Representative Color Transform for Image Enhancement

Hanul Kim\textsuperscript{1}, Su-Min Choi\textsuperscript{2}, Chang-Su Kim\textsuperscript{3}, Yeong Jun Koh\textsuperscript{2*}

\textsuperscript{1}Seoul National University of Science and Technology
\textsuperscript{2}Chungnam National University
\textsuperscript{3}Korea University
hukim@seoultech.ac.kr, smchoi@cnu.ac.kr, changsukim@korea.ac.kr, yjkoh@cnu.ac.kr,

Abstract

Recently, the encoder-decoder and intensity transformation approaches lead to impressive progress in image enhancement. However, the encoder-decoder often loses details in input images during down-sampling and up-sampling processes. Also, the intensity transformation has a limited capacity to cover color transformation between low-quality and high-quality images. In this paper, we propose a novel approach, called representative color transform (RCT), to tackle these issues in existing methods. RCT determines different representative colors specialized in input images and estimates transformed colors for the representative colors. It then determines enhanced colors using these transformed colors based on the similarity between input and representative colors. Extensive experiments demonstrate that the proposed algorithm outperforms recent state-of-the-art algorithms on various image enhancement problems.

1. Introduction

Nowadays, more and more people take photographs to record and to share their valuable moments. Unfortunately, their photographs often have low dynamic ranges or distorted color tones due to inadequate lighting conditions. Therefore, image enhancement becomes popular to improve the visual aesthetics of these photos. For image enhancement, many attempts have been proposed, and considerable progress has been made.

In particular, some studies [6, 50, 52, 22] based on the encoder-decoder architecture [38] in Figure 1a provide promising results by learning a robust non-linear mapping from large amounts of paired data composed of low-quality and high-quality images. In these models, the encoder extracts features from the input image to exploit the high-level context information for image enhancement. The decoder conveys the high-level information to low-level pixel values while recovering the spatial information. Although these methods have led to the performance improvement, they have some limitations. First, details of the input image are not preserved in the up-sampling process of the decoder, even though they employ skip-connections. Second, these approaches train networks with fixed input size, which makes it difficult to enhance images of arbitrary spatial resolutions in the inference phase.

To overcome these issues, some methods [7, 21, 34, 16, 25, 13] estimate transformation functions to enhance images globally as in Figure 1b. Since these global enhancement methods do not require the down-sampling and up-sampling processes for image enhancement, images can be enhanced while preserving details. However, the existing global methods rely on intensity transformation functions on specific color space e.g. RGB [21, 13] or CIELab [7], pre-defined lookup tables [54], and pre-defined enhancement operations [34, 16, 25]. Also, they perform channel-wise color transformation and thus fail to consider all channels simultaneously. These pre-defined models have the limited capacity to cover color transformation between low-quality and high-quality images.

In this paper, we propose a novel enhancement approach, called representative color transform (RCT), which effectively achieves a large capacity for color transformation. First, we encode an input image to extract the high-level context information for image enhancement. Using the high-level context, we determine representative colors for the input image and estimate transformed colors for the representative colors, as in Figure 1c. Then, we compute the similarity between the input image and the representative colors in an embedding space. Finally, we develop a representative color transform to obtain the enhanced image by combining the similarity and the representative color transformation. Based on the proposed RCT, we propose a representative color transform network (RCTNet), which con-
Figure 1: Outlines of image enhancement approaches: (a) encoder-decoder, (b) intensity transformation, and (c) representative color transform models.

sists of encoder, feature fusion, global RCT, and local RCT modules. The proposed RCTNet predicts different representative colors specialized in input images as in Figure 1c and enlarges the capacity for color transformation by combining several representative color transformations.

Experimental results demonstrate that the proposed RCTNet outperforms recent state-of-the-art algorithms on the MIT-Adobe 5K dataset [3]. Also, we validate the scalability of the proposed RCT on specific image enhancement problems: low-light image enhancement [49] and underwater image enhancement [28, 19].

The main contributions of this paper are three folds:

- The representative color transformation to enlarge the capacity for color transformation is developed for image enhancement.
- Development of RCTNet composed of encoder, feature fusion, global RCT, and local RCT modules.
- We demonstrate excellent scalability of RCTNet for various image enhancement problems.

2. Related Work

Early studies on image enhancement improve the global contrast of an input image. For instance, power-law (gamma) and logarithmic transformation [12], which map input pixel values to output pixel values using pre-defined transformations, are well-known enhancement methods. Histogram equalization [12] improves the limited dynamic range of an image by modifying its histogram. Many attempts [23, 48, 41, 2, 26] have been developed based on these approaches to enhance the visual quality.

Recent image enhancement methods mainly focus on learning mapping functions between low-quality and high-quality images based on data-driven approaches. Bychkovsky et al. [3] provided the MIT-Adobe 5K dataset that includes 5,000 input images, where 5 different photographers manually enhance each image. This dataset is widely adopted to train image enhancement models based on deep learning. Yan et al. [51] proposed the first deep learning model for image enhancement, where the network predicts a pixel-wise color mapping from hand-crafted feature descriptors. Lore et al. [32] employed a stacked sparse denoising autoencoder to enhance a low-light image. However, these methods [3, 32] employed neural networks with small receptive fields. As a result, their models may be insufficient to exploit high-level contexts for image enhancement.

The encoder-decoder structure [38] in Figure 1a has drawn much attention to image enhancement. The encoder incrementally increases the size of receptive fields by reducing the input’s resolution to extract a deep feature containing the useful high-level information. From the deep feature, the decoder recovers the original resolution while enhancing images. Based on the encoder-decoder approach, Chen et al. [6] introduced the U-Net structure, which yields a residual image to enhance the input image. Gharbi et al. [11] predicted affine coefficients for each pixel in a low-resolution image and developed the bilateral interpolation method that effectively restores the image’s original resolution. Wang et al. [46] decomposed an input image into the reflectance and illumination and estimated the illumination to enhance the input image. Xu et al. [50] developed the frequency-based decomposition for enhancement of low-light images. Yang et al. [53] constructed two encoder-decoder structures for image correction of under-
exposed inputs. Yang et al. [52] proposed the deep recursive band network and trained it in the semi-supervised framework. Kim et al. [22] designed the encoder-decoder network to produce a personalized image according to the user’s preference. However, these encoder-decoder architectures [6, 11, 46, 50, 52, 22] have the problem that details of input images are not preserved in down-sampling and up-sampling processes. Also, they train networks with fixed input sizes, which makes it difficult to enhance images of arbitrary spatial resolutions.

As in Figure 1b, some methods [7, 13, 21, 34, 16, 25, 54] perform global enhancement through transformation functions or pre-defined enhancement operations. Deng et al. [7] estimated piece-wise intensity transform functions on the CIELab color space. Guo et al. [13] developed pixel-wise and high-order curves for dynamic range adjustment of an input image. Kim et al. [21] proposed the non-monotonic and channel-wise intensity transformation for both paired and unpaired image enhancement. In [34, 16, 25], neural networks are trained to select the best operation among some pre-defined enhancement operations based on deep reinforcement learning. Zeng et al. [54] learned image-adaptive 3-dimension lookup tables for global image enhancement. These global-based methods [7, 13, 21, 34, 16, 25, 54] can enhance low-quality images without image resize unlike the encoder-decoder models. However, they have the limitation in that transformation functions on the pre-defined color space, pre-defined lookup tables, or pre-defined operations may not be sufficient to estimate highly non-linear mapping between low-quality and high-quality images. In contrast, the proposed method estimates adaptive representative colors according to the input image, and predicts color transformation for each representative color based on the attention mechanism.

Finally, we review palette-based image enhancement methods [5, 42], which interpolate colors based on the palette colors. Chang et al. [5] set initial palettes using the K-mean clustering, and then users manually change palette colors for image enhancement. Tan et al. [42] determined initial palettes using vertices of the convex hull wrapped around input colors. The proposed algorithm is related to them in that representative colors are similar to initial palettes. However, the proposed algorithm automatically determines representative colors and their transformed colors, while they require user interaction to update palettes.

3. Method

In this section, we propose the representative color transform (RCT), which is a simple and effective approach to improve the visual quality of input images. Based on RCT, we develop the representative color transform network (RCTNet) that contains global and local enhancement modules, which are trained in an end-to-end manner. Figure 2 summarizes the proposed RCTNet architecture.

3.1. Representative Color Transform

Let \( X \in \mathbb{R}^{H \times W \times 3} \) denote an input low-quality image, where \( H \times W \) is the spatial resolution of the image. We encode it to a feature representation \( Z \) to embed the high-level context for image enhancement. Given \( Z \), we extract features and transformed colors for \( N \) representative colors. Let \( R \) denote the set of representative features, which is given by

\[
R = [r_1, r_2, \ldots, r_N] \in \mathbb{R}^{C \times N}
\]  

where \( r_i \) denotes a feature vector of \( i \)th representative color and \( C \) is a feature dimension. Also, the set of transformed colors is defined as

\[
T = [t_1, t_2, \ldots, t_N] \in \mathbb{R}^{3 \times N}
\]  

where \( t_i \) denotes the transformed RGB values of \( i \)th representative color. In other words, the \( i \)th representative color should be transformed to \( t_i \) for image enhancement.

Note that \( T \) contains the transformation for only \( N \) representative colors, not all colors. Thus, we should map each pixel color in the input image to the representative colors. For this purpose, we compute the similarity between the input color and the representative features in the embedding space and perform the color transformation based on the similarity. We extract a image feature \( F \in \mathbb{R}^{H \times W \times C} \) from the input \( X \) using a stack of convolution layers. Then, we perform the matrix multiplication to obtain the attention matrix \( A \) via the scaled-dot product [45].

\[
A = \text{softmax}(\frac{F \cdot R}{\sqrt{C}}) \in \mathbb{R}^{HW \times N}
\]  

where \( F \in \mathbb{R}^{HW \times C} \) a is reshaped tensor of \( F \). The element \( a_{ij} \) of \( A \) is the attention weight that represents the similarity between \( i \)th pixel in the input image and the \( j \)th representative color. Thus, the attention matrix determines all similarities on every transformed color in \( T \) for each pixel in the input image.

Let us consider enhancement of \( i \)th pixel in the input image. Then, enhanced RGB values of \( i \)th pixel is determined by the combination of \( N \) transformed colors for the representative colors with attention weights as \( \sum_{j=1}^{N} a_{ij} t_j \). To this end, the enhanced image \( Y \) is obtained by

\[
Y = AT^T.
\]
Feature Fusion: Feature maps provide different context information according to resolutions. In general, coarse-scale feature maps contain the global context due to large receptive fields. In contrast, fine-scale feature maps preserve the detailed local context. Since both global and local contexts are essential for image enhancement, we aggregate multi-scale feature maps through the feature fusion module.

To construct the feature fusion module, we employ the bidirectional cross-scale connections [43]. In the feature fusion module in Figure 2, nodes that have a single input represent the ‘conv-bn-swish’ block. On the other hand, nodes, which have multiple inputs, contain a feature fusion layer before the ‘conv-bn-swish’ block to mix multiple inputs effectively. When M inputs are provided to the feature fusion layer, the output of the feature fusion layer is defined as

\[ O = \sum_{i=1}^{M} \frac{w_i}{\epsilon + \sum_j w_j} \cdot I_i \]  

where \( w_i \) is a non-negative learnable weight for the \( i \)th input \( I_i \), and \( \epsilon = 0.0001 \). All nodes have 128 convolution filters of size 3 \times 3 except nodes at the coarsest-scale (red nodes in Figure 2). Since the spatial resolution of the coarsest feature map is 1 \times 1, these nodes have convolution filters of size 1 \times 1.

3.2. Representative Color Transform Network

As in Figure 2, the proposed RCTNet consists of four modules: encoder, feature fusion, global RCT, and local RCT. Given an input low-quality image \( X \), RCTNet produces a high-quality image:

\[ \hat{Y} = \alpha Y_G + \beta Y_L \]  

where \( Y_G \) and \( Y_L \) are enhanced images obtained from the global and local RCT modules, respectively. Also, \( \alpha \) and \( \beta \) are non-negative learnable weights to combine two images effectively. Let us describe each module subsequently.

Encoder: Encoder is a convolutional neural network to encode an input image to extract the high-level context information for image enhancement. Table 1 describes the detailed architecture of the encoder. The input image is resized to 256 \times 256 and fed into the encoder, composed of a stack of 6 ‘conv-bn-swish’ blocks. Each ‘conv-bn-swish’ block contains a convolution, a batch normalization [17], and a swish activation [37] layers. All convolution layers except the last block have 3 \times 3 filters. Unlike the others, the last block uses a convolution layer with 1 \times 1 filter and employs a global average pooling layer to extract a global feature vector. In the encoder, we extract multi-scale feature maps from the last four blocks to combine them in the feature fusion module.

Table 1: Specification of the encoder architecture.

<table>
<thead>
<tr>
<th>Stage</th>
<th>Operations</th>
<th>Outputs</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>resize</td>
<td>256 x 256 x 3</td>
</tr>
<tr>
<td>1</td>
<td>conv-bn-swish, k3x3</td>
<td>128 x 128 x 16</td>
</tr>
<tr>
<td>2</td>
<td>conv-bn-swish, k3x3</td>
<td>64 x 64 x 32</td>
</tr>
<tr>
<td>3</td>
<td>conv-bn-swish, k3x3</td>
<td>32 x 32 x 64</td>
</tr>
<tr>
<td>4</td>
<td>conv-bn-swish, k3x3</td>
<td>16 x 16 x 128</td>
</tr>
<tr>
<td>5</td>
<td>conv-bn-swish, k3x3</td>
<td>8 x 8 x 256</td>
</tr>
<tr>
<td>6</td>
<td>conv-bn-swish-pool, k1x1</td>
<td>1 x 1 x 1024</td>
</tr>
</tbody>
</table>

Global RCT: Let \( Z_G \in \mathbb{R}^{C'} \) denote an output feature at the coarsest-scale in the feature fusion module, where \( C' \) is set to 128. By analyzing the feature vector \( Z_G \), which includes the global context for image enhancement, the global RCT module determines representative features \( R_G \in \mathbb{R}^{C \times N_G} \) and transformed colors \( T_G \in \mathbb{R}^{3 \times N_G} \) through two different ‘conv-bn-swish-conv’ blocks. One ‘conv-bn-swish-conv’ block yields a vector with \( C N_G \) dimension, while another block produces a vector with \( 3 N_G \) dimension. These output vectors are reshaped to the 2D structures, \( R_G \) and \( T_G \), respectively. In this work, we set \( C \) and \( N_G \) to 16 and 64, respectively. Also, the input image is transformed to the image feature \( F \in \mathbb{R}^{H \times W \times C} \) through one ‘conv-bn-swish-conv’ block and \( F \) is reshaped to \( F_r \in \mathbb{R}^{H W \times C} \). Finally, the global enhanced image \( Y_G \) is obtained by applying \( R_G \), \( T_G \), and \( F_r \) to (3) and (4).

Local RCT: The local RCT module determines region-wise representative colors to consider local region characteristics for image enhancement. For this purpose, the local RCT module takes a feature map \( Z_L \in \mathbb{R}^{32 \times 32 \times C'} \), extracted from the finest-scale in the feature fusion module, whose the spatial resolution is 32 \times 32. Then, given \( Z_L \), the sets of representative features and transformed colors are generated for each spatial position. Specifically, \( Z_L \) are fed into two different ‘conv-bn-swish-conv’ blocks, where the first convolution layers have 128 convolution filters of size 3 \times 3 and each second convolution layer has \( C N_L \) and \( 3 N_L \) filters of size 3 \times 3, respectively. Then, the local RCT produces the representative feature sets \( R_L \in \mathbb{R}^{32 \times 32 \times C \times N_l} \) and the transformed color sets \( T_L \in \mathbb{R}^{32 \times 32 \times 3 \times N_L} \), where \( N_L \) is 16. To this end, the set of representative features \( R_L(u, v) \) and the set of transformed colors \( T_L(u, v) \) are obtained for each spatial position \( (u, v) \).

Given \( R_L \) and \( T_L \), the local RCT module assigns different sets of representative features and transformed colors to
an input color according to its pixel coordinates. First, we set a $31 \times 31$ uniform mesh grid on the input image, and thus there are $32 \times 32$ corner points. We regard representative features and transformed colors at corner points on spatial position $(u, v)$ as $R_L(u, v)$ and $T_L(u, v)$, respectively. Then, grid-wise RCT is performed in the local RCT module. Specifically, the $k$th grid $B_k$ has four corner points, which means that $B_k$ are related to the four sets of representative features and transformed colors. We then determine the representative features $R_k$ for the grid $B_k$ by concatenating the four sets of representative features at corner points. Also, the transformed colors $T_k$ for $B_k$ is obtained similarly. From the image feature $F$, a grid feature $F_k$ is extracted by cropping on the grid region. Finally, given $R_k$, $T_k$, and $F_k$, enhanced colors for $B_k$ is computed by (3) and (4). The local RCT module repeat this process for all grids to yield the local enhanced image $Y_L$. Figure 3 shows an example of how the local RCT is performed for the grid $B_k$. For the simplicity, we set a $5 \times 5$ mesh grid in this example.

Though the encoder requires the fixed-size input to extract multi-scale feature maps, both global and local RCT modules enhance input images without any image resize by extracting the image feature $F$ without any down-sampling.

3.3. Loss Functions

Let us consider a pair $(X, Y)$, where $X$ and $Y$ are an input low-quality image and its high-quality image, respectively. Given $X$, the proposed RCTNet produces an enhanced image $\tilde{Y}$. We then define the loss function between $\tilde{Y}$ and $Y$ by

$$L = \|\tilde{Y} - Y\|_1 + \lambda \sum_{k=2,4,6} \|\phi^k(\tilde{Y}) - \phi^k(Y)\|_1.$$  

Here, the first term is the mean absolute error between the predicted and ground-truth enhanced images. And the second term penalizes the difference between them in the well-defined embedding space. Specifically, the embedding function $\phi^k(\cdot)$ is the output of $k$th layer in VGG-16 [40], which is pre-trained on the ImageNet [39] dataset. The hyper parameter $\lambda$ is fixed to 0.04 to balance two terms.

4. Experiments

In this section, we verify the effectiveness of the proposed method through extensive experiments:

- We compare the proposed algorithm with recent state-of-the-arts in standard image enhancement.
- We evaluate the scalability of the proposed RCTNet on specific image enhancement problems: low-light image enhancement and underwater image enhancement.
- We analyze parameters and components of RCT-Net through ablation studies on the MIT-Adobe 5K dataset [3].
Table 2: Quantitative comparison on the MIT-Adobe 5K dataset [3]. The best results are boldfaced and the second best ones are underlined.

<table>
<thead>
<tr>
<th>Method</th>
<th>PSNR</th>
<th>SSIM</th>
</tr>
</thead>
<tbody>
<tr>
<td>HDRNet [11]</td>
<td>23.44</td>
<td>0.882</td>
</tr>
<tr>
<td>DPE [6]</td>
<td>23.34</td>
<td>0.873</td>
</tr>
<tr>
<td>DUPE [46]</td>
<td>23.61</td>
<td>0.887</td>
</tr>
<tr>
<td>DLPF [33]</td>
<td>24.48</td>
<td>0.887</td>
</tr>
<tr>
<td>3D LUT [54]</td>
<td>25.21</td>
<td>0.922</td>
</tr>
<tr>
<td>GEN-LEN [21]</td>
<td>25.88</td>
<td></td>
</tr>
<tr>
<td>RCTNet</td>
<td>26.02</td>
<td>0.915</td>
</tr>
<tr>
<td>RCTNet + BF</td>
<td>26.07</td>
<td>0.923</td>
</tr>
</tbody>
</table>

We adopt PSNR and SSIM metrics for the quantitative evaluation in all experiments, which measure color and structural similarity between predicted and ground-truth images. More results are available in the supplementary material.

4.1. Datasets

MIT-Adobe 5K: The MIT-Adobe 5K dataset [3] consists of 5,000 images, each of which was manually enhanced by five different photographers (A/B/C/D/E). There are five sets (one set per photographer) consisting of 5,000 pairs of input and retouched images. Among these sets, we only use images retouched by photographer C as done in most existing image enhancement methods [6, 46, 21]. And we decompose it to 4,500 and 500 images for training and test set, respectively.

Low Light (LoL): LoL [49] is a dataset for low-light image enhancement. The LoL dataset contains 500 pairs of low-light and normal-light images in which 500 pairs are separated into 485 training images and 15 testing images. We use the training images for training RCTNet and the test images for experiments.

Enhancing Underwater Visual Perception (EUVP): The EUVP dataset [19] provides subsets of the paired and unpaired collections for underwater images. The paired dataset separates pairs of low-quality and high-quality images into 11435, 570, and 515 pairs for the training, validation, and test sets. The pairs in the training and test set are used for the training and evaluation, respectively.

Underwater Image Enhancement Benchmark (UIEB): The UIEB dataset [28] includes 890 pairs of underwater image and its enhanced image. These pairs are divided into 800 and 90 for training and test, respectively. We train RCTNet using 800 training images and evaluate the proposed algorithm on the test set.

4.2. Implementation Details

We train the proposed model for 100, 500, 500, and 100 epochs with batch size of 8 for the MIT-Adobe-5K, LoL, EUVP, and UIEB datasets, respectively. We use Adam optimizer [24] to minimize the loss function, with an initial learning rate of $5 \times 10^{-4}$ and a weight decay of $1 \times 10^{-5}$. We decrease learning rate according to the cosine learning rate scheduling. Following the literature [21], we randomly crop image and then rotate them by multiples of 90 degrees for data augmentation. We fix the hyperparameter $\lambda$ to 0.04.

4.3. Comparison with state-of-the-arts

MIT-Adobe 5K: We compare the performance of the proposed method with recent state-of-the-art methods [11, 6, 46, 33, 54, 21]. For comparison, we obtain the results of existing algorithms using their published source codes and default settings. Table 2 lists the PSNR and SSIM performances on the MIT-Adobe 5K dataset. We resize each test image to have 512 pixels in the long side of each test image as done in existing algorithms [33, 21] for the comparison. In Table 2, the proposed RCTNet achieves the best on PSNR, which indicates that the proposed RCTNet is effective for color enhancement. In contrast, RCTNet yields the second best performance on SSIM, since it does not perform the spatial filtering that suppresses noises. To address denoising problem, we can employ the simple filtering method, such as the bilateral filter (BF) [44], as the post-processing. As in Table 2, the bilateral filter improves the SSIM score to 0.923.

Low-light Image Enhancement: Next, we evaluate the proposed RCTNet on the low-light image enhancement problem. Table 3 compares the proposed RCTNet with state-of-the-art low-light image enhancement algorithms [47, 14, 9, 31, 4, 49, 55, 52, 13, 20] on the LoL dataset [49]. ZeroDCE [13] and EnlightenGAN [20] provides relatively lower performance because they train their network with unpaired images. The proposed RCTNet achieves the best PSNR score by enhancing input colors ef-
Table 4: Quantitative comparison on the UIEB dataset [28]. The best results are boldfaced and the second best ones are underlined.

<table>
<thead>
<tr>
<th>Method</th>
<th>PSNR</th>
<th>SSIM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fusion [1]</td>
<td>17.60</td>
<td>0.772</td>
</tr>
<tr>
<td>Retinex [10]</td>
<td>17.02</td>
<td>0.607</td>
</tr>
<tr>
<td>GDCP [35]</td>
<td>12.09</td>
<td>0.512</td>
</tr>
<tr>
<td>Histogram [30]</td>
<td>15.82</td>
<td>0.539</td>
</tr>
<tr>
<td>Blurriness [36]</td>
<td>15.32</td>
<td>0.603</td>
</tr>
<tr>
<td>Water CycleGAN [29]</td>
<td>15.75</td>
<td>0.521</td>
</tr>
<tr>
<td>Dense GAN [15]</td>
<td>17.28</td>
<td>0.443</td>
</tr>
<tr>
<td>WaterNet [28]</td>
<td>19.11</td>
<td>0.797</td>
</tr>
<tr>
<td>Ucolor [27]</td>
<td>20.63</td>
<td>0.770</td>
</tr>
<tr>
<td>RCTNet</td>
<td>22.45</td>
<td>0.891</td>
</tr>
</tbody>
</table>

Table 5: Quantitative comparison on the EUVP dataset [19]. The best results are boldfaced and the second best ones are underlined.

<table>
<thead>
<tr>
<th>Method</th>
<th>PSNR</th>
<th>SSIM</th>
</tr>
</thead>
<tbody>
<tr>
<td>U-GAN [8]</td>
<td>23.49</td>
<td>0.842</td>
</tr>
<tr>
<td>Funie-GAN [19]</td>
<td>23.40</td>
<td>0.827</td>
</tr>
<tr>
<td>Deep SESR [18]</td>
<td>24.21</td>
<td>0.840</td>
</tr>
<tr>
<td>RCTNet</td>
<td>26.43</td>
<td>0.891</td>
</tr>
</tbody>
</table>

Underwater Image Enhancement: Finally, we assess the performance of the proposed RCTNet on the underwater image enhancement problem. Tables 4 and 5 compare the proposed algorithm with the existing underwater image enhancement algorithms [1, 10, 35, 30, 36, 29, 8, 28, 19, 15, 18, 27] on the UIEB [28] and EUVP [19] datasets, respectively. Underwater images tend to be degraded by wavelength-dependent absorption and scattering due to shooting environments. Nevertheless, RCTNet faithfully enhances underwater images and significantly outperforms the existing state-of-the-art methods in PSNR and SSIM scores on both datasets. Figures 5 qualitatively compare the proposed algorithm with the second-best methods in EUVP dataset. Remarkably, we see that RCTNet produces visually pleasing results.

4.4. Ablation Studies

Component Analysis: We analyze the efficacy of the three components of feature fusion, global RCT, and local RCT modules in RCTNet. In this test, we measure the three
performed the importance of RCTNet: 1) without the feature fusion, 2) without the local RCT, and 3) without the local RCT. Let us refer to these settings as ‘w/o global RCT,’ ‘w/o local RCT,’ and ‘w/o feature fusion.’ Table 6 summarizes the average PSNR scores of these settings on four datasets. Without the global RCT, the local RCT, or the future fusion, the PSNR scores are degraded severely. This indicates that the proposed components are essential for image enhancement.

Representative Color Transform: We verify the efficacy of the proposed RCT by replacing it with different enhancement models. First, we employ the decoder in the U-Net architecture [6], which includes 6 up-sample blocks to perform bilinear interpolation, concatenation, and convolution filtering. Second, we substitute the proposed RCT with the channel-wise intensity transform model in [21]. Third, we exclude the local RCT module from RCTNet for the fair comparison. All enhancement models include the same encoder and the feature fusion module in Figure 2.

Table 7 summarizes the PSNR scores of three image enhancement approaches. Here, ‘Decoder,’ ‘Global IT,’ and ‘Global RCT’ denote the encoder-decoder model, the global intensity transform, and the proposed global RCT, respectively. ‘Decoder’ provides the worst performance on all datasets except LoLo dataset. This indicates that approaches based on the color transformation are more effective for image enhancement than encoder-decoder structures. Also, ‘Global RCT’ outperforms ‘Global IT’ by providing larger color transformation capacities.

5. Conclusion

We proposed the novel image enhancement algorithm based on the representative color transform. The proposed RCT determines different representative colors specialized in input images and enhances input images using representative features and transformed colors. Then, with the proposed RCT, we developed RCTNet, composed of encoder, feature fusion, global RCT, and local RCT modules. The global RCT predicts representative colors for an input image, while the local RCT determines region-wise representative colors to consider local region characteristics for image enhancement. Extensive experiments demonstrated that the proposed RCTNet outperforms the recent state-of-the-art algorithms on various datasets with standard image enhancement, low-light image enhancement, and underwater image enhancement.

Acknowledgement

This work was supported by the National Research Foundation of Korea (NRF) grants funded by the Korea government (MSIT) (No. NRF-2018R1A2B3003896), (No. NRF-2019R1F1A1062907), and (No. NRF-2021R1A4A1031864).
References


