancement network. Each row describes a stage and an output size.

	Operation	Output shape (H× W× C)
A. Encoder		
1	Conv2d, 1 × 1	$H \times W \times 16$
2a	Conv2d, 5 × 5	$H \times W \times 16$
2b	Conv2d, 1 × 1	$H \times W \times 16$
3	Add	$H \times W \times 16$
B. Representative Feature		
1	Resize	256 × 256 × 16
2	Dwconv2d, 4 × 4	64 × 64 × 64
3	Dwconv2d, 3 × 3	32 × 32 × 64
4	Dwconv2d, 3 × 3	16 × 16 × 64
5	Dwconv2d, 3 × 3	8 × 8 × 64
6	Global Avg. Pool	64
7	MLP	1024
8	Reshape	64 × 16
C. Transformed / Representative Color		
1	Resize	256 × 256 × 16
2	Dwconv2d, 4 × 4	64 × 64 × 64
3	Dwconv2d, 3 × 3	32 × 32 × 64
4	Dwconv2d, 3 × 3	16 × 16 × 64
5	Dwconv2d, 3 × 3	8 × 8 × 64
6	Global Avg. Pool	64
7	MLP	192
8	Reshape	64 × 3



(a) Encoder

(b) Dwconv2d, $k \times k$

FIGURE 2. (a) The detailed structure of the encoder module. (b) The detailed structure of depth-wise separable convolution block.

The second branch, on the other hand, incorporates a 5 × 5 convolution, a batch normalization [38], a sigmoid linear unit (SiLU) activation [39], and a 1 × 1 convolution layers. Then, the encoder produces the image feature Z by merging the outputs from both branches. Note that the second branch is essential to improving the RCT++ results. Despite RCT++ being a color transformation model, we have found that exploiting the local structure of input colors is beneficial. This facilitates the mapping of the same input colors to different transformed colors based on the neighboring context. Further discussions regarding this will be presented in Section IV.

2) REPRESENTATIVE FEATURE MODULE

The representative feature module takes the image feature map Z and estimates the representative features F. Specifically, the representative feature module first reduces the spatial resolution of the feature map to 256 × 256. This downsampling brings two benefits: It helps decrease the computational cost, making the RCT++ process more efficient. Also, it enables the subsequent convolution layers to have larger receptive fields, which aid in the extraction of the global context, leading to more effective image enhancement.

Next, the module feeds the resized feature map into four depth-wise separable convolution (Dwconv) blocks. As depicted in Figure 2, each Dwconv block is made up of a depth-wise convolution and a point-wise convolution, which leads to fewer parameters and operations compared to standard convolutions. These computational and memory gains become more significant in the later stages of the network when the number of feature channels increases. Subsequently, the module performs a global average pooling. It then estimates $c = 16$ dimensional $n = 64$ representative features using a multilayer perceptron (MLP) block, which consists of a linear layer, a layer normalization [40], a SiLU activation, and another linear layer. Here, we set the number of neurons of the first and second linear layers to 64 and 1024.

3) TRANSFORMED/REPRESENTATIVE COLOR MODULE

Our design involves the prediction of two color sets: one for improving the input images and another for restoring them. To achieve this, we design both the transformed and representative color modules to have identical architectures. As specified in Table 1, these architectures are the same as the representative feature module, except they predict n colors by adjusting the number of neurons in the second linear layer of the MLP block. Note that the representative color module only works during the training phase and thus does not increase computational or memory load during the testing phase.

D. LOSS FUNCTION

We train the enhancement network by minimizing the total loss, which encompasses multiple loss components: a color loss ($\mathcal{L}_{\text{col}}$), a reconstruction loss ($\mathcal{L}_{\text{rec}}$), an entropy loss ($\mathcal{L}_{\text{ent}}$), and a grid frequency loss ($\mathcal{L}_{\text{freq}}$). More precisely, we define the total loss as

$$\mathcal{L} = \mathcal{L}_{\text{col}} + \mathcal{L}_{\text{rec}} + \mathcal{L}_{\text{ent}} + \mathcal{L}_{\text{freq}} \tag{11}$$

The individual loss terms are carefully designed to capture various aspects of ground-truth images. Let us describe each term subsequently.

1) COLOR LOSS

The color loss is the mean absolute error between the ground-truth image Y and the predicted image $\widehat{Y}$

$$\mathcal{L}_{\text{col}} = ||Y - \widehat{Y}||_1 \tag{12}$$

The color loss encourages the predicted image to be close to the ground-truth image in the color space. For this reason, we employ color loss as the primary loss function in our work.

2) RECONSTRUCTION LOSS

The reconstruction loss is the mean absolute error between the input image X and the reconstructed image $\widehat{X}$

$$\mathcal{L}_{\text{rec}} = ||X - \widehat{X}||_1 \tag{13}$$

It is worth noting that inaccurate representative colors make it difficult to correctly rebuild the input image, resulting in increased reconstruction loss. Hence, the reconstruction loss enforces that the representative colors become an actual significant color set of the input image.

3) ENTROPY LOSS

We set the entropy loss to the inverse of the entropy term.

$$\mathcal{L}_{\text{ent}} = \frac{1}{\text{entropy} + \epsilon} \tag{14}$$

Thus, minimizing entropy loss encourages the scattering of representative features F, i.e., increases their entropy. Here, $\epsilon = 0.00001$ for numerical stability.

4) GRID FREQUENCY LOSS

To compute the grid frequency loss, we decompose images into m non-overlapping grids. Let Y_i and $\widehat{Y_i}$ be the i th grid of ground-truth and predicted images, respectively. We then define the grid frequency loss as follows:

$$\mathcal{L}_{\text{freq}} = \sum_{i=1}^{m} ||\mathcal{F}(Y_i) - \mathcal{F}(\widehat{Y_i})||_1 \tag{15}$$

where $\mathcal{F}(\cdot)$ indicates the 2D Fast Fourier Transform (FFT) function. Likewise, with other losses [14], [41], [42] based on the frequency domain, our loss enforces that the enhancement network retains high-frequency details. However, our loss has an additional advantage in that it can concentrate on local high-frequency details. We empirically set the hyperparameter m to 4.

IV. EXPERIMENTS

In this section, we present a comprehensive evaluation of the proposed method. We compare our method with state-of-the-art methods on three datasets: MIT-Adobe 5K (Adobe5K) [4], Low-Light (LoL) [5], and Underwater Image Enhancement Benchmark (UIEB) [6]. These datasets are collected in a diverse range of shooting environments, providing a faithful evaluation of our method. Furthermore, we study the impact of the proposed components. For quantitative comparison, we use three performance metrics: peak signal-to-noise ratio (PSNR), structural similarity index measure (SSIM), and learned perceptual image patch similarity (LPIPS) [43]. These metrics measure the color, structural, and perceptual similarity between the enhanced image and the ground-truth image, respectively. Also, we evaluate the efficiency of our method compared to existing methods in terms of a number of parameters and run-time cost.

A. DATASETS

1) MIT-ADOBE 5K

The Adobe5K [4] dataset contains 5,000 image pairs. Each pair consists of a low-quality image and manually retouched versions by five experts (A/B/C/D/E). For experiments, we use the enhanced images improved by expert C as the ground truth, following the experiment setting of recent image enhancement methods [18], [19], [21]. We split the dataset into 4,500 pairs for training and 500 pairs for validation. As done in [18], we resize each image to have 480 pixels on the short side, while maintaining its aspect ratio.

2) LOW LIGHT

The LoL [5] dataset is developed for low-light image enhancement. This dataset comprises 500 pairs of low-light and normal-light images, all of which were taken from real-world scenes. The dataset is divided into 485 pairs for training and 15 pairs for testing. Compared to the Adobe5K dataset, the LoL dataset contains a considerable amount of noise generated during the image capture process. All images in this dataset have a resolution of 400 × 600.

3) UNDERWATER IMAGE ENHANCEMENT BENCHMARK

The UIEB [6] dataset consists of 950 real-world underwater images, 890 of which have their reference images and the others do not. Thus, we divide 890 reference pairs into 800 pairs of training and 90 pairs for our experiments evaluation, following the previous work [22].

B. IMPLEMENTATION DETAILS

For training, we use the AdamW optimizer [44] with an initial learning rate of 0.0005 and a weight decay of 0.05. We set the batch size to 16. The training is done for 600 and 5000 epochs for experiments on Adobe5k and the other datasets, respectively. We schedule a learning rate via cosine annealing. For data augmentation, we randomly crop images and then resize them to 256 × 256. Subsequently, we randomly apply horizontal and vertical flips. All experiments are carried out on an NVIDIA A6000 GPU.

C. MAIN RESULTS

We compare the proposed RCT++ with recent state-of-the-art methods. For evaluation, we obtain the results of existing methods by executing the published codes of these algorithms. If the code is not available, we use the results reported in its paper.

1) RESULTS ON MIT-ADOBE 5K

Table 2 compares the proposed RCT++ with recent efficient image enhancement algorithms on the Adobek5K dataset [4]: HDRNet [35], 3D-LUT [18], Sep-LUT [19], and AdaInt [21]. In Table 2, our method establishes the best results on all three metrics with large margins. Compared to AdaInt [21], which is the LUT-based efficient image enhancement algorithm giving the second best algorithm in Table 2, our method

TABLE 2. Quantitative results on the Adobe5K [4] dataset. The best and second results are highlighted in bold and underline.

Method	PSNR ↑	SSIM ↑	LPIPS ↓	#Parameters	Runtime (ms)
UPE [29]	21.88	0.853	-	927.1K	4.27
HDRNet [35]	24.66	0.915	0.084	483.1K	3.49
3D-LUT [18]	25.29	0.923	0.063	593.5K	1.17
DSN [45]	23.75	0.875	0.061	4.42M	94.33
AdaInt [21]	25.49	0.926	0.052	619.7K	1.29
Sep-LUT [19]	25.47	0.921	-	**119.8K**	**1.10**
FECNet [46]	24.39	0.881	0.085	153.6K	14.80
Ours	**25.60**	**0.932**	**0.049**	131.4K	2.82

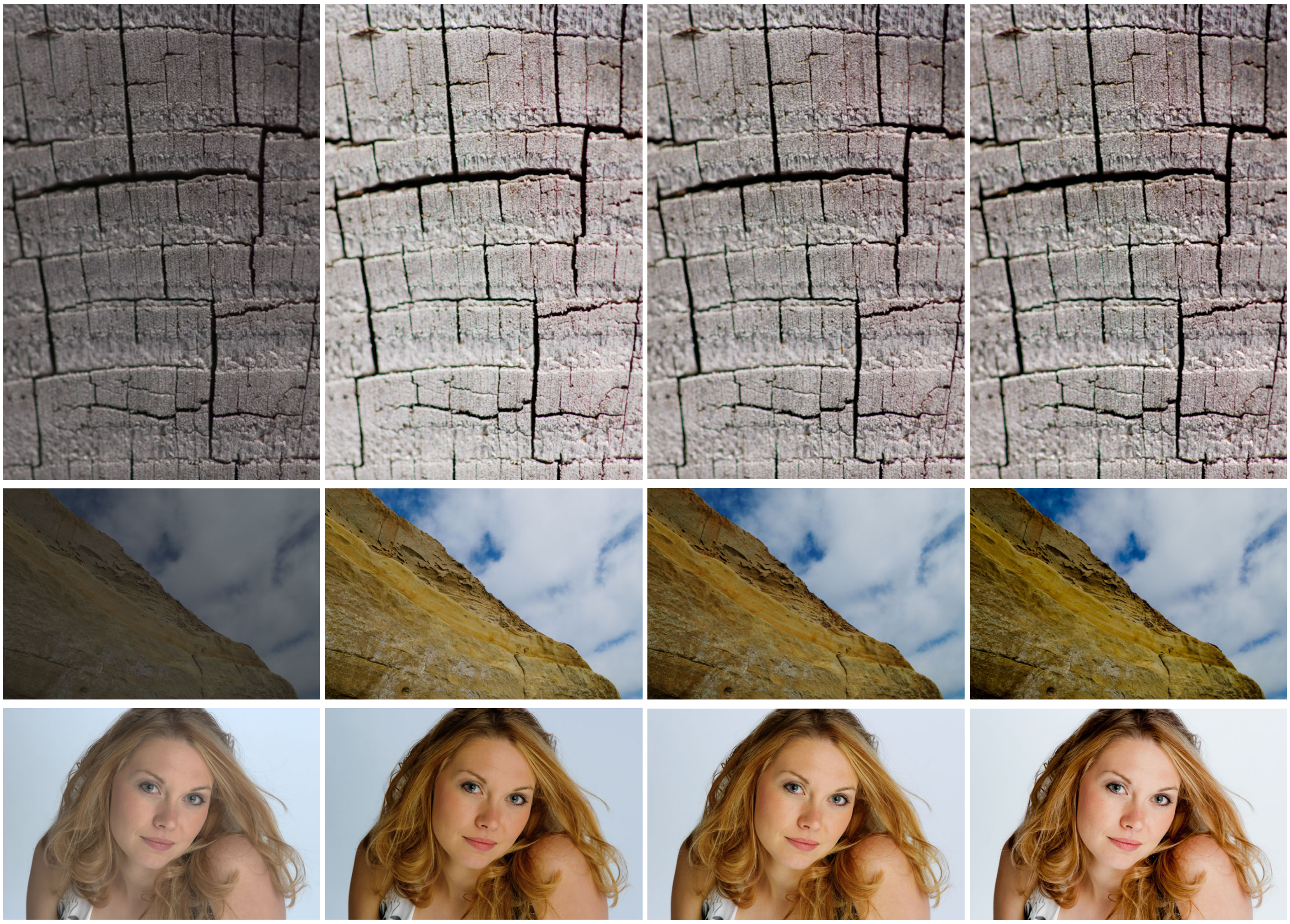

(a) Input
(b) AdaInt [21]
(c) Ours
(d) GT

FIGURE 3. Qualitative comparison with the existing method [21] on the Adobe5K [4] dataset.

provides better scores by 0.11, 0.008, and 0.003 in terms of PSNR, SSIM, and LPIPS, respectively, despite having the five times fewer parameters. In addition, our network has a comparable size to Sep-LUT [19] but demonstrates superior performance. For a comprehensive evaluation, Figure 3 shows the enhanced images of ours and AdaInt. We see that AdaInt fails to estimate accurate color mappings for the three examples due to the limitation of a predefined look-up table. In contrast, our method is the more flexible color transformation model, resulting in more similar images to manually retouched ground-truth images.

2) RESULTS ON LOW LIGHT

We evaluate the performance of our method in extremely low-light shooting condition. Table 3 lists the quantitative comparisons with existing methods [12], [13], [48], [49], [50], [51] on the LoL dataset [5]. Our method achieves the highest PSNR score, meaning the best color enhancement results. Despite Zero-DCE [49] and RUAS [50] demonstrating efficient low-light enhancement networks with the lowest and second-lowest network parameter consumption, respectively, they show poor performance. On the other hand, our method not only utilizes a low number of parameters but

TABLE 3. Quantitative results on the LoL [5] dataset. The best and second results are highlighted in bold and underline.

Method	PSNR ↑	SSIM ↑	LPIPS ↓	#Parameters
RetinexNet [47]	16.77	0.562	0.465	440K
KIND [12]	20.38	0.831	0.159	9.91M
KIND++ [13]	21.80	**0.836**	0.158	8.27M
DRBN [48]	19.86	0.834	0.155	1.12M
Zero-DCE [49]	14.86	0.562	0.335	79.42K
RUAS [50]	18.23	0.717	0.354	**3.44K**
URetinexNet [51]	21.33	0.833	**0.120**	340.9K
Ours	22.36	0.781	0.217	131.4K
+ bilateral filter	**22.47**	0.825	0.156	

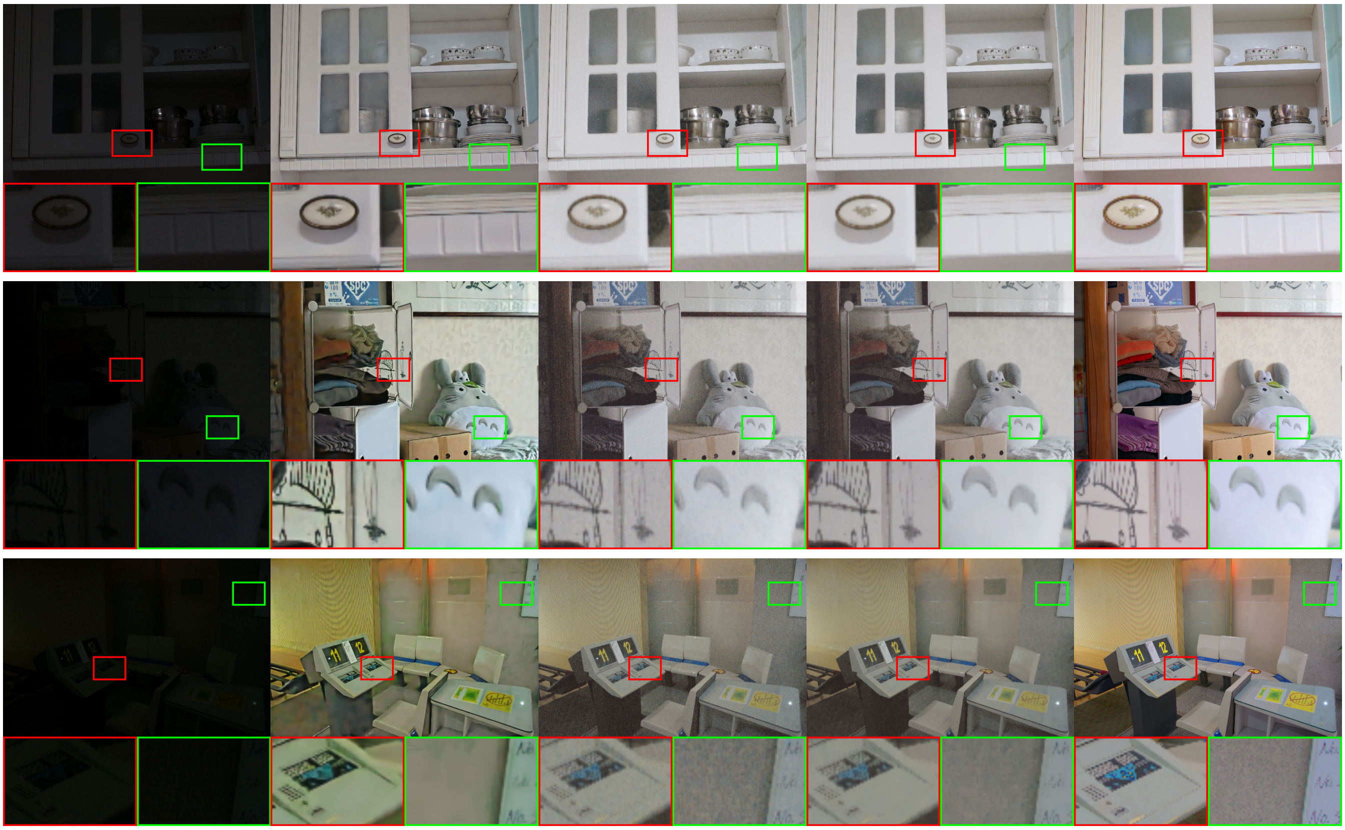

(a) Input (b) KIND++ [13] (c) Ours (d) Ours + BF (e) GT

FIGURE 4. Qualitative comparison with the existing method [13] on the LOL [5] dataset. +BF denotes the post-processed results through bilateral filtering.

also achieves superior performance across various metrics. Our method shows relatively lower performance on SSIM and LPIPS, which are more sensitive to sensor noise due to low-light shooting condition. This is because, the RCT++ models a global color transformation, resulting in less effectiveness in denoising than spatial filtering-based methods. However, this problem can be mitigated by using simple denoising techniques as the post-processing. As shown in Table 3, a simple bilateral filter improves our method in all of the performance metrics. Specifically, PSNR, SSIM, and LPIPS scores increase to 22.47 dB, 0.825, and 0.156, respectively.

Figure 4 qualitatively compares the enhancement results of our method with KIND++ [13], the second best algorithm in Table 3. For all input images, KIND++ [13] fails to faithfully restore the ground-truth images. In contrast, the proposed RCT++ produces enhanced images with color tones more similar to ground-truth images. We also see that our results contain slightly more noise than KIND++. However, it can be suppressed through simple post-processing using a bilateral filter.

3) RESULTS ON UNDERWATER IMAGE BENCHMARK

Finally, we assess the enhancement results of the proposed RCT++ using underwater images to validate its scalability to various shooting environments. Table 4 summarizes the quantitative results of RCT++ and those of the existing algorithms [6], [52], [53], [54] on the UIEB [6] dataset.

TABLE 4. Quantitative results on the UIEB [6] dataset. The best and second results are highlighted in bold and underline.

Method	PSNR ↑	SSIM ↑	LPIPS ↓	#Parameters
WaterNet [6]	19.81	0.861	-	24.81M
FUnIE [52]	17.09	0.728	0.353	7M
Ucolor [53]	20.78	0.871	-	157.4M
PUIE-Net [54]	22.07	0.884	0.166	<u>1.4M</u>
Ours	**23.24**	**0.893**	**0.148**	**131.4K**

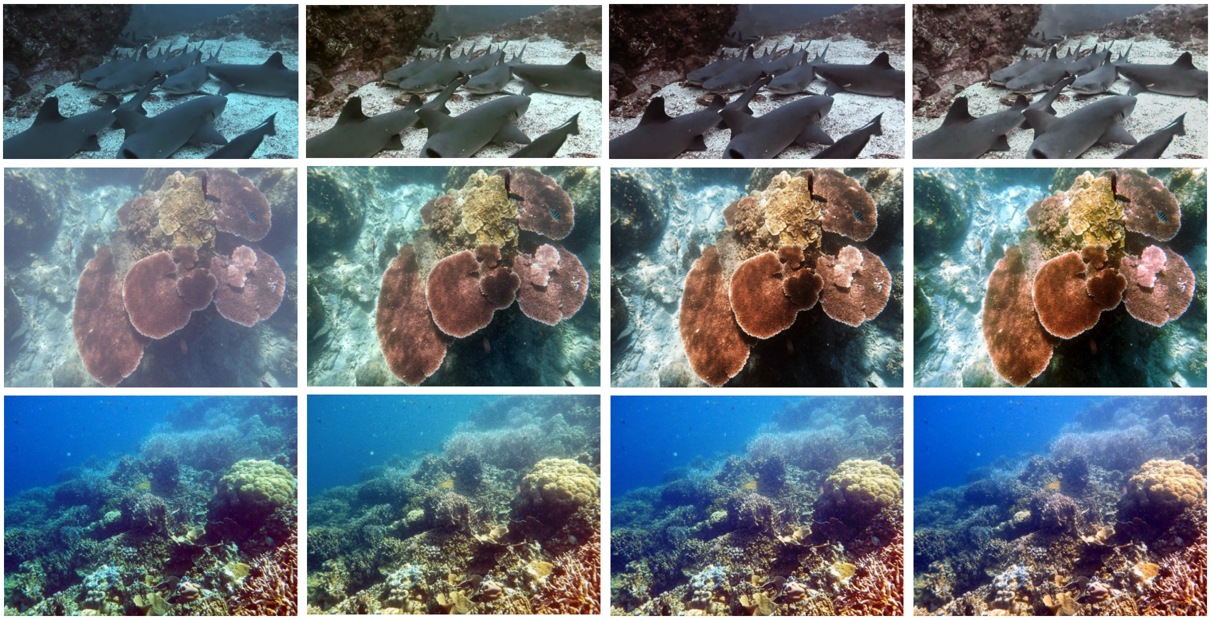

(a) Input (b) PUIE-Net [13] (c) Ours (d) GT

FIGURE 5. Qualitative comparison with the existing method [54] on the UIEB [6] dataset.

In Table 4, our method exceeds conventional algorithms in all of the performance metrics, with significantly fewer parameters. Figure 5 illustrates the enhanced results on the UIEB dataset. We see that RCT++ effectively corrects biased hue caused by underwater shooting environment. In the first and second rows, in which input images are predominantly biased toward blue tone, our method successfully balances the overall hue and restores the original color of the shark and coral. Also, in the third row, we see that our method successfully enhances the input image to a color tone similar to ground-truth.

D. ABLATION STUDY

We perform ablation studies on the Adobe5k [4] dataset to analyze our design. For all ablation studies, we use the same experiment settings in Section IV-B.

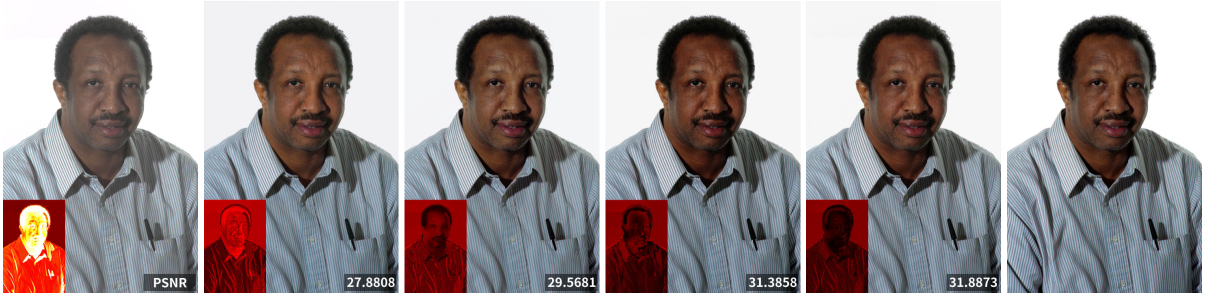
1) LOSS FUNCTIONS

Table 5 reports the performance of the proposed method with different combinations of loss terms. The model trained with only the color loss L_{col} yields the worst results. The grid frequency loss slightly improves the enhancement results. Especially, it boosts the SSIM score by preserving high-frequency details. The reconstruction loss encourages our method to estimate more meaningful representative colors and features, resulting in a more accurate RCT++ process. As a result, the PSNR score, which measures color restoration performance, has improved significantly. The best performance is obtained by incorporating all loss terms. Remarkably, the entropy loss improves the performance of our method in all metrics by diversifying representative features.

TABLE 5. Results on the Adobe5K [4] with different loss functions. The best results are highlighted in bold.

L_{col}	L_{freq}	L_{rec}	L_{ent}	PSNR ↑	SSIM ↑	LPIPS ↓
✓				25.46	0.929	0.050
✓	✓			25.47	0.932	0.050
✓	✓	✓		25.54	0.931	0.049
✓	✓	✓	✓	**25.60**	**0.932**	**0.049**

Figure 6 visualizes the enhanced images and error maps according to different loss combinations. As shown in Figure 6b, the output image with only color loss L_{col} exhibits unsatisfactory contrast and saturation. Figure 6c shows better result by considering the grid frequency loss L_{freq}. It gives enhanced results, especially for local contrast in hair, shirts, and facial structures. Figure 6d further enhance the image by using the reconstruction loss L_{rec}. However, it slightly distorts the red saturation, resulting in excessive red tones on the skin. Finally, adding the entropy



(a) Input (b) L_{col} (c) $L_{\text{col}} + L_{\text{freq}}$ (d) $L_{\text{col}} + L_{\text{freq}} + L_{\text{rec}}$ (e) Ours (f) GT

FIGURE 6. Results on the Adobe5K [4] with different loss functions. The error map is located at the bottom left of each image, with brighter pixels indicating higher errors.

TABLE 6. Results on the Adobe5K [4] with different encoder structures. The best results are highlighted in bold.

Method	PSNR ↑	SSIM ↑	LPIPS ↓
Ours	**25.60**	**0.932**	**0.049**
- w/o first branch	25.28	0.924	0.052
- w/o second branch	25.24	0.924	0.055

loss L_{ent}, Figure 6e makes the overall color tone more balanced and gives the best output. From this, we can make a conclusion that the entropy loss effectively promotes the diversity of the representative features and leads to more favorable enhanced outputs.

2) ENCODER DESIGN

Next, we study the effectiveness of the encoder structure. To this end, we detach the first or second branch from the encoder module, respectively. In Table 6, the second branch improves the performance in all three metrics. This means that considering neighboring pixels helps generate better image features for the RCT++ process. This is because it allows the RCT++ to enhance the same input colors to different transformed colors. It also helps mitigate the negative effects of noise in the input color. Note that the improvement is much more significant in SSIM (5.0%) and LPIPS (10.1%) than in PSNR (1.4%), which are more sensitive metrics to noise levels. In Table 6, the best result is obtained when we use the output of both branches.

V. CONCLUSION

We presented a novel algorithm, called the improved representative color transform (RCT++), for efficient image enhancement. The algorithm predicts image adaptive representative colors and their features, and their transformed colors. It then interpolates output colors for all pixels based on the similarities between representative features and image features. Compared to our conference version paper [22], this work has distinct improvements in that we clarified the role of representative colors and diversified the representative features by introducing the reconstruction and entropy losses, respectively. Also, we developed a fast and light enhancement network for efficient processing. We validated the effectiveness and efficiency of our method through extensive experiments on three different image enhancement datasets [4], [5], [6]. Notably, our method outperforms the existing methods in efficient image enhancement with comparable memory and computation costs.

REFERENCES

[1] S. Ren, K. He, R. Girshick, and J. Sun, ‘‘Faster R-CNN: Towards real-time object detection with region proposal networks,’’ in *Proc. Adv. Neural Inf. Process. Syst.*, 2015.

[2] J. Long, E. Shelhamer, and T. Darrell, ‘‘Fully convolutional networks for semantic segmentation,’’ in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3431–3440.

[3] K. He, G. Gkioxari, P. Dollár, and R. Girshick, ‘‘Mask R-CNN,’’ in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2980–2988.

[4] V. Bychkovsky, S. Paris, E. Chan, and F. Durand, ‘‘Learning photographic global tonal adjustment with a database of input / output image pairs,’’ in *Proc. CVPR*, Jun. 2011, pp. 97–104.

[5] C. Wei, W. Wang, W. Yang, and J. Liu, ‘‘Deep Retinex decomposition for low-light enhancement,’’ in *Proc. Brit. Mach. Vis. Conf.*, 2018.

[6] 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, 2020.

[7] Y. LeCun, Y. Bengio, and G. Hinton, ‘‘Deep learning,’’ *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[8] Z. Yan, H. Zhang, B. Wang, S. Paris, and Y. Yu, ‘‘Automatic photo adjustment using deep neural networks,’’ *ACM Trans. Graph.*, vol. 35, no. 2, pp. 1–15, May 2016.

[9] S. Nam and S. J. Kim, ‘‘Modelling the scene dependent imaging in cameras with a deep neural network,’’ in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 1726–1734.

[10] K. G. Lore, A. Akintayo, and S. Sarkar, ‘‘LLNet: A deep autoencoder approach to natural low-light image enhancement,’’ *Pattern Recognit.*, vol. 61, pp. 650–662, Jan. 2017.

[11] X. Yang, K. Xu, Y. Song, Q. Zhang, X. Wei, and R. W. H. Lau, ‘‘Image correction via deep reciprocating HDR transformation,’’ in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1798–1807.

[12] Y. Zhang, J. Zhang, and X. Guo, ‘‘Kindling the darkness: A practical low-light image enhancer,’’ in *Proc. 27th ACM Int. Conf. Multimedia (ACM MM)*, Oct. 2019, pp. 1632–1640.

[13] Y. Zhang, X. Guo, J. Ma, W. Liu, and J. Zhang, ‘‘Beyond brightening low-light images,’’ *Int. J. Comput. Vis.*, vol. 129, no. 4, pp. 1013–1037, Apr. 2021.

[14] Z. Tu, H. Talebi, H. Zhang, F. Yang, P. Milanfar, A. Bovik, and Y. Li, ‘‘MAXIM: Multi-axis MLP for image processing,’’ in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 5759–5770.

[15] Y. Cai, H. Bian, J. Lin, H. Wang, R. Timofte, and Y. Zhang, ‘‘Retinexformer: One-stage retinex-based transformer for low-light image enhancement,’’ in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2023, pp. 12504–12513.

[16] Y. Deng, C. C. Loy, and X. Tang, ‘‘Aesthetic-driven image enhancement by adversarial learning,’’ in *Proc. ACM Int. Conf. Multimedia*, 2018, pp. 870–878.

[17] H.-U. Kim, Y. J. Koh, and C.-S. Kim, "Global and local enhancement networks for paired and unpaired image enhancement," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 339–354.

[18] H. Zeng, J. Cai, L. Li, Z. Cao, and L. Zhang, "Learning image-adaptive 3D lookup tables for high performance photo enhancement in real-time," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 4, pp. 2058–2073, Apr. 2022.

[19] C. Yang, M. Jin, Y. Xu, R. Zhang, Y. Chen, and H. Liu, "SepLUT: Separable image-adaptive lookup tables for real-time image enhancement," in *Proc. Eur. Conf. Comput. Vis.*, 2022, pp. 201–217.

[20] T. Wang, Y. Li, J. Peng, Y. Ma, X. Wang, F. Song, and Y. Yan, "Real-time image enhancer via learnable spatial-aware 3D lookup tables," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 2451–2460.

[21] C. Yang, M. Jin, X. Jia, Y. Xu, and Y. Chen, "AdaInt: Learning adaptive intervals for 3D lookup tables on real-time image enhancement," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 17501–17510.

[22] H. Kim, S.-M. Choi, C.-S. Kim, and Y. J. Koh, "Representative color transform for image enhancement," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 4439–4448.

[23] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1800–1807.

[24] K. Dale, M. K. Johnson, K. Sunkavalli, W. Matusik, and H. Pfister, "Image restoration using online photo collections," in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, Sep. 2009, pp. 2217–2224.

[25] B. Wang, Y. Yu, and Y.-Q. Xu, "Example-based image color and tone style enhancement," *ACM Trans. Graph.*, vol. 30, no. 4, pp. 1–12, Jul. 2011.

[26] S. J. Hwang, A. Kapoor, and S. B. Kang, "Context-based automatic local image enhancement," in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 569–582.

[27] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 1999, pp. 1150–1157.

[28] Y.-S. Chen, Y.-C. Wang, M.-H. Kao, and Y.-Y. Chuang, "Deep photo enhancer: Unpaired learning for image enhancement from photographs with GANs," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6306–6314.

[29] R. Wang, Q. Zhang, C.-W. Fu, X. Shen, W.-S. Zheng, and J. Jia, "Underexposed photo enhancement using deep illumination estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 6842–6850.

[30] K. Xu, X. Yang, B. Yin, and R. W. H. Lau, "Learning to restore low-light images via decomposition-and-enhancement," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 2278–2287.

[31] H.-U. Kim, Y. J. Koh, and C.-S. Kim, "PieNet: Personalized image enhancement network," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Aug. 2020, pp. 374–390.

[32] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. 18th Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.*, vol. 9351. Cham, Switzerland: Springer, 2015, pp. 234–241.

[33] E. Land and J. L. McCann, "Retinex theory," *J. Opt. Soc. Amer.*, vol. 61, no. 1, pp. 1–11, 1971.

[34] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017.

[35] M. Gharbi, J. Chen, J. T. Barron, S. W. Hasinoff, and F. Durand, "Deep bilateral learning for real-time image enhancement," *ACM Trans. Graph.*, vol. 36, no. 4, pp. 1–12, Aug. 2017.

[36] L. Ma, T. Ma, R. Liu, X. Fan, and Z. Luo, "Toward fast, flexible, and robust low-light image enhancement," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 5627–5636.

[37] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[38] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 448–456.

[39] S. Elfwing, E. Uchibe, and K. Doya, "Sigmoid-weighted linear units for neural network function approximation in reinforcement learning," *Neural Netw.*, vol. 107, pp. 3–11, Nov. 2018.

[40] J. Lei Ba, J. Ryan Kiros, and G. E. Hinton, "Layer normalization," 2016, *arXiv:1607.06450*.

[41] L. Jiang, B. Dai, W. Wu, and C. C. Loy, "Focal frequency loss for image reconstruction and synthesis," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 13899–13909.

[42] S.-J. Cho, S.-W. Ji, J.-P. Hong, S.-W. Jung, and S.-J. Ko, "Rethinking coarse-to-fine approach in single image deblurring," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 4621–4630.

[43] 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/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 586–595.

[44] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *Proc. Int. Conf. Learn. Represent.*, 2018.

[45] L. Zhao, S.-P. Lu, T. Chen, Z. Yang, and A. Shamir, "Deep symmetric network for underexposed image enhancement with recurrent attentional learning," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 12055–12064.

[46] J. Huang, Y. Liu, F. Zhao, K. Yan, J. Zhang, Y. Huang, M. Zhou, and Z. Xiong, "Deep Fourier-based exposure correction network with spatial-frequency interaction," in *Proc. Eur. Conf. Comput. Vis.*, 2022.

[47] A. Sharma and R. T. Tan, "Nighttime visibility enhancement by increasing the dynamic range and suppression of light effects," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 11972–11981.

[48] W. Yang, S. Wang, Y. Fang, Y. Wang, and J. Liu, "From fidelity to perceptual quality: A semi-supervised approach for low-light image enhancement," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 3060–3069.

[49] C. Guo, C. Li, J. Guo, C. C. Loy, J. Hou, S. Kwong, and R. Cong, "Zero-reference deep curve estimation for low-light image enhancement," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 1777–1786.

[50] R. Liu, L. Ma, J. Zhang, X. Fan, and Z. Luo, "Retinex-inspired unrolling with cooperative prior architecture search for low-light image enhancement," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 10556–10565.

[51] W. Wu, J. Weng, P. Zhang, X. Wang, W. Yang, and J. Jiang, "URetinex-Net: Retinex-based deep unfolding network for low-light image enhancement," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 5891–5900.

[52] M. J. Islam, Y. Xia, and J. Sattar, "Fast underwater image enhancement for improved visual perception," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 3227–3234, Apr. 2020.

[53] 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.

[54] Z. Fu, W. Wang, Y. Huang, X. Ding, and K.-K. Ma, "Uncertainty inspired underwater image enhancement," in *Proc. Eur. Conf. Comput. Vis.*, 2022.

YEJI JEON (Student Member, IEEE) received the B.S. degree in electrical engineering from Seoul National University of Science and Technology, in 2023, where she is currently pursuing the M.S. degree in applied artificial intelligence. Her research interests include computer vision and machine learning, especially in the problems of image enhancement, and open vocabulary segmentation problems.

HANUL KIM (Member, IEEE) received the B.S. and Ph.D. degrees in electrical engineering from Korea University, Seoul, South Korea, in 2014 and 2020, respectively. From 2020 to 2021, he was a Senior Engineer with the Qualcomm AI Research. In July 2021, he joined with the Department of Applied Artificial Intelligence, Seoul National University of Science and Technology, as an Assistant Professor. His research interests include computer vision and machine learning, especially in the problems of low-level vision, autonomous driving, and vision-language models.