Real-time Image Enhancer via Learnable Spatial-aware 3D Lookup Tables

Tao Wang*, Yong Li*, Jingyang Peng*, Yipeng Ma, Xian Wang, Fenglong Song†, Youliang Yan‡
Huawei Noah’s Ark Lab
{wangtao10, liyong156, pengjingyang1, mayipeng, wangxian10, songfenglong, yanyouliang}@huawei.com

Abstract

Recently, deep learning-based image enhancement algorithms achieved state-of-the-art (SOTA) performance on several publicly available datasets. However, most existing methods fail to meet practical requirements either for visual perception or for computation efficiency, especially for high-resolution images. In this paper, we propose a novel real-time image enhancer via learnable spatial-aware 3-dimensional lookup tables (3D LUTs), which well considers global scenario and local spatial information. Specifically, we introduce a light weight two-head weight predictor that has two outputs. One is a 1D weight vector used for image-level scenario adaptation, the other is a 3D weight map aimed for pixel-wise category fusion. We learn the spatial-aware 3D LUTs and fuse them according to the aforementioned weights in an end-to-end manner. The fused LUT is then used to transform the source image into the target tone in an efficient way. Extensive results show that our model outperforms SOTA image enhancement methods on public datasets both subjectively and objectively, and that our model only takes about 4ms to process a 4K resolution image on one NVIDIA V100 GPU.

1. Introduction

Recently, many deep learning-based approaches have been proposed and achieved SOTA results [9, 15, 25, 4, 20, 14, 26, 19, 28] in the field of computational imaging. However, complex network architecture and high computation overheads prevent them from real-time processing. Figure 1 shows the comparison of performance and efficiency (i.e., execution time) of several network architectures on HDR+ Burst Photography dataset [6]. Most existing methods cannot produce visually pleasant results in real time.

Considering both performance and efficiency, it is still a big challenge for image enhancement due to the diversity of capture scenarios. Recently, many hybrid methods [8, 25, 30], which combine image prior in traditional approaches and multi-level features in deep learning-based approaches, are proposed and achieve SOTA performance. [25] proposes a new image enhancement method with good image quality, high computation efficiency and low memory consumption. However, as the limitations pointed out by authors, it works simply based on pixel values, without considering local information. This may produce less satisfactory results in local areas. For example, as shown in Figure 7, local contrast is limited in some results captured in high dynamic range scenes. In addition, there are also some color distortion and artifacts as shown in Figure 8.

To solve these issues, we present a novel CNN-based image enhancement approach, where spatial information is introduced to traditional 3D lookup tables to boost its performance. Particularly, T spatial-aware 3D LUTs, each of which is a set of M basic learnable 3D LUTs, and a two-head weight predictor are trained simultaneously under a new loss function to balance well between details, colors, and visual perception. The weight predictor has two outputs. One is a 1D weight vector with global informa-
Figure 2: Overview of our proposed framework. It consists of multiple spatial-aware 3D LUTs (i.e., $T$ spatial-aware 3D LUTs, each with $M$ basic 3D LUTs selected by $M$-channel pixel-wise category information), a self-adaptive two-head weight predictor, and interpolation for spatial-aware 3D LUTs. The weight predictor takes down-sampled images as input and generates two outputs. One is a 1D weight vector used for image-level scenario adaptation, the other is a 3D weight map aimed for pixel-wise category fusion, enabling our LUT-based enhancer with image-adaptive spatial-aware ability.

2. Related Work

Existing learning-based approaches can be broadly divided into three categories, namely pixel-level, patch-level, and image-level methods.

**Pixel-level methods.** This kind of methods adopt CNN to extract features from input images of initial size and reconstruct every pixel from dense pixel-to-pixel mapping or transformation operations. This kind of approaches have made great breakthroughs and achieved SOTA performance in many image enhancement tasks [11, 22, 29, 3, 24, 16, 2]. [10] proposes a residual CNN architecture as enhancer to learn the pixel-wise translation function between low-quality cellphone images and high-quality Digital Single-Lens Reflex (DSLR) images. [3, 11, 2, 8] all employ U-Net-style structure originated from [18] for different image quality enhancement tasks. Despite their SOTA performance, these dense pixel-wise feature extraction and regeneration methods are too heavy to be used for practical applications, especially for high resolution input images [25].

**Patch-level methods.** These methods generate compressed features from a down-sampled input image. Different parts of features are then applied on different local input patches to reconstruct the enhanced image. [5] extracts local and global features as a bilateral grid in low resolution, and then applies interpolation according to the grid and a
learned feature map of full resolution. Based on the same interpolation operation, [21] learns a full resolution illumination map to retouch the input image. Wu et al. [23] introduce the guided filter proposed in [7], and they build a trainable guided filtering layer and plug it in the network for up-sampling the enhanced low resolution image. Although patch-level methods perform well both on computation and memory consumption, they still overload hardware resources, especially for ultra-high resolution images.

**Image-level methods.** These approaches have the highest computation efficiency and lowest memory consumption. They calculate global scaling factors or mapping curves from a down-sampled input image, which are then applied on the whole input image for enhancement. [25] propose image-adaptive 3D LUTs for efficient image enhancement and it takes only 1.66 ms to process a 4K image on NVIDIA Titan RTX GPU. However, it is hard to ensure robustness since spatial information is not considered, which may easily result in low local contrast or even wrong color in some local areas, as shown in Figure 7 and Figure 8.

### 3. Methodology

In this section, we present our network framework and loss functions in detail. Figure 2 illustrates fundamental modules of our network architecture, consisting of multiple spatial-aware 3D LUTs, a self-adaptive two-head weight predictor and spatial-aware trilinear interpolation.

#### 3.1. Network architecture

**Spatial-aware 3D LUTs.** 3D LUT is an effective color mapping operator, which contains two basic operations: lookup and interpolation. For simplicity of description, we do not describe the interpolation operation in 3D LUT, but simplify it to lookup only in this subsection.

Equation 1 indicates the mapping function. In RGB color domain, a classical 3D LUT is defined as a 3D cube which contains \( N^3 \) elements, where \( N \) is the number of bins in each color channel. Each element defines a pixel-to-pixel mapping \( \mu^c(i,j,k) \), where \( i,j,k = 0,1,\ldots,N-1 \), abbreviated as \( i,j,k \in \mathbb{N}_0^{N-1} \) in the following section, are elements’ coordinates within 3D LUT and \( c \) indicates one of channels. Inputs of the mapping are RGB colors \( \{I^r_{(i,j,k)}, I^g_{(i,j,k)}, I^b_{(i,j,k)}\} \), where \( i,j,k \) are indexed by the corresponding RGB value, and output is the pixel value after mapping for channel \( c \), as in Equation 1. \( O^c \) is output for 3D LUT with \( c \in \{r,g,b\} \), and \( r,g,b \) is the color value for red, green, blue channel respectively.

\[
O^c_{(i,j,k)} = \mu^c(I^r_{(i,j,k)}, I^g_{(i,j,k)}, I^b_{(i,j,k)}) \quad (1)
\]

Obviously, mapping for traditional 3D LUT depends merely on pixel values, but fails to consider spatial information. In other words, the transformation is only sensitive to pixel values, and discards pixels’ spatial information. We propose new spatial-aware 3D LUTs involving \( M \) traditional 3D LUTs, each of which represents a kind of mapping. For the final result, our method adaptively fuses multiple LUTs according to pixel-wise weight map. As shown in Equation 2, \( \phi^h,w,c \) is the entire mapping, \( \nu^c \) is a mapping for the \( m \)-th LUT and \( \alpha^h,w = \{\alpha^h,w_m | h \in \mathbb{N}_0^{H-1}, w \in \mathbb{N}_0^{W-1}, m \in \mathbb{N}_0^{M-1}\} \) is a spatial-aware pixel-wise weight map for \( M \) 3D LUTs at location \((h,w)\).

\[
O^{h,w,c}_{(i,j,k)} = \phi^{h,w,c}(I^r_{(i,j,k)}, I^g_{(i,j,k)}, I^b_{(i,j,k)}, \alpha^{h,w}) = \sum_{m=0}^{M-1} \alpha^{h,w}_m \nu^c(I^r_{(i,j,k)}, I^g_{(i,j,k)}, I^b_{(i,j,k)}, m) = \sum_{m=0}^{M-1} \alpha^{h,w}_m O^{m,c}_{(i,j,k)} \quad (2)
\]

where \( O^{h,w,c}_{(i,j,k)} \) is the final spatial-aware result and \( O^{m,c}_{(i,j,k)} \) is the mapping result of the \( m \)-th 3D LUT.

Note that pixels are adaptively classified into different categories through an end-to-end learning approach according to color, illumination, semantic and other information. This generalizes our model to different use cases and promotes its learning ability. Figure 3 visualizes our spatial-aware 3D LUTs.

We use \( V = \{\phi^{h,w,c}(i,j,k,\alpha^{h,w})\} \) to represent a set of all mappings in spatial-aware 3D LUTs. Thus, \( Y = V(X,A) \) indicates applying spatial-aware 3D LUTs on input image \( X \). \( A = \{\alpha^{h,w}_m | h \in \mathbb{N}_0^{H-1}, w \in \mathbb{N}_0^{W-1}, m \in \mathbb{N}_0^{M-1}\} \) is the pixel-wise category information, which is introduced in next part.
Self-adaptive two-head weight predictor. We propose a self-adaptive two-head weight predictor to support image-adaptive spatial-aware 3D LUTs. Upper part of Figure 2 shows its framework, which is a UNet-style backbone with two outputs. The first one is a 1D weight vector with T probabilities \( \{\omega_t | t \in \mathbb{Z}_0^{T-1}\} \), where \( T \) is scene number. These \( T \) probabilities are used for scene adaptation. We assume that the scene is a global feature, and its probability can be expressed by a single value in the probability vector. With these probabilities, a scenario-adaptive 3D LUT can be jointly leaned by \( T \) spatial-aware 3D LUTs. For an input image \( X \), the final enhancement result \( Y \) can be expressed as follows. In the following experiments, we set \( T = 3 \) according to [25].

\[
Y = \sum_{t=0}^{T-1} \omega_t * V_t(X, A) \quad (3)
\]

The second output is an \( M \)-channel 3D weight map with \( H \times W \times M \) probabilities \( A = \{ \alpha_{h,w}^m | h \in \mathbb{Z}_0^H, w \in \mathbb{Z}_0^W, m \in \mathbb{Z}_0^M \} \), as shown in Figure 2. Each channel corresponds to fusion weight for specific LUTs as shown in Figure 3. With the pixel-wise weight information, spatial feature is fused to 3D LUTs, which greatly promotes enhancement result in many aspects, e.g., local contrast and saturation. Detailed results are analyzed in Section 4.

Our weight predictor takes resized low resolution images as inputs, enabling it to process arbitrary size images in real time. Moreover, the Encoder-Decoder architecture increases the receptive field size, which is powerful in generating pixel-wise category feature.

Spatial-aware trilinear interpolation. Considering the efficiency and performance, trilinear based interpolation is used in our method to improve the smoothness of the enhanced result. For detailed derivation, please refer to the supplementary material. Owing to the spatial-aware attribute of the pixel-wise category weight map \( \alpha_{m,w}^h \), the interpolation is defined as spatial-aware trilinear interpolation.

3.2. Loss function

Our loss function consists of MSE Loss, Smooth Loss [25], Monotonicity Regularization Loss [25], Color Difference Loss and Perception Loss. MSE Loss \( (L_r) \) ensures content consistency of generated image. Smooth Loss \( (L_s) \) and Monotonicity Regularization Loss \( (L_m) \) are introduced to ensure LUTs’ smoothness and reduce artifacts.

Additionally, in order to promote enhancement quantitatively and perceptually, we introduce Color Difference Loss \( (L_c) \) and Perceptual Loss \( (L_p) \) to the optimization process.

Color Difference Loss. To measure the color distance and encourage the color in the enhanced image to match that in the corresponding learning target, we use CIE94 in LAB space as our color loss. Detailed description can be found in [13] and supplementary material.

\[
L_c = \sqrt{\Delta L^2 + \left( \frac{\Delta C}{S_C} \right)^2 + \left( \frac{\Delta H}{S_H} \right)^2 + \epsilon} \quad (4)
\]

Perception loss. LPIPS loss [27] is chosen to improve the perceptual quality of the enhanced image.

\[
L_p = \frac{1}{H \cdot W} \sum_{h=1}^{H} \sum_{w=1}^{W} \left\| \hat{y}_{h,w}^l - y_{h,w}^l \right\|^2_2 \quad (5)
\]

where \( l \) is the layer chosen to calculate lppis loss, and \( \hat{y}, y \) is the corresponding ground truth features and enhanced features on a pre-trained AlexNet.

Finally, the loss function is defined as a weighted sum of different losses with following coefficients.

\[
L = L_r + 0.0001 \cdot L_s + 10 \cdot L_m + 0.005 \cdot L_c + 0.05 \cdot L_p \quad (6)
\]

4. Experiments

Datasets. We evaluate our method on two publicly available datasets: MIT-Adobe FiveK [1] and HDR+ burst photography [6]. Since [25] achieved the SOTA performance on both dataset and also published its 480p dataset (only 480p, w/o full resolution), we directly take their released 480p dataset for performance evaluation. We also construct two new dataset for further comparison. One is full resolution MIT-Adobe FiveK dataset. The ExpertC images are used as the ground truth while the input DNG images are automatically converted to PNG images as input. We use the same filelist as [25] for training and testing. The other is 480p and full resolution HDR+ dataset. Our input images are merged DNG images (i.e., merge.dng) post-processed by python rawpy library with automatic white balance, while ground truth images are kept as the software output (i.e., final.jpg). Since most scenes are not well aligned in the original dataset, we conduct manual comparison and remove image pairs with large offset. In this way, we construct a dataset with 2041 image pairs. Finally, we randomly split image pairs in the dataset into two subsets: 1837 image pairs for training and the rest 204 pairs for testing. Since the number of the released 480p HDR+ dataset by [25] is relatively small (675 pairs), we also construct our 480p HDR+ dataset with the short side resized to 480 pixels and long side proportionally.

Evaluation metrics. We employ three common-used metrics (i.e., PSNR, SSIM and LPIPS) to quantitatively evaluate the performance of different methods. Generally speaking, a higher PSNR/SSIM and lower LPIPS means better results.
### Table 1: Ablation study for different combinations of CNN weight predictor and 3DLUTs. A spatial-aware 3D LUTs is composed with M basic 3D LUTs.

<table>
<thead>
<tr>
<th>Method(T,M)</th>
<th>Configuration</th>
<th>CNN input</th>
<th>Weight predictor</th>
<th>#3DLUTs</th>
<th>GFLOPS</th>
<th>#Params</th>
<th>PSNR ↑</th>
<th>SSIM ↑</th>
<th>LPIPS ↓</th>
</tr>
</thead>
<tbody>
<tr>
<td>3DLUT(3,0)</td>
<td>[25] 480p 1D</td>
<td>3× basic</td>
<td>19.91</td>
<td>0.6567</td>
<td>0.2455</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3DLUT(30,0)</td>
<td>[25] 480p 1D</td>
<td>30× basic</td>
<td>20.29</td>
<td>0.6614</td>
<td>0.2306</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ours(0,30)</td>
<td>ours 480p 1D</td>
<td>30× basic</td>
<td>20.38</td>
<td>0.6888</td>
<td>0.2249</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ours(3,10)</td>
<td>ours 480p 1D &amp;3D</td>
<td>3× spatial-aware</td>
<td>22.58</td>
<td>0.7316</td>
<td>0.1878</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ours*(3,10)</td>
<td>ours 4K 1D &amp;3D</td>
<td>3× spatial-aware</td>
<td>22.56</td>
<td>0.7420</td>
<td>0.1580</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ours-noresize*(3,10)</td>
<td>ours 4K 1D &amp;3D</td>
<td>3× spatial-aware</td>
<td>22.65</td>
<td>0.7323</td>
<td>0.2142</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 4: Ablation studies on the number of categories (M).

**Application settings.** We implement our network with pytorch [17] and train all modules on a NVIDIA V100 GPU for 400 epochs with a mini-batch of 1. The entire network is optimized using standard Adam [12] with a cosine annealing learning rate with amplitude 2e-4 and period 20 epochs. The spatial-aware trilinear interpolation is accelerated via customized CUDA code.

### 4.1. Ablation study

To demonstrate the effectiveness of different components of our approach, we conduct several ablation studies on our HDR+ dataset.

**Number M of LUTs.** We assess the performance of different settings to determine the number of pixel-wise category for spatial-aware 3D LUTs with $T = 3$. Figure 4 shows models’ performance with different pixel-wise category number (M) from $M = 1$ to $M = \{2, 3, 4, 6, 8, 10, 12, 14, 16, 24, 32, 64\}$. We can see an evident improvement by increasing $M$ from 1 to 10, but minor improvement or even deterioration if $M$ is further increased. Therefore, $M$ is set to 10 in all our following experiments.

**Two-head weight predictor.** To further demonstrate the contribution of our whole architecture, we continuously conduct the following experiments with different combination of CNN weight predictor and 3D LUTs. $(t, m)$ represents the CNN configuration with $T = t, M = m$.

As shown in Table 1, directly increasing the number of LUTs based on Zeng’s [25] method cannot improve performance effectively. Both our two-head weight predictor and spatial-aware 3D LUTs are important. Our 1D weight (i.e., ours(30,0)) cannot work well alone when it is used alone, even if the number of LUTs is the same as our final configuration. The 3D weight (i.e., ours(0,30)) shows effectiveness in performance improvement when co-operated with our spatial-aware interpolation. When both 1D weight and 3D weight are utilized (i.e., ours(3,10)), our model performs better, which shows a total of 2.82 dB improvement in PSNR when compared with the original one. Additionally, our method can also work well on full resolution image (i.e., ours*(3,10)), only with 0.17dB degradation in PSNR. After deleting the first and last resize operation in CNN weight predictor (i.e., ours(3,10)), it can achieve 0.09 dB improvement, but the computation FLOPS improves from 8.111G to 113.79G. Therefore, we use ours(3,10) with resize operation as shown in Figure 2 as our final architecture.

**Loss function.** The loss function in [25] is defined as our basic loss ($L_b$), which is a combination of MSE loss, smooth loss and monotonicity loss. We train our models using different combinations of losses to evaluate the influence of our loss function.
quantitative results are demonstrated in Table 2, indicating that the model trained with only \( L_b \) gets relatively poor results, and that after the introduction of color loss and perception loss, all 3 metrics are dramatically improved. More analysis can be found in Figure 5.

4.2. Analysis of pixel-wise category map

Some pixel-wise category maps are visualized for analysis. We do not apply any loss on the category map, willing that it can be image adaptive for the network, but not perception adaptive from person’s point of view. On one hand, a perception adaptive category map does not guarantee better performances. In fact, we first apply explicit loss on the category map, but found 1.52dB degradation in PSNR. On the other hand, images can be categorized more flexibly, according to semantic, illumination, or other properties learned by the network itself. For the first row in Figure 6, the image may be classified semantically, with three maps indicating person, sky, and building respectively. For the second row, brightness is learned by our weight predictor, where three maps represent middle, low and high illumination areas.

4.3. Comparison with State-of-the-Arts

We compare our approach with several SOTA supervised image enhancement methods including DPED [10], RSGUNet [9], HPEU [8], HDRNet [5], UPE [21], DeepLPF [16], 3DLUT [25] on MIT-Adobe FiveK and HDR+ dataset. Among these methods, DPED, RSGUNet HPEU and DeepLPF are pixel-level enhancement methods based on ResNet and Unet backbone, while HDRNet and UPE belongs to patch-level methods, and 3DLUT is the image-level method. All of these methods are trained by publicly available source codes with recommended configurations.

As shown in Table 3, our approach outperforms other methods in terms of PSNR and LPIPS on MIT-Adobe FiveK. For SSIM on 480p, our result is a bit lower (<1%) than DeepLPF, but all other metrics are much better than DeepLPF. Particularly, due to the large memory consumption, the complicated DeepLPF algorithm cannot be applied on full resolution image (i.e., 4K resolution image). Similar result can be seen in Table 4 on HDR+ dataset. Our model outperforms the second best model by 0.85dB and 1.39dB on 480p and full resolution respectively. The performance gap between our 480p HDR+ dataset and [25] may be that the number of our HDR+ dataset is much larger than [25], resulting more serious disalignment. Mapping for HDR+ dataset is more locally complicated that it contains scenarios with wider dynamic range and more various illumination. Hence, our spatial-aware 3D LUTs with pixel-wise category map is more adaptive to those local variant transformations and have an evident improvement. On all datasets, our method achieves great improvement compared with basic 3DLUT method in all criterions. As the visual result shown in Figure 7 and Figure 8, it indicates that our results are more visually pleasant, and are closer to the ground-truth. More visual results can be found in supplementary material.

Apart from pleasant visual perception, our method is efficient for both low and high resolution images. Table 5 shows the inference time for different models with input size \( 640 \times 480, 1920 \times 1080, \) and \( 3840 \times 2160 \) on 32GB NVIDIA V100 GPU. Our model takes a bit longer running time when compared with 3DLUT, but it is about two-order faster than all other methods. Additionally, it only takes about 4 ms for our model to process a 4K resolution image, which exceeds the requirement of real-time processing by a large amount. The high efficiency mainly owns to the characteristic that our CNN network generates two
kinds of weight information from a low resolution input, and that the spatial-aware interpolation sensitive to the image size is greatly accelerated via customized CUDA codes. Thus, running time for our spatial-aware 3D LUTs remains approximately unchanged, while other competing methods except 3DLUT takes exponentially longer time as the resolution gets higher.

5. Discussion and Conclusion

Traditional 3DLUTs interpolate merely through RGB colors, leading to pool local contrast, while bilateral grids interpolate through luminance and space. However, this leads to more computational overheads and longer inference time for bilateral grids, since they are strongly coupled with slicing operation and a guide map of input’s resolution. Computation of this guide map is heavy and time consuming, especially for high resolution inputs. Table 4 shows that bilateral grids in HDRNet are sensitive to resolution, with 0.91dB difference in PSNR for inferring 480p and 4K images. We can conclude that for a fixed grid size, performance for HDRNet decreases as input’s resolution gets larger.

Our proposed spatial-aware 3D LUTs, on the other hand, produce charming results with good local contrast in high efficiency. Its key idea is constructing spatial-aware 3D LUTs with pixel-wise category map to improve the robustness in local regions for traditional 3D LUT. Further, we design a two-head weight predictor that generates different level of category information, enabling our network to be image-level scenario and pixel-wise category adaptive. Extensive experiments on public datasets demonstrate the superiority of our method against many SOTA methods on both performance and efficiency.
Figure 7: Results comparison on 'a3909' of 480p MIT-Adobe FiveK dataset. Our result outperforms all other methods both in color and detail. For example, (b) is bit of red on face and cloths, (c), (d) and (f) suffer from insufficient saturation in background areas, (c) has obvious contour artifacts around the person, and (e) is a bit darker compared with our result and some textures on cloths are lost.

Figure 8: Results comparison on '5a9e_20150403_162152_482' of our 480p HDR+ burst photography dataset. Results from (c), and (d) are both slightly yellow especially in the blue block area. (b) shows severe bending artifact in sky. The result of (e) is blurred in the red block area. Our model is much closer to the ground-truth, with much better color, clearer texture and less artifacts. In addition, owning to the pixel-aware category information, our model is able to enhance local areas differently, while traditional (f) can only enhance the whole image uniformly with local area in red block remaining dark.
References


