IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received May 11, 2022, accepted May 19, 2022, date of publication May 30, 2022, date of current version June 10, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3178745

Learning Tone Curves for Local Image Enhancement

LUXI ZHAO, ABDELRAHMAN ABDELHAMED, AND MICHAEL S. BROWN
Samsung AI Center-Toronto, Toronto, ON M5G 1L7, Canada

Corresponding authors: Luxi Zhao (lucy.zhao@samsung.com) and Abdelrahman Abdelhamed (a.abdelhamed@samsung.com)

ABSTRACT Image enhancement methods can be formulated as global transformations, local transformations, pixel-wise processing, or a mixture of these operations. Global transformations are limited in enhancing local image regions. Existing local and pixel-wise methods mitigate this issue, but give rise to the additional challenge of limited interpretability. Bridging the gap between global and local methods, we propose a local tone mapping network (LTMNet) that learns a grid of tone curves to locally enhance an image. Tone curves are commonly used by photo-editing software and offer an intuitive representation to photographers, facilitating subsequent customization of the image. Tone curves are also widely used in image signal processors (ISPs), making our method easy to deploy on cameras. Because existing datasets contain image enhancement and photofinishing beyond global and local tone mapping, we also propose a new dataset representative of local tone mapping—the LTM dataset. We evaluate our method on this new dataset as well as MIT-Adobe and HDR+ datasets. We show that the proposed LTMNet outperforms existing methods in local tone mapping while achieving competitive performance modeling additional photofinishing. Furthermore, we show that our method can be assistive in user-interactive photo-editing tools. Our code, model, and data will be released publicly at https://github.com/SamsungLabs/ltmnet.

INDEX TERMS Deep learning, image enhancement, local tone mapping, tone curves.

I. INTRODUCTION

Cameras capture valuable moments in our daily life in the form of photographs. Most cameras use dedicated image signal processors (ISPs) to process the captured sensor image into the final output image. ISPs apply several steps in a pipeline fashion to process images. One of the key operations is tone mapping. Tone mapping is an essential step in the photo enhancement stages of ISPs and has a major impact on the quality of the final image by enhancing the contrast and color tones of the image.

A tone map converts an input pixel intensity to a new output intensity. Generally, the same or different tone maps are applied to the R, G, and B channels of a color image. This operation is efficient to perform in hardware using a lookup table (LUT). Tone maps are often called by other names: for example, tone curve, transfer function, and 1D LUT. Tone mapping is widely used in dynamic range compression (e.g., [3], [4], [5]), reducing a high dynamic range (HDR) image to a lower dynamic range (LDR) while preserving an aesthetically pleasing appearance. In this paper, however, we place our focus on transformations within the same dynamic range, rather than HDR to LDR. Tone mapping can be applied in two manners, *global* and *local*. Global tone mapping (GTM) maps each pixel value to another value regardless of pixel location. For a typical real-world ISP, GTM alone is rarely sufficient to adequately enhance an image. In particular, GTM lacks flexibility and produces over/under-enhanced local regions, as shown in Fig. 1-(B). Local tone mapping (LTM), on the other hand, spatially adjusts image regions using different tone curves based on local characteristics. LTM offers more fine-grained control and helps bring out highlights. As shown in Fig. 1-(D), our LTM method provides transformations tailored to a local region—for example, increasing the visibility of regions of dense content (the bushes), and increasing contrast of the shadow regions to provide more vibrant imagery effects.

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

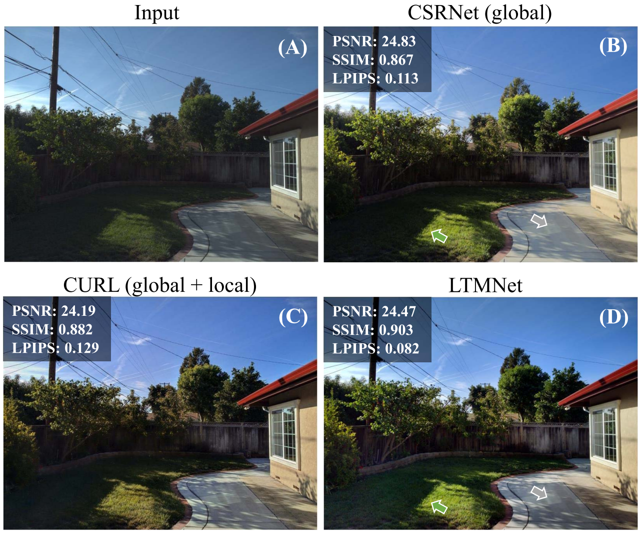


FIGURE 1. An example comparing our method to two state-of-the-art enhancement methods. (A) Input image. (B) CSRNet [1] uses a multi-layer perceptron as a global transfer function. (C) CURL [2] uses pixel-wise processing and global transfer functions. (D) Our LTMNet uses local tone curves that produce better local contrast than CSRNet (see the more visible bushes on the left of the image), while avoiding color shifts produced by CURL (see the sky region). Green and gray arrows point to areas of notable differences.

Many existing methods either perform GTM only (e.g., [1], [6]), or perform pixel-wise enhancements (e.g., [2], [7]), which output an enhanced image rather than transfer functions. Little work focuses on explicitly learning local tone curves, which can be efficiently integrated into existing ISPs as 1D LUTs, making them more convenient than pixel-wise processing and more powerful than GTM.

Contribution We propose a deep-learning approach to local image enhancement that estimates local tone curves. Unlike existing methods, our approach is trained to output a grid of local tone curves instead of pixel-wise processing. Tone curves are more intuitive for post-processing editing and implementation in hardware. Because tone curves can be applied to images of any size, our method is not limited to a specific resolution. We also introduce a new image dataset representative of local tone mapping, consisting of tone-mapped and non-tone-mapped image pairs. In addition, we provide a tool for interactive LTM manipulation that can be used to manually fine-tune the tone curves predicted automatically by our method.

II. RELATED WORK

There is a large body of work on image enhancement; only representative works are presented here. We divide these methods into three main categories based on the way they process the image: (1) methods that apply global transfer functions or LUTs to the whole image; (2) methods that apply local enhancement to local image regions; (3) methods that apply pixel-level mapping from input to enhanced image. Some methods may combine two or more of global, local, and pixel-wise processing.

A. GLOBAL ENHANCEMENT

Traditional image enhancement methods apply pre-defined global transfer functions, such as gamma correction, or a transfer function estimated from the intensity distribution, such as histogram equalization [8] and its extensions: contrast-limited histogram equalization (CLHE) [9] and histogram modification framework (HMF) [10].

Recent methods use neural networks to predict global transformations. For example, the method in [11] proposes a neural network to implement CLHE and HMF. SpliNet [12] performs personalized enhancements using a learned global tone curve. The method in [6] learns a 3D lookup table to achieve fast and robust enhancement. White Box [13] selects the best sequence of global enhancement operations from a pre-defined set based on deep reinforcement learning (RL) guided by generative adversarial networks (GANs). Distort-and-recover [14] also uses RL to explicitly model the step-wise nature of the human retouching process. Similarly, [15] uses RL and unpaired images.

As mentioned in the introduction, global transformations can under- or over-enhance local image content, thus leading to the need for local enhancement methods.

B. LOCAL ENHANCEMENT

Many methods extend histogram equalization techniques to be locally adaptive [16]–[19]. One prominent example is adaptive histogram equalization (AHE) [20], [21] which involves equalizing a set of histograms computed from local image regions, typically a grid of patches. One further extension is contrast-limited AHE (CLAHE) [9], where the contrast amplification is limited by clipping the computed histograms. CLAHE is an industry standard adopted by many camera ISPs and typically used as a local tone mapping operator; however, it requires careful parameter tuning.

Some methods perform color transformations for local image enhancement. Color palette-based methods [22], [23] interpolate colors based on a sparse set of colors in the palette. However, updating the palette requires user interaction or example images. Representative color transform (RCT) [24] learns and transforms a set of representative colors in the image, globally and locally. HDRNet [25] learns a bilateral grid [26] of 3 × 4 affine transformation matrices from a down-sampled image. Each matrix maps an input color to an output color. These affine coefficients are then applied to the full-resolution input image through bilateral guided upsampling [27]. HDRNet is expressive at modeling complex transformations. However, the matrices are difficult to visualize and interpret. In contrast, our method learns a grid of 1D curves, which are intuitive to photographers, easy to interpret and edit. StarEnhancer [28] also learns a set of curves that transform an image based on both intensity values and pixel location. The curves can be manually fine-tuned, but it is difficult to pinpoint adjustments to a specific spatial coordinate. Our method, in comparison, explicitly maps each curve to a local region.

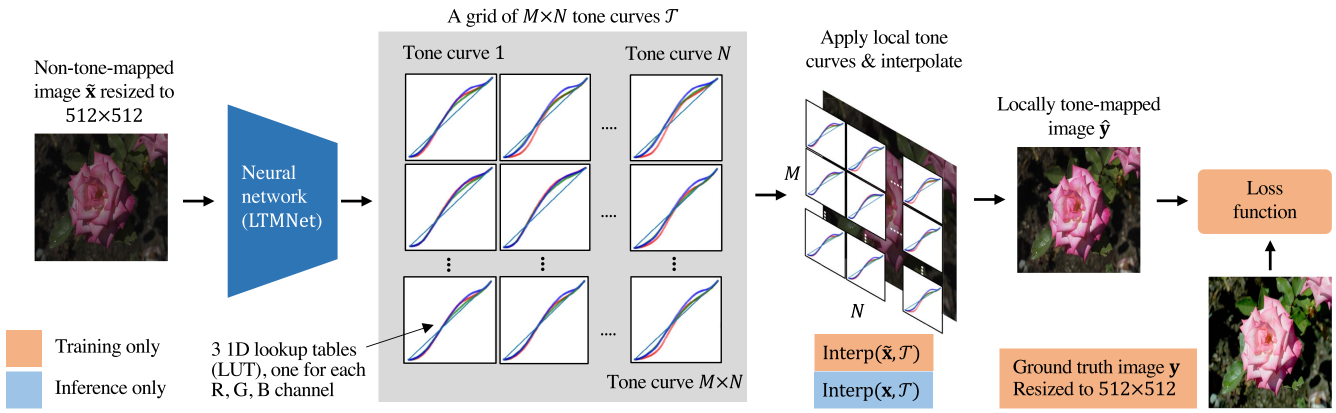

A grid of M×N tone curves $\mathcal{T}$
Non-tone-mapped image $\tilde{\mathbf{x}}$ resized to 512×512
Neural network (LTMNet)
Tone curve 1
Tone curve N
Tone curve M×N
Apply local tone curves & interpolate
M
N
Locally tone-mapped image $\hat{\mathbf{y}}$
Loss function
Interp($\tilde{\mathbf{x}}$, $\mathcal{T}$)
Interp($\mathbf{x}$, $\mathcal{T}$)
Ground truth image $\mathbf{y}$ Resized to 512×512
Training only
Inference only
3 1D lookup tables (LUT), one for each R, G, B channel

FIGURE 2. Overview of our local image enhancement pipeline. We first learn a grid of tone curves using a neural network. Each predicted tone curve corresponds to one patch of the input image. The predicted tone curves are then applied to each patch with tile-based interpolation.

C. PIXEL-WISE ENHANCEMENT

Traditional pixel-wise methods perform base-detail layer decomposition to enhance an image's high-frequency details. These methods include bilateral filtering [29], Laplacian operators [30], guided filtering [31], and just-noticeable-difference (JND) transform [32]. Numerous recent methods use convolutional neural networks (CNNs), especially encoder-decoder architectures [33]. The method in [34] maps input images to enhanced images using per-pixel quadratic color transforms. Work by [35] maps low-quality smartphone images to corresponding DSLR high-quality images. Some methods employ GANs, such as WESPE [36] and deep photo enhancer (DPE) [37]. EnhanceGAN [38] applies weak supervision using binary labels of image aesthetic quality to estimate piece-wise transfer functions on the CIELab color space. PieNet [39] incorporates user preferences by injecting a preference vector into its base network. CSRNet [1] processes pixels independently using a multi-layer perceptron (MLP) modulated by a global feature vector extracted from a condition network. Neural curve layers (CURL) [2] predicts a sequence of global transfer functions applied in different color spaces while using a backbone CNN for local enhancement. Both [7] and [40] learn global tone curves and a pixel-wise residual map for local enhancement. IceNet [41] personalizes local contrast enhancement by predicting per-pixel gamma-correction values based on a global brightness parameter and a scribble map, both interactively provided by the user. Our method combines automatic image enhancement with the option of manual post-editing to minimize user efforts while allowing interactivity.

Pixel-wise methods are less explainable as it is difficult to identify what operations are performed on the input image, whereas our method explicitly specifies the transformations.

D. OTHER METHODS

Another set of methods addresses underexposure enhancement, such as DeepUPE [42], DRHT [43], and [44]. Similarly, other methods focus on low-light image enhancement, such as [45]–[47]. Zero-reference deep curve estimation (Zero-DCE) [48] is a pixel-wise curve-based method that targets low-light images without reference images. Some methods rely on physical models of image formation (e.g., the Retinex theory of color vision [49]). Such methods include exposure correction methods based on separation of scene reflectance and illumination [50], illumination estimation [42], [51], and modeling of camera response functions [52].

As an alternative to CNNs, STAR [53] is a fast and lightweight backbone network for multiple image enhancement tasks, such as white-balancing, low light image enhancement, and photofinishing.

Global transformations may not be sufficient to estimate highly non-linear mapping between low-quality and high-quality images. Methods based on pixel-wise processing usually are hard to interpret, fine-tune, or integrate into ISPs or photo-editing software. Our method is based on learning local tone curves for local image regions; this makes it more flexible than global enhancement methods. Also, tone curves are well understood, interpretable, and widely used in many camera ISPs and photo-editing software. To the best of our knowledge, our method is the first to introduce learning local tone mapping automatically in a data-driven manner instead of manual tuning.

III. LTMNet

Our method, illustrated in Fig. 2, aims to perform local image enhancement through learning a grid of local tone maps, inspired by the well-established CLAHE algorithm [9]. Given an input image $\mathbf{x} \in \mathbb{R}^{H\times W\times C}$, we use a neural network LTMNet to predict a set of local tone maps (LTMs) $\mathcal{T} \in \mathbb{R}^{M\times N\times C\times L}$ from an input image:

$$\mathcal{T} = \text{LTMNet}(\mathbf{x}), \tag{1}$$

where H, W, and C are image height, width, and number of channels, respectively. M and N are the height and width, respectively, of a grid of image patches. L is the number of intensity levels, typically 256 for 8-bit integer images.

LTMNet layers serve two purposes: feature extraction and tone curve prediction. For feature extraction, a wide range of architectures can be used, as long as the receptive fields of the output neurons composing the tone curves cover the image patches on which they are applied. A tone curve prediction head can be stacked on top of the feature extraction layers to ensure tone curve entries are in the desired shape and range (i.e., $M \times N \times C \times L$). For efficiency, we design LTMNet such that the input image is always resized to a fixed input size (e.g., 512 × 512).

A. LOCAL TONE CURVES

The output of LTMNet, $\mathcal{T}$, represents a set of transfer functions (i.e., tone curves) that are applied to the input image to adjust its local contrast, brightness, and colors. LTMNet predicts a number of tone curves or 1D lookup tables (LUTs) for each image patch in an $M \times N$ grid. For a typical standard RGB (sRGB) image, three 1D LUTs are predicted for each patch, one for each R, G, and B channel:

$$\mathcal{T} = \{\mathbf{t}_{m,n,c}\} \tag{2}$$

where $m \in \{0, \ldots, M-1\}$, $n \in \{0, \ldots, N-1\}$, and $c \in \{0, 1, 2\}$. Thus, $M \times N \times 3$ tone curves are predicted in total. Each tone curve is represented by a 1D LUT that has L entries, $\mathbf{t} \in \mathbb{R}^L$. Each entry maps an input pixel intensity to an output enhanced intensity.

The application of the predicted local tone curves on the input image is performed using bilinear interpolation between each set of local tone curves in order to produce a smooth and artifact-free locally tone-mapped image $\hat{\mathbf{y}} \in \mathbb{R}^{H \times W \times C}$:

$$\hat{\mathbf{y}} = \text{Interp}(\mathbf{x}, \mathcal{T}). \tag{3}$$

B. LOCAL TONE CURVE INTERPOLATION

A predicted tone curve $\mathbf{t}_{m,n}$ is most appropriate for the center pixel of patch (m, n) in the $M \times N$ grid. Intuitively, all other pixels in the patch are influenced by the tone curves of neighbouring patches by varying degrees, according to the distance of the pixel to the neighbouring patch centers. This way, the tone curve for each pixel smoothly transitions to another, resulting in a continuous output image free of boundary artifacts.

Our tone curve interpolation module, Interp, transforms all non-center pixels by a combination of neighboring tone curves whose patch centers are closest to it, as shown in Fig. 3. Pixels in the center region of the image are bilinearly interpolated, combining the influence of the four neighboring tone curves.

Specifically, suppose (i_1, j_1), (i_2, j_1), (i_1, j_2), (i_2, j_2) are the (x, y) coordinates of the four patch centers closest to location (i, j) of input image $\mathbf{x}$, in the order of top left, top right, bottom left, and bottom right respectively. Moreover, suppose $\mathbf{t}_1$, $\mathbf{t}_2$, $\mathbf{t}_3$, $\mathbf{t}_4$ are the predicted tone curves of the four patch centers in the same order. The interpolated pixel value at (i, j) is given by Equation 4:

$$\begin{aligned}
\hat{\mathbf{y}}(i,j) \leftarrow\ & (i_2 - i)(j_2 - j)\mathbf{t}_1([\mathbf{x}(i,j) \cdot (L-1)]) \\
& + (i - i_1)(j_2 - j)\mathbf{t}_2([\mathbf{x}(i,j) \cdot (L-1)]) \\
& + (i_2 - i)(j - j_1)\mathbf{t}_3([\mathbf{x}(i,j) \cdot (L-1)]) \\
& + (i - i_1)(j - j_1)\mathbf{t}_4([\mathbf{x}(i,j) \cdot (L-1)]) \\
\hat{\mathbf{y}}(i,j) \leftarrow\ & \hat{\mathbf{y}}(i,j)/(i_2 - i_1)(j_2 - j_1)
\end{aligned} \tag{4}$$

where $\mathbf{x}(i, j) \in [0, 1]$, and $[\cdot]$ indicates rounding to the nearest integer. L is the number of intensity levels, typically 256 for 8-bit integer images.

Similarly, pixels in the border region are linearly interpolated. Take a location (i, j) in the top or bottom border region as an example; suppose (i_1, j_1) and (i_2, j_1) are the (x, y) coordinates of the two patch centers closest to location (i, j), in the order from left to right. Moreover, suppose $\mathbf{t}_1$ and $\mathbf{t}_2$ are the predicted tone curves of the two patch centers in the same order. The interpolated pixel value at (i, j) is given by:

$$\begin{aligned}
\hat{\mathbf{y}}(i,j) \leftarrow\ & (i_2 - i)\mathbf{t}_1([\mathbf{x}(i,j) \cdot (L-1)]) \\
& + (i - i_1)\mathbf{t}_2([\mathbf{x}(i,j) \cdot (L-1)]) \\
\hat{\mathbf{y}}(i,j) \leftarrow\ & \hat{\mathbf{y}}(i,j)/(i_2 - i_1)
\end{aligned} \tag{5}$$

Finally, pixels in the four corner regions are not interpolated. Suppose $\mathbf{t}$ is the predicted tone curve of the patch center closest to a position (i, j) in one of the corner regions; the tone-mapped pixel value at (i, j) is given by:

$$\hat{\mathbf{y}}(i,j) = \mathbf{t}([\mathbf{x}(i,j) \cdot (L-1)]) \tag{6}$$

The input image to both the tone curve prediction network LTMNet and the interpolation module Interp can take on any shape because the application of tone curves only transforms pixel values and is independent of the image's spatial dimensions. The final output is a continuous locally tone-mapped image $\hat{\mathbf{y}}$, with the same resolution as the input image $\mathbf{x}$.

C. TONE CURVE CONSTRAINTS

For each image patch in the $M \times N$ grid, its corresponding lookup table maps each pixel value to some other value according to the table entries. The entries in the lookup table are enforced to be non-decreasing to maintain intensity rank consistency. Furthermore, maximum intensity is kept unchanged in the LUT to preserve information in the overexposed regions. The tone curve constraints are implemented through integrating and normalizing non-negative output neurons:

$$t_l = \frac{1}{\sum_{i=0}^{L-1} \hat{t}_i} \sum_{i=0}^{l} \hat{t}_i, \tag{7}$$

where $\hat{t}$ is one output neuron from the last layer of the neural network, which is followed by a sigmoid activation to constrain the neurons such that $\hat{t} \in [0, 1]$. Integration of $\hat{t}$ enables t_l, an entry in tone curve $\mathbf{t}$, to be non-decreasing over the range $l \in [0, L-1]$.

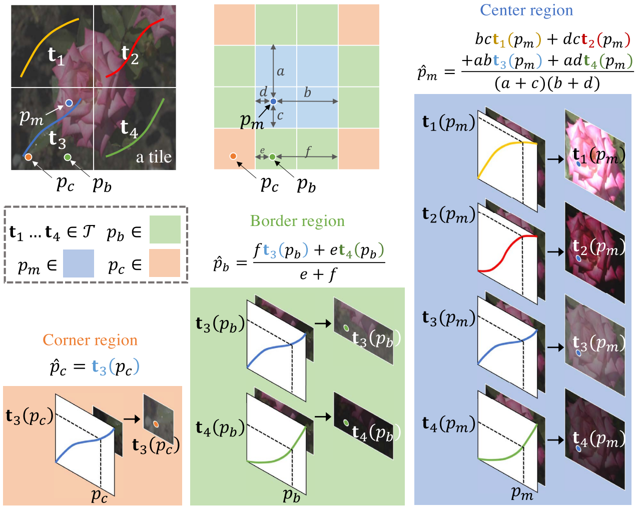

t1
t2
t3
t4
pm
pc
pb
a tile
a
b
c
d
e
f
t1 ... t4 ∈ 𝒯
pb ∈ 🟩
pm ∈ 🟦
pc ∈ 🟧
Center region
$\hat{p}_m = \frac{bc\mathbf{t}_1(p_m) + dc\mathbf{t}_2(p_m) + ab\mathbf{t}_3(p_m) + ad\mathbf{t}_4(p_m)}{(a+c)(b+d)}$
Border region
$\hat{p}_b = \frac{f\mathbf{t}_3(p_b) + e\mathbf{t}_4(p_b)}{e+f}$
Corner region
$\hat{p}_c = \mathbf{t}_3(p_c)$
t1(pm)
t2(pm)
t3(pm)
t4(pm)
t3(pb)
t4(pb)
t3(pc)

FIGURE 3. Illustration of interpolation of tone curves for a 2 × 2 grid. p_m, p_b, and p_c are input pixels located in the center, border, and corner regions respectively. $\hat{p}_m$, $\hat{p}_b$, and $\hat{p}_c$ are output pixels resulting from the corresponding input pixels being transformed by the tone curves and interpolated. $\mathbf{t}_1$ to $\mathbf{t}_4$ are the predicted tone curves (LUTs) of the four patch centers. $\mathbf{t}_k(p_k)$ indicates looking up pixel value p_k in $\mathbf{t}_k$. For center regions, bilinear interpolation is performed between the four neighbor tone curves. For border regions, linear interpolation is performed between the two neighbor tone curves. For corner regions, the corner tone curve is applied directly.

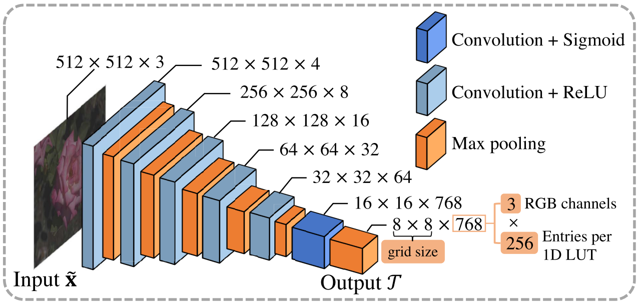


FIGURE 4. Example network architecture of our LTMNet. $M = N = 8$. $L = 256$. The first layer size is 512 × 512 × 4, followed by a sequence of convolutional, non-linear activation [58], and max pooling layers. The number of layers is adjusted such that the output shape is consistent with the shape of the predicted tone curves $\mathcal{T}$.

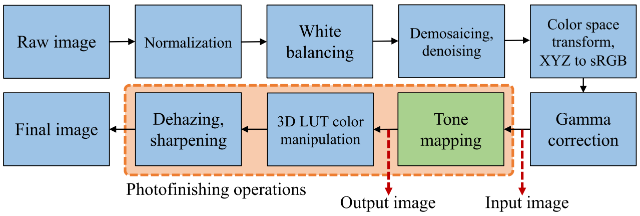


FIGURE 5. Illustration of a typical camera pipeline. Our LTM dataset targets the tone mapping stage only. Input and output image pairs are extracted immediately before and after tone mapping, respectively.

D. LOSS FUNCTIONS

We use two loss functions to drive model training: L_1 and perceptual loss [54]. L_1 loss minimizes the fidelity difference between the predicted image $\hat{\mathbf{y}}$ and its corresponding ground-truth image $\mathbf{y}$. For perceptual loss, we use the initial two layers of VGG19 [55] (`block1_conv1` and `block2_conv1`), which is trained on ImageNet [56] to minimize squared L_2 distance between the features of predicted and target images. Since the predicted and target images differ only in terms of low-level features, such as brightness, contrast, and color, only layers of the initial two VGG blocks are used for the loss function. Deeper VGG layers are not used because they primarily encode high-level information, such as object shape and spatial arrangement [57], which are already identical between our paired images. Our loss function is

$$\mathcal{L} = \lambda_{l_1} \|\hat{\mathbf{y}} - \mathbf{y}\|_1 + \lambda_p \sum_{k=1,2} \|\phi_k(\hat{\mathbf{y}}) - \phi_k(\mathbf{y})\|_2^2, \quad (8)$$

where ϕ_k indicates VGG19 features from the first convolutional layer in the k^{th} block. We empirically set the L_1 loss weight λ_{l_1} to 3.0 and the perceptual loss weight λ_p to 10^{-4}.

E. NETWORK ARCHITECTURE

The architecture of LTMNet is shown in Fig. 4. The first layer size is 512×512×4, followed by a sequence of convolutional, non-linear activation [58], and max pooling layers. Either the number of layers or the pool size of the last pooling layer can be adjusted such that the output shape is consistent with the shape of the tone curves $\mathcal{T}$.

IV. DATASETS

A. EXISTING DATASETS

Two commonly used datasets in image enhancement are MIT-Adobe FiveK [59] and HDR+ [60]. MIT-Adobe FiveK contains 5,000 pairs of input and enhanced images retouched by five professional experts. However, this dataset involves mostly global tone mapping among other photo retouching operations [61].

The HDR+ dataset consists of 3,640 image bursts, which make up 28,461 images in total. Each burst is processed into a merged, aligned, and enhanced single output high dynamic range (HDR) image. This dataset includes strong local tone mapping and is more suitable for evaluating our method. However, it also includes other photofinishing operations, such as sharpening and hue/saturation adjustment.

For evaluation on the HDR+ dataset, we prepare image pairs as follows. We process the raw-RGB merged frame into a gamma-corrected sRGB image using a simulated image signal processor (ISP) [62] and use it as input. We use the final photofinished JPEG image as output. We prepared around 2,000 image pairs.

B. OUR LTM DATASET

As illustrated in Fig. 5, in a typical ISP, tone mapping and other photofinishing operations, such as color manipulation, are often performed in separate stages. To the best of our knowledge, there is no image dataset involving local tone mapping only; existing datasets include global tone mapping or local tone mapping mixed with other photofinishing

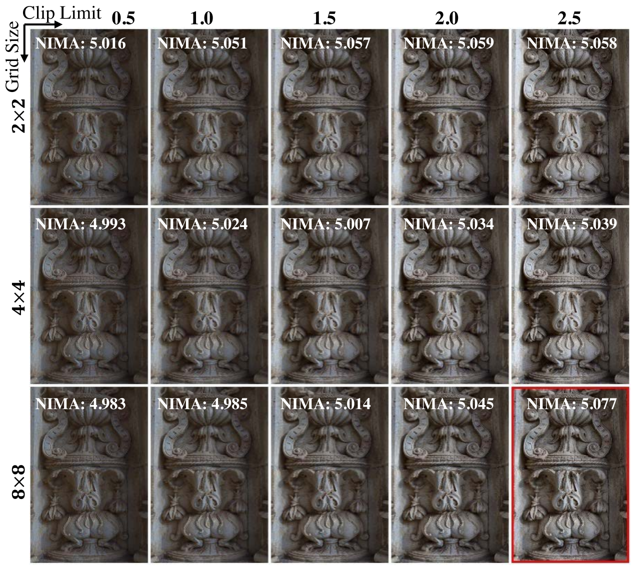


FIGURE 6. Example of all 15 versions of an image enhanced by CLAHE [9]. The version with the highest NIMA [63] score is highlighted in red.

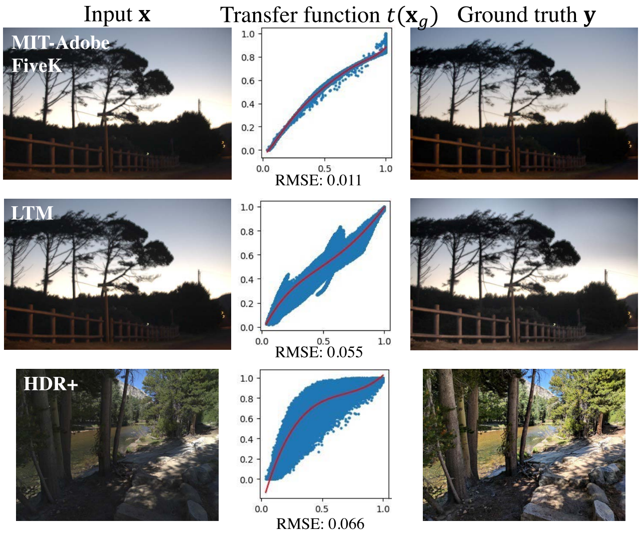


FIGURE 7. Example best-fit 4-degree polynomial transfer functions between image pairs from MIT-Adobe FiveK, HDR+, and our LTM datasets. For display purposes, only the green channel's transfer function is shown. The MIT-Adobe dataset barely contains local processing. HDR+ contains significant local processing, including local tone mapping. Our LTM dataset includes local tone mapping only.

operations. To overcome this issue, we used CLAHE [9], a widely adopted industry standard for local tone mapping in ISPs, to generate a dataset of image pairs. Each pair consists of an sRGB image with global gamma correction and the corresponding locally tone-mapped image using CLAHE. We used MIT-Adobe FiveK to generate our dataset. A major limitation of CLAHE is that it requires manual tuning of its parameters, the grid size and the contrast limit. Instead of manually tuning these parameters for each image, we perform a grid search on the parameters for each image and automatically select the parameter values that produce an image with the highest non-reference image quality metric. We use neural image assessment (NIMA) [63] as the non-reference metric as it corresponds well with human perception. Specifically, out of all versions of an enhanced image, NIMA is able to select one without artifacts. Appendix A provides further justifications for our choice of NIMA. Fig. 6 showcases examples of grid-searched images over 15 parameter combinations. Although the images selected by NIMA are mostly artifact-free, some poor-quality ones may still be selected, which are then manually removed. In the end, we removed 91 images out of 2,500 ($< 4\%$). These 4% are mostly images with large homogeneous regions (e.g., sky) that may not require local processing. Finally, we round down our LTM dataset to consist of 2,000 image pairs.

C. QUANTIFYING LOCAL TONE MAPPING

To estimate the extent of local tone mapping in each dataset, we perform the following experiment. We compute the root mean squared error (RMSE) of the best-fit 4-degree polynomial between the input and output image intensities, averaged over all images in the dataset. This metric gives an indication of how much the transformation between an image pair deviates from a single global transfer function, and hence, it also indicates how much local processing exists in the images. The RMSEs for MIT-Adobe, HDR+, and our LTM dataset are 0.0229, 0.0483, and 0.0404, respectively. The results indicate that the MIT-Adobe dataset does not contain much local processing, while HDR+ contains significant local processing, including local tone mapping. Our LTM dataset contains noticeable local processing; but unlike the other datasets, it is restricted only to local tone mapping. Fig. 7 shows an example of the fitted transfer functions between example images from the three datasets.

V. EXPERIMENTS

For the following experiments, we use the LTMNet architecture shown in Fig. 4 that contains six convolutional layers and produces a 3D grid of $8 \times 8 \times 3$ tone curves of size 256.

A. EVALUATION ON THE LTM DATASET

We evaluated our method on our LTM dataset as it contains local tone mapping only and to avoid the effect of other photofinishing operations that exist in other datasets. We compare our method against state-of-the-art (SOTA) image enhancement methods: CURL [2], Zero-DCE [48], HDRNet [25], CSRNet [1], and Pix2Pix [64]. We use the following metrics: peak signal-to-noise ratio (PSNR), structural similarity (SSIM) [65], and learned perceptual image patch similarity (LPIPS) [54]. Table 1 shows the performance of our method and SOTA methods on the LTM dataset. Our method outperforms all SOTA methods in all metrics. Fig. 8 shows some visual comparisons. Our LTMNet produces visually enhanced images with vivid local contrast while avoiding structural and color artifacts. CURL and CSRNet seem to be limited at enhancing local contrast, while Pix2Pix

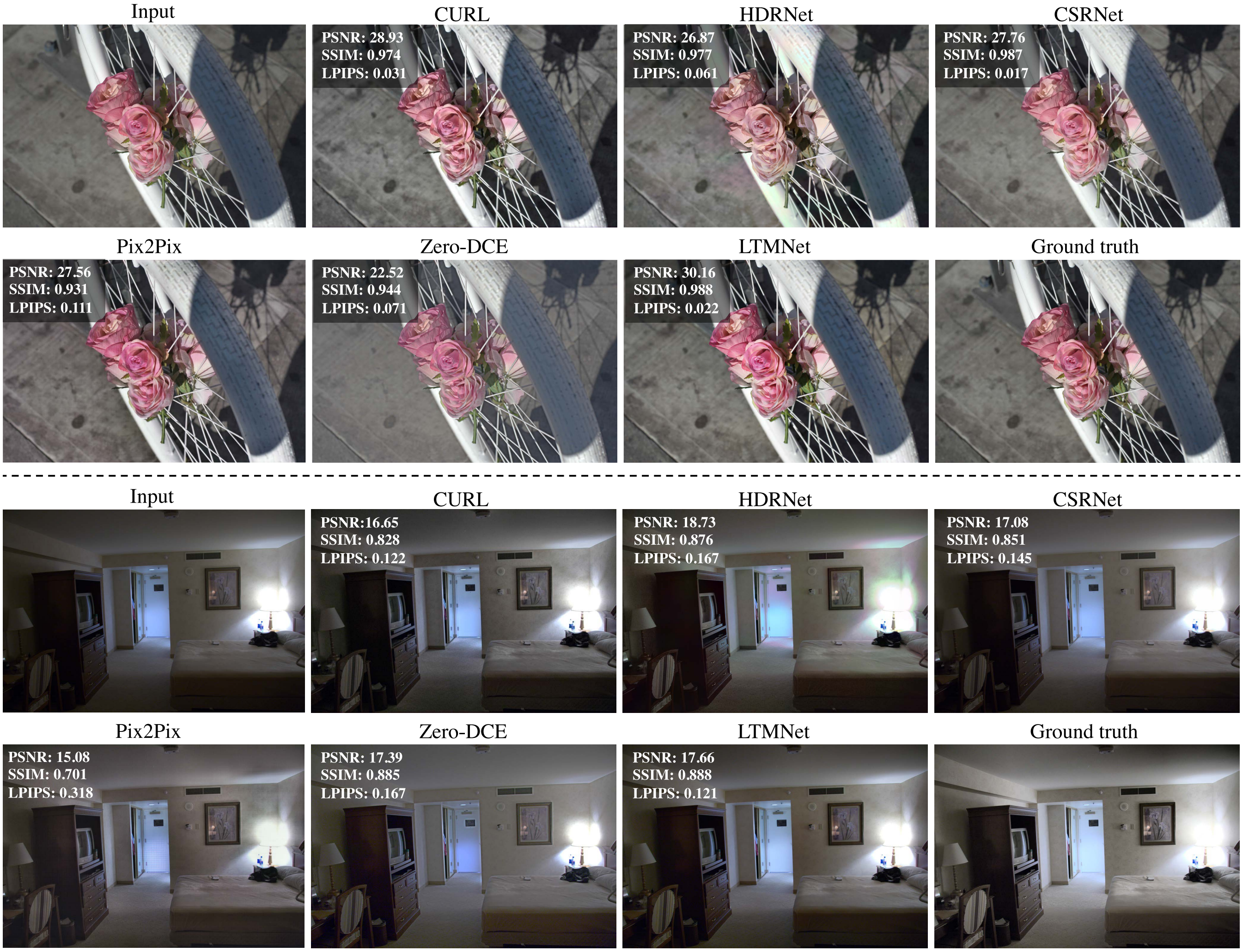


FIGURE 8. Visual comparison of our LTMNet against SOTA methods: CURL [2], HDRNet [25], CSRNet [1], Pix2Pix [64], and Zero-DCE [48], on our LTM dataset. Our LTMNet produces visually enhanced results while avoiding structural and color artifacts.

and HDRNet are prone to structural or color degradations. Zero-DCE has a relatively low performance because it uses non-reference loss functions. Additional results are provided in Appendix B.

B. EVALUATION ON THE HDR+ DATASET

To verify our method's capability of modeling generic local tone mapping effects in addition to the CLAHE algorithm, we also evaluated our method on HDR+ against SOTA methods. The quantitative results are shown in Table 1. Visual comparisons are shown in Fig. 10. Our LTMNet method yields comparable results to SOTA. LTMNet does not outperform SOTA methods on the HDR+ dataset because such a dataset contains more processing beyond local tone mapping, such as sharpening and hue/saturation adjustments, while our LTMNet is limited to using local tone curves only. Also, other methods use pixel-wise processing, which is more expressive in modeling fine local detail enhancement, such as sharpening. To see if pixel-wise processing can help our LTMNet in modeling the additional photofinishing in HDR+ images, we append a small residual network with 1.6K parameters to LTMNet, naming this model "LTMNet + Res." This model closes the gap with SOTA methods in terms of SSIM and LPIPS, while boosting PSNR by a large margin. This indicates the effectiveness of pixel-wise processing in modeling additional photofinishing operations. Additional results are provided in Appendix B.

C. CHOICE OF GRID SIZE

To select the best size for the tone curve grid, we evaluated multiple grid sizes on our LTM dataset, as shown in Table 2. Grid size 8 × 8 produces the best results for all metrics.

D. CHOICE OF CONTROL POINTS

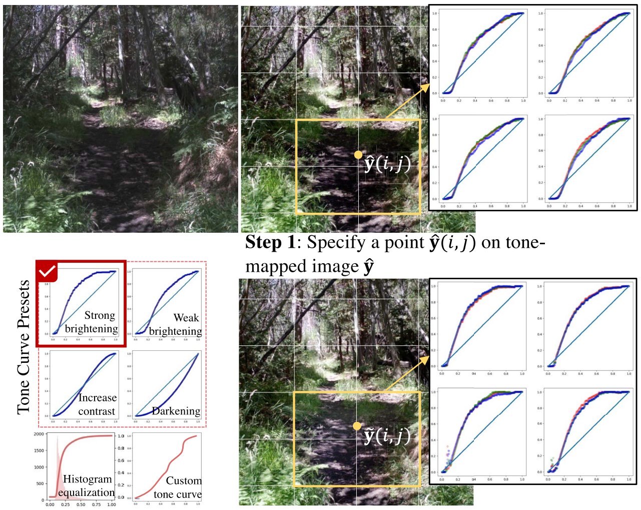
We performed experiments with smaller numbers of control points for the LUTs, as shown in Table 5. Control points are interpolated with monotone cubic splines. Fewer

TABLE 1. Quantitative comparison between our method and SOTA methods on our LTM dataset and HDR+ dataset. ↓ means smaller values indicate better performance and vice versa.

Dataset	Method	PSNR↑	SSIM↑	LPIPS↓
LTM	CURL [2]	26.46	0.934	0.063
	Zero-DCE [48]	21.13	0.898	0.109
	HDRNet [25]	25.75	0.946	0.075
	CSRNet [1]	26.31	0.943	0.059
	Pix2Pix [64]	22.00	0.784	0.198
	LTMNet (ours)	**27.00**	**0.953**	**0.058**
HDR+	CURL [2]	25.32	0.878	0.098
	Zero-DCE [48]	16.35	0.677	0.285
	HDRNet [25]	**26.90**	0.892	0.082
	CSRNet [1]	26.13	0.877	0.101
	Pix2Pix [64]	22.85	0.791	0.207
	LTMNet (ours)	24.76	0.876	0.099
	LTMNet + Res. (ours)	25.20	**0.893**	**0.080**

TABLE 2. Ablation studies for grid size on the LTM dataset.

Grid Size	PSNR↑	SSIM↑	LPIPS↓
1 × 1	26.39	0.937	0.070
2 × 2	26.93	0.947	0.064
4 × 4	26.97	0.951	0.062
8 × 8	**27.00**	**0.953**	**0.058**
16 × 16	26.81	0.952	0.058



Step 1: Specify a point $\hat{\mathbf{y}}(i,j)$ on tone-mapped image $\hat{\mathbf{y}}$

Step 2: Specify a target tone curve

Step 3: Apply the target tone curve on $\hat{\mathbf{y}}(i,j)$ to obtain edited image $\tilde{\mathbf{y}}$

FIGURE 9. A use case of integrating our method with interactive editing of local tone curves. (Step 1) the user selects a local region in the locally tone-mapped image; (Step 2) the user selects a preset tone curve to enhance that region; (Step 3) we use tile-based interpolation to apply the selected tone curve on the local region while smoothly propagating its effect to the surrounding regions.

control points produce less optimal performance but use fewer parameters.

E. TRAINING AND HYPERPARAMETER SETTINGS

For both experiments on the HDR+ dataset and our LTM dataset, 1,400 images are used for training, 100 used for validation, and 500 used for testing. All visual results in the paper are sampled from the test set. For fairness of comparison, all SOTA methods are re-trained on the two datasets.

For training our model, we use Adam [66] as the optimizer with a learning rate of 0.001. Models are trained for 150 epochs and 250 epochs for the LTM dataset and HDR+ dataset, respectively, both with a batch size of 20. We augment the input with random flip to generalize the models for inputs of different orientations.

F. INTERACTIVE EDITING OF TONE CURVES

In addition to automatic local tone mapping, our method can be used in an interactive setting and integrated with photo-editing software. Users can apply our method to produce an automatically enhanced photo, and then manually enhance a local region of the image by modifying the local tone curve corresponding to that region. Fig. 9 shows a use case for integrating our method with interactive editing of local tone curves. We also prepared a video to show local tone curve editing: link.

After producing the locally tone-mapped image using our method, the user selects a point $\hat{\mathbf{y}}(i,j)$ on the image to modify the patch containing the point. The tone curve applied at location (i,j) is a weighted average of tone curves predicted at its closest patch centers. Suppose (i,j) is located in the center region; the tone curve applied at (i,j) can be computed as follows:

$$\tilde{\mathbf{t}}_{ij} = \sum_{k=1}^{4} w_{ijk}\mathbf{t}_k, \tag{9}$$

where $\mathbf{t}_k$ is one of the four component tone curves in Equation 4. w_{ijk} represents the weight given to a component tone curve at location (i,j) that is inversely proportional to its distance from point (i,j). For example, $w_{ij1} = \frac{(i_2-i)(j_2-j)}{(i_2-i_1)(j_2-j_1)}$, which corresponds to the first weight term in Equation 4. Similarly, the interpolated tone curves at the border regions can be inferred using Equation 5.

Afterwards, the user defines a target tone curve at location (i,j). Step 2 in Fig. 9 presents three possible options: (1) selecting from a set of preset tone curves, (2) using the cumulative distribution function of the selected region, and (3) using a self-defined LUT. The target tone curve $\mathbf{t}^*$ can be treated as a scaled version of $\tilde{\mathbf{t}}$:

$$\mathbf{t}^*_{ij} = \mathbf{s} \odot \tilde{\mathbf{t}}_{ij}, \tag{10}$$

where elements of $\mathbf{s}$ are the scaling factors transforming each entry in $\tilde{\mathbf{t}}$ to the target tone curve entry. Given a target tone curve, the scaling factors can be computed by element-wise division of the target tone curve $\mathbf{t}^*$ by the original tone curve $\tilde{\mathbf{t}}$.

Next, tone curves predicted at the closest patch centers—namely, $\mathbf{t}_k$, $k \in \{1, \ldots, 4\}$—are modified so that their interpolated result matches exactly with the target tone curve. This can be achieved by simply multiplying the component tone

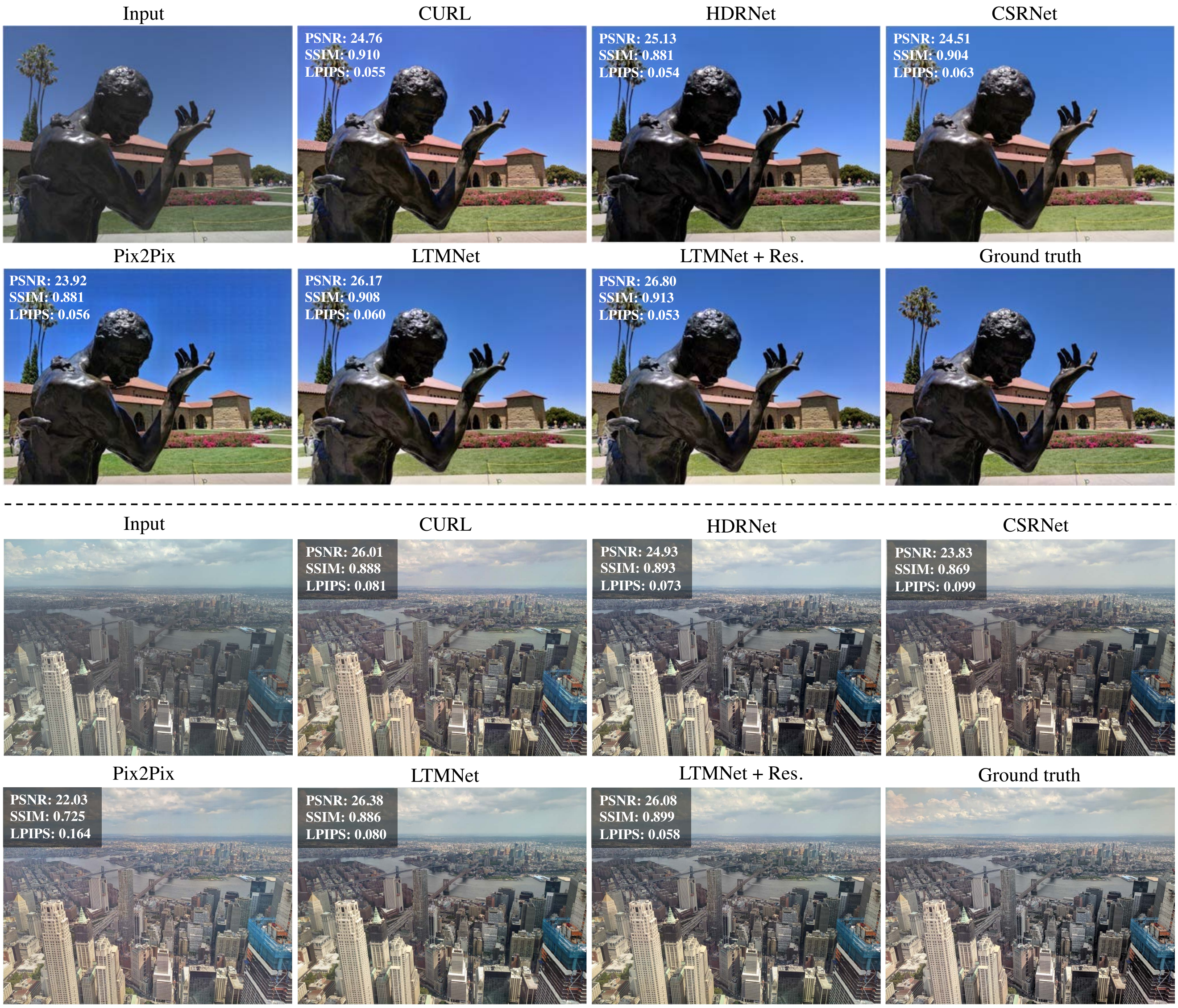


FIGURE 10. Visual comparison of our LTMNet against SOTA methods: CURL [2], HDRNet [25], CSRNet [1], Pix2Pix [64], and Zero-DCE [48], on the HDR+ [60] dataset. Our LTMNet produces visually enhanced results while avoiding structural and color artifacts.

curves by the same scaling factors, such that:

$$\mathbf{s} \odot \tilde{\mathbf{t}}_{ij} \equiv \sum_{k=1}^{4} w_{ijk}(\mathbf{s} \odot \mathbf{t}_k) \tag{11}$$

Finally, the edited image $\tilde{\mathbf{y}}$ is obtained by applying $\mathrm{Interp}(\mathbf{x}, \tilde{\mathcal{T}})$ to the tone curve set $\tilde{\mathcal{T}}$ that contains edited tone curves.

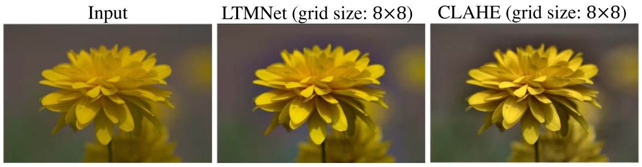


FIGURE 11. A failure example. The image contains an object in need of a significantly different transformation function from its neighbouring regions.

VI. EVALUATION ON MIT-ADOBE FiveK DATASET

In Section IV, we have discussed the lack of local processing in the MIT-Adobe FiveK [59] dataset. We provide additional evidence by comparing our local tone mapping model with a global tone mapping model trained on MIT-Adobe FiveK. Results are shown in Table 3. The local tone mapping (LTM) model has grid size 8 × 8. The global tone mapping (GTM) model has grid size 1 × 1. The GTM+LTM model predicts both an 8 × 8 grid of local tone curves and a global tone curve, with the local tone curves applied after the globally tone-mapped image. The quantitative results indicate that performance increases when the model architecture enables more global tone mapping effects, which suggests that the MIT-Adobe FiveK dataset is better modeled by global, rather than local, transformations, and thus is unsuitable for our local tone mapping task.

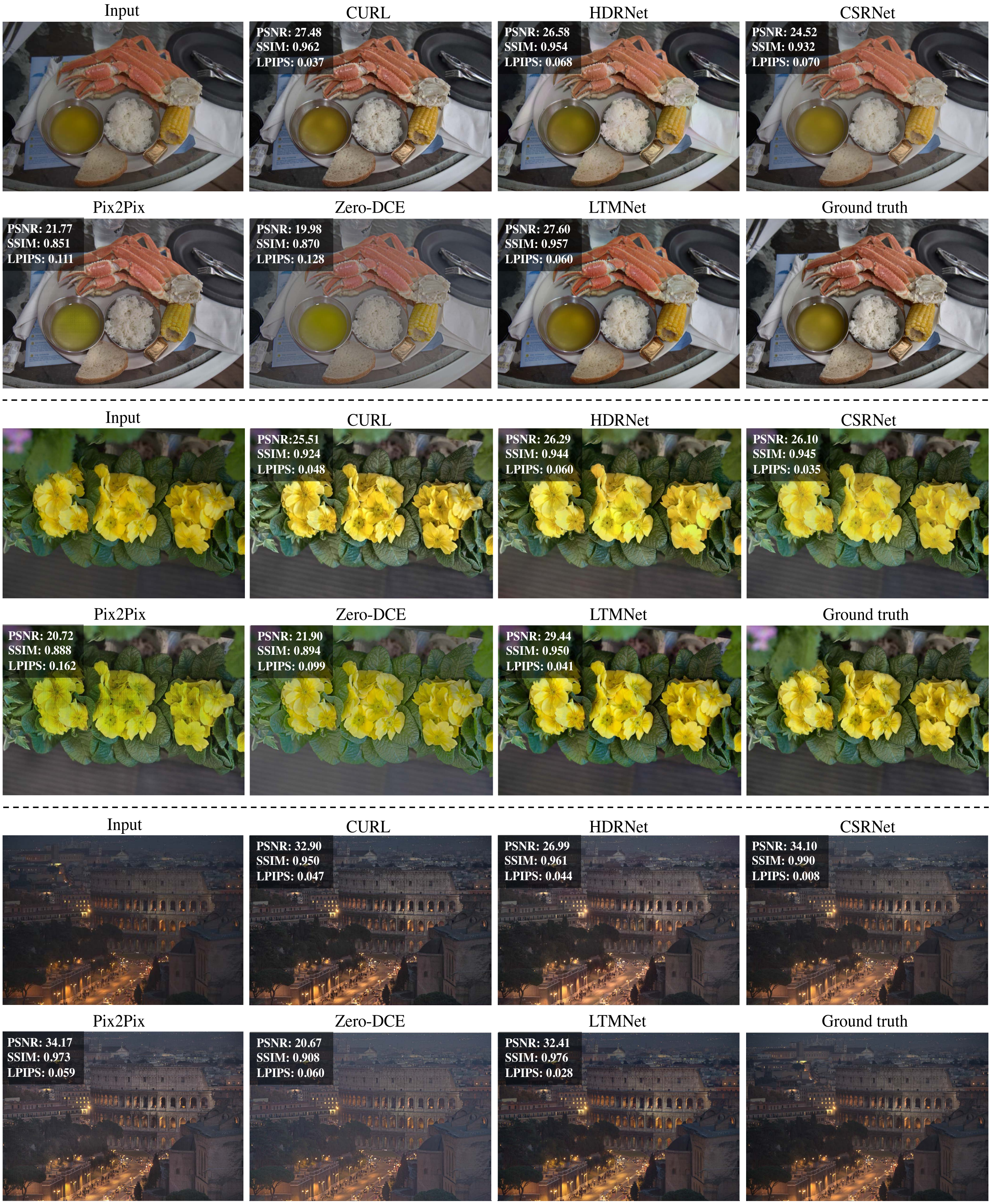


FIGURE 12. Visual comparison of our LTMNet against SOTA methods: CURL [2], HDRNet [25], CSRNet [1], Pix2Pix [64], and Zero-DCE [48], on our LTM dataset. Our LTMNet produces visually enhanced results while avoiding structural and color artifacts.

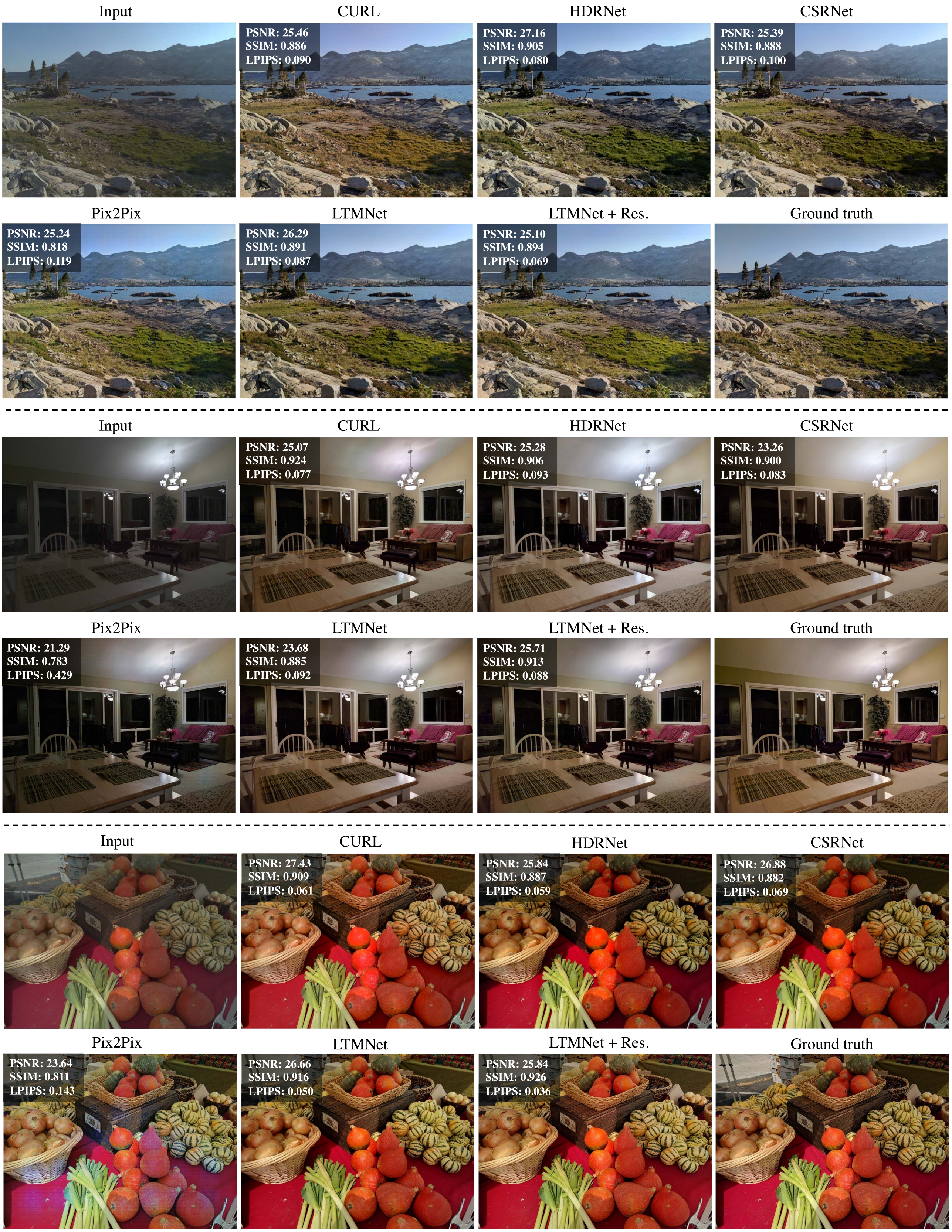


FIGURE 13. Visual comparison of our LTMNet against SOTA methods: CURL [2], HDRNet [25], CSRNet [1], and Pix2Pix [64], on the HDR+ dataset. Our LTMNet produces visually enhanced results while avoiding structural and color artifacts.

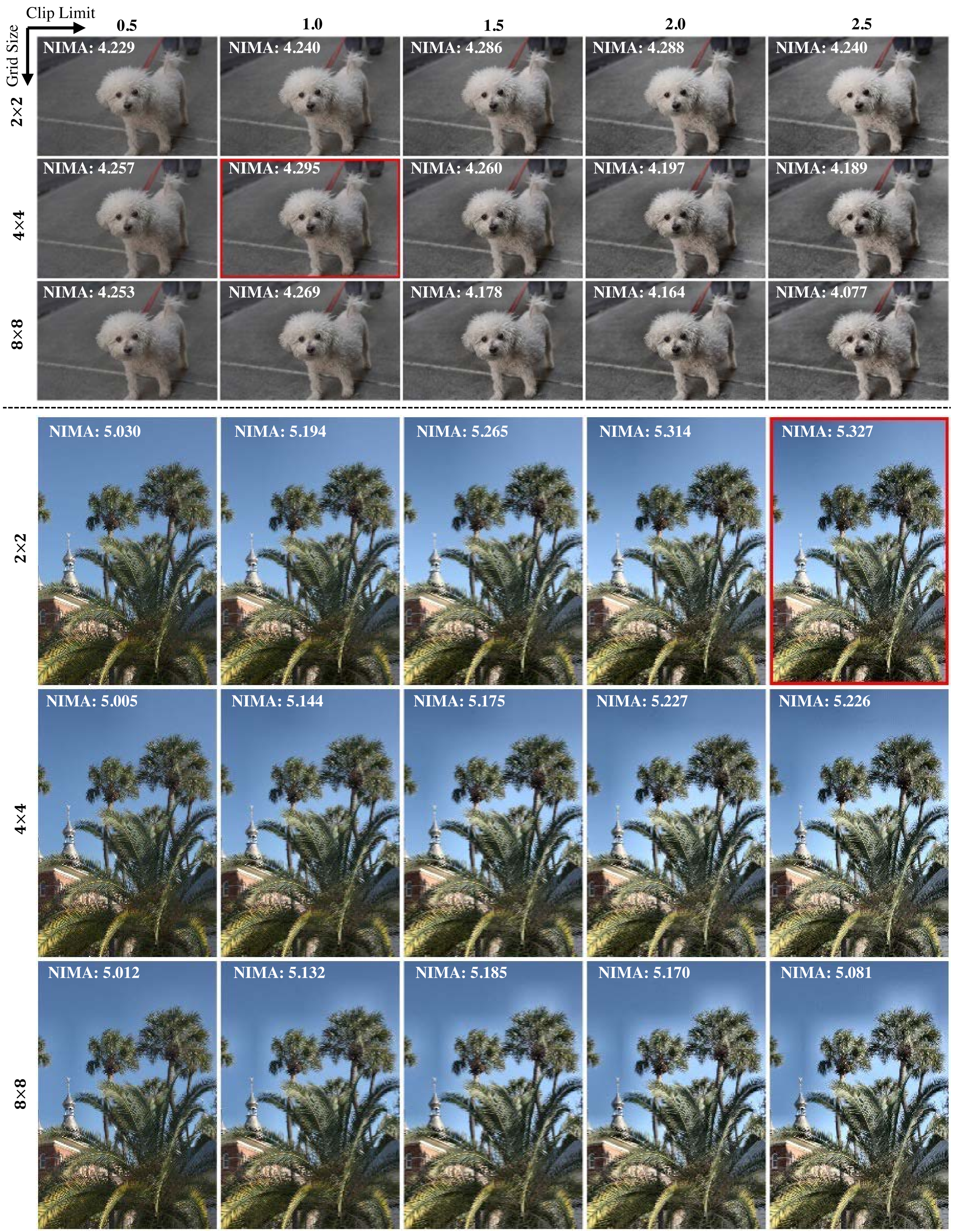


FIGURE 14. Example of all 15 versions of an image enhanced by CLAHE [9]. CLAHE requires careful parameter tuning and not all parameter combinations produce high-quality results. We use a non-reference metric, NIMA [63], to automatically select the CLAHE parameters that give the most visually pleasing version of an enhanced image. The version with the highest NIMA score is highlighted in red. NIMA is able to select images without halo artifacts.

TABLE 3. Results for global vs. local tone mapping on the MIT-Adobe FiveK dataset.

Model	PSNR↑	SSIM↑	LPIPS↓
LTM	22.55	0.903	0.071
GTM+LTM	23.85	0.902	0.076
GTM	24.27	0.913	0.068

TABLE 4. Quantitative comparison between our method (LTMNet) and SOTA methods on the MIT-Adobe FiveK dataset.

Method	PSNR↑	SSIM↑	LPIPS↓
CURL [2]	24.19	0.892	0.079
DeepUPE [42]	23.40	0.875	0.093
CSRNet [1]	23.64	0.906	0.069
Zero-DCE [48]	16.85	0.796	0.202
LTMNet (grid 1x1)	**24.27**	**0.913**	**0.068**
LTMNet (grid 8x8)	22.55	0.903	0.071

TABLE 5. Ablation studies for using less control points (CPs) on the MIT-Adobe FiveK dataset. All experiments are performed with grid size 8 × 8.

Control Points	PSNR↑	SSIM↑	LPIPS↓
LTMNet (256 CPs)	**22.55**	**0.903**	**0.071**
LTMNet (128 CPs)	22.07	0.897	0.074
LTMNet (64 CPs)	22.18	0.898	0.072
LTMNet (32 CPs)	22.14	0.896	0.073
LTMNet (16 CPs)	21.99	0.894	0.076

Despite being not well suited for our task, for completeness, we still evaluated on MIT-Adobe FiveK and the results are shown in Table 4. We used 1,000/100/500 images for training/validation/testing. LTMNet with a 1 × 1 grid (i.e., GTM) outperforms other methods; with a 8 × 8 grid (i.e., LTM), performance on PSNR is worse because, as mentioned, MIT-Adobe contains mostly GTM images. However, there is no significant decrease in perceptual metrics (a 0.01 difference in SSIM), which indicates that a finer grid can still model global operations.

VII. LIMITATIONS AND FUTURE WORK

Our method can experience halo artifacts when the input image has a foreground object that is transformed by a drastically different function from the background scene. This is a result of pixels close to the object boundaries receiving influence from two different transfer functions. As illustrated in Fig. 11, background pixels close to the flower are interpolated by both the tone curves predicted for the gray background and the tone curves predicted for the yellow flower. Influence from the flower's tone curves results in a dark halo. This is a limitation of CLAHE [9] as well, which uses the same interpolation scheme. This issue may be addressed by semantic segmentation, which separates prominent objects in a scene so that each segment has its own transformation functions, unaffected by neighboring segments. Furthermore, we may learn different grid sizes for each segment, so that homogeneous segments are assigned a smaller grid to reduce the amount of spatial variation, and textured segments are assigned a larger grid size to leverage more expressive local enhancements.

Another potential future direction is to condition our network on tunable parameters to allow both automatic enhancements and manual tuning. Although the output images from our method can be adjusted by post-editing the predicted tone curves, our network itself is fully automatic. This poses challenges if the user would like to make customized adaptations to the neural network based on personal preferences—for example, tuning a few parameters so that the network consistently produces different styles for different scene categories. We would like to investigate strategies that tackle these challenges in our future work.

VIII. CONCLUSION

We proposed LTMNet, a method for local image enhancement that learns a grid of local tone curves. LTMNet enhances local image regions more effectively compared with global transformations and offers higher interpretability than pixel-wise methods. LTMNet outperforms existing methods in local tone mapping and achieves competitive results in modeling additional photofinishing operations. In addition, we proposed a new dataset representative of local tone mapping (LTM dataset) that, unlike existing datasets, represents only global and local tone mapping. Our method is quite advantageous in that it can be easily integrated into both camera ISPs and user-interactive image editing tools.

APPENDIX A NIMA FOR LTM DATASET PREPARATION

We performed a user study to compare NIMA [63] with three other commonly used non-reference metrics: BRISQUE [67], NIQE [68], and PIQE [69]. We randomly selected 100 images from our dataset. For each image, we produced 15 CLAHE versions and selected the best version using all four metrics (NIMA, BRISQUE, NIQE, and PIQE). We asked 40 users to select the image they prefer from the four "best" versions. The average user preference (i.e., the percentage of time one image version is preferred over the others) is shown in Table 6. The results are statistically significant; applying the ANOVA test, we obtain $F_{3,39} = 10.33$ and $p < 0.0001$. The results indicate that NIMA aligns with user preference better than other metrics. The NIMA paper and other works [70], [71] confirm that NIMA aligns with perceptual quality well. We have obtained informed consent for the user study.

TABLE 6. Average user preference over 40 users, 100 images. Results are statistically significant (ANOVA, $F_{3,39} = 10.33$, $p < 0.0001$).

Method	NIMA	BRISQUE	NIQE	PIQE
Avg. user preference	**0.299**	0.224	0.259	0.218
Standard deviation	0.022	0.039	0.016	0.051

Fig. 14 provides more qualitative examples of NIMA's selection of the best image version over 15 parameter combinations.

APPENDIX B ADDITIONAL VISUAL COMPARISONS

Fig. 12 and 13 showcase more qualitative comparisons between our method and SOTA methods, on our LTM dataset and the HDR+ dataset [60] respectively.

REFERENCES

[1] J. He, Y. Liu, Y. Qiao, and C. Dong, "Conditional sequential modulation for efficient global image retouching," in *Proc. ECCV*, 2020, pp. 679–695.

[2] S. Moran, S. McDonagh, and G. Slabaugh, "CURL: Neural curve layers for global image enhancement," in *Proc. 25th Int. Conf. Pattern Recognit. (ICPR)*, Jan. 2021, pp. 9796–9803.

[3] K. Panetta, L. Kezebou, V. Oludare, S. Agaian, and Z. Xia, "TMO-Net: A parameter-free tone mapping operator using generative adversarial network, and performance benchmarking on large scale HDR dataset," *IEEE Access*, vol. 9, pp. 39500–39517, 2021.

[4] C. Guo and X. Jiang, "Deep tone-mapping operator using image quality assessment inspired semi-supervised learning," *IEEE Access*, vol. 9, pp. 73873–73889, 2021.

[5] I. R. Khan, W. Aziz, and S.-O. Shim, "Tone-mapping using perceptual-quantizer and image histogram," *IEEE Access*, vol. 8, pp. 31350–31358, 2020.

[6] 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.

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

[8] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 4th ed. London, U.K.: Pearson, 2018.

[9] S. M. Pizer, E. P. Amburn, J. D. Austin, R. Cromartie, A. Geselowitz, T. Greer, B. ter Haar Romeny, J. B. Zimmerman, and K. Zuiderveld, "Adaptive histogram equalization and its variations," *Comput. Vis., Graph., Image Process.*, vol. 39, no. 3, pp. 355–368, 1987.

[10] T. Arici, S. Dikbas, and Y. Altunbasak, "A histogram modification framework and its application for image contrast enhancement," *IEEE Trans. Image Process.*, vol. 18, no. 9, pp. 1921–1935, Sep. 2009.

[11] J. McVey and G. Finlayson, "Towards a generic neural network architecture for approximating tone mapping algorithms," *London Imag. Meeting*, vol. 2, no. 1, pp. 93–96, Sep. 2021. [Online]. Available: https://www.ingentaconnect.com/content/ist/lim/2021/00002021/00000001/a%rt00018

[12] S. Bianco, C. Cusano, F. Piccoli, and R. Schettini, "Personalized image enhancement using neural spline color transforms," *IEEE Trans. Image Process.*, vol. 29, pp. 6223–6236, 2020.

[13] Y. Hu, H. He, C. Xu, B. Wang, and S. Lin, "Exposure: A white-box photo post-processing framework," *ACM Trans. Graph.*, vol. 37, no. 2, pp. 1–17, 2018.

[14] J. Park, J.-Y. Lee, D. Yoo, and I. S. Kweon, "Distort-and-recover: Color enhancement using deep reinforcement learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 5928–5936.

[15] S. Kosugi and T. Yamasaki, "Unpaired image enhancement featuring reinforcement-learning-controlled image editing software," in *Proc. AAAI*, 2020, pp. 11296–11303.

[16] Y.-T. Kim, "Contrast enhancement using brightness preserving bi-histogram equalization," *IEEE Trans. Consum. Electron.*, vol. 43, no. 1, pp. 1–8, Feb. 1997.

[17] Y. Wang, Q. Chen, and B. Zhang, "Image enhancement based on equal area dualistic sub-image histogram equalization method," *IEEE Trans. Consum. Electron.*, vol. 45, no. 1, pp. 68–75, Feb. 1999.

[18] J. A. Stark, "Adaptive image contrast enhancement using generalizations of histogram equalization," *IEEE Trans. Image Process.*, vol. 9, no. 5, pp. 889–896, May 2000.

[19] C. Lee, C. Lee, and C.-S. Kim, "Contrast enhancement based on layered difference representation of 2D histograms," *IEEE Trans. Image Process.*, vol. 22, no. 12, pp. 5372–5384, Dec. 2013.

[20] D. J. Ketcham, R. W. Lowe, and J. W. Weber, "Image enhancement techniques for cockpit displays," Display Syst. Lab., Hughes Aircr. Company, Culver City, CA, USA, Tech. Rep. P74-530R/D0802, 1974.

[21] R. Hummel, "Image enhancement by histogram transformation," *Comput. Graph. Image Process.*, vol. 6, no. 2, pp. 184–195, 1977. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0146664X77800117

[22] H. Chang, O. Fried, Y. Liu, S. DiVerdi, and A. Finkelstein, "Palette-based photo recoloring," *ACM Trans. Graph. (TOG)*, vol. 34, no. 4, pp. 1–139, 2015.

[23] J. Tan, J. Echevarria, and Y. Gingold, "Efficient palette-based decomposition and recoloring of images via RGBXY-space geometry," *ACM Trans. Graph.*, vol. 37, no. 6, pp. 1–10, Dec. 2018.

[24] 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. 4459–4468.

[25] 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, Jul. 2017.

[26] J. Chen, S. Paris, and F. Durand, "Real-time edge-aware image processing with the bilateral grid," *ACM Trans. Graph.*, vol. 26, no. 3, p. 103, Jul. 2007.

[27] J. Chen, A. Adams, N. Wadhwa, and S. W. Hasinoff, "Bilateral guided upsampling," *ACM Trans. Graph.*, vol. 35, no. 6, pp. 1–8, 2016.

[28] Y. Song, H. Qian, and X. Du, "StarEnhancer: Learning real-time and style-aware image enhancement," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 4126–4135.

[29] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Proc. 6th Int. Conf. Comput. Vis.*, Jan. 1998, pp. 839–846.

[30] M. Aubry, S. Paris, S. W. Hasinoff, J. Kautz, and F. Durand, "Fast local Laplacian filters: Theory and applications," *ACM Trans. Graph.*, vol. 33, no. 5, pp. 1–14, Sep. 2014.

[31] K. He, J. Sun, and X. Tang, "Guided image filtering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 6, pp. 1397–1409, Jun. 2013.

[32] L. Yu, H. Su, and C. Jung, "Perceptually optimized enhancement of contrast and color in images," *IEEE Access*, vol. 6, pp. 36132–36142, 2018.

[33] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. MICCAI*, 2015, pp. 234–241.

[34] 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.

[35] A. Ignatov, N. Kobyshev, R. Timofte, and K. Vanhoey, "DSLR-quality photos on mobile devices with deep convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 3277–3285.

[36] A. Ignatov, N. Kobyshev, R. Timofte, K. Vanhoey, and L. Van Gool, "WESPE: Weakly supervised photo enhancer for digital cameras," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2018, pp. 691–700.

[37] 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.

[38] Y. Deng, C. C. Loy, and X. Tang, "Aesthetic-driven image enhancement by adversarial learning," in *Proc. 26th ACM Int. Conf. Multimedia*, Oct. 2018, pp. 870–878.

[39] H.-U. Kim, Y. J. Koh, and C.-S. Kim, "PieNet: Personalized image enhancement network," in *Proc. ECCV*, Aug. 2020, pp. 374–390.

[40] S. Bianco, C. Cusano, F. Piccoli, and R. Schettini, "Content-preserving tone adjustment for image enhancement," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2019, pp. 1936–1943.

[41] K. Ko and C.-S. Kim, "IceNet for interactive contrast enhancement," *IEEE Access*, vol. 9, pp. 168342–168354, 2021.

[42] 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. 6849–6857.

[43] 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.

[44] M. Afifi, K. G. Derpanis, B. Ommer, and M. S. Brown, "Learning multi-scale photo exposure correction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 9157–9167.

[45] 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.

[46] 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. 2181–2290.

[47] 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. 3063–3072.

[48] 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. 1780–1789.

[49] E. H. Land, "The retinex theory of color vision," *Sci. Amer.*, vol. 237, no. 6, pp. 108–129, Dec. 1977.

[50] X. Fu, D. Zeng, Y. Huang, X.-P. Zhang, and X. Ding, "A weighted variational model for simultaneous reflectance and illumination estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2782–2790.

[51] Q. Zhang, G. Yuan, C. Xiao, L. Zhu, and W.-S. Zheng, "High-quality exposure correction of underexposed photos," in *Proc. 26th ACM Int. Conf. Multimedia*, Oct. 2018, pp. 582–590.

[52] Z. Ying, G. Li, Y. Ren, R. Wang, and W. Wang, "A new low-light image enhancement algorithm using camera response model," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops (ICCVW)*, Oct. 2017, pp. 3015–3022.

[53] Z. Zhang, Y. Jiang, J. Jiang, X. Wang, P. Luo, and J. Gu, "STAR: A structure-aware lightweight transformer for real-time image enhancement," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 4086–4095.

[54] 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.

[55] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. ICLR*, 2015, pp. 1–14.

[56] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015.

[57] L. A. Gatys, A. S. Ecker, and M. Bethge, "A neural algorithm of artistic style," *J. Vis.*, vol. 16, no. 12, p. 326, 2016.

[58] V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *Proc. ICML*, 2010, pp. 807–814.

[59] 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.

[60] S. W. Hasinoff, D. Sharlet, R. Geiss, A. Adams, J. T. Barron, F. Kainz, J. Chen, and M. Levoy, "Burst photography for high dynamic range and low-light imaging on mobile cameras," *ACM Trans. Graph.*, vol. 35, no. 6, pp. 1–12, Nov. 2016.

[61] Q. Gao and X. Wu, "Real-time deep image retouching based on learnt semantics dependent global transforms," *IEEE Trans. Image Process.*, vol. 30, pp. 7378–7390, 2021.

[62] A. Abdelhamed, S. Lin, and M. S. Brown, "A high-quality denoising dataset for smartphone cameras," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1692–1700.

[63] H. Talebi and P. Milanfar, "NIMA: Neural image assessment," *IEEE Trans. Image Process.*, vol. 27, no. 8, pp. 3998–4011, Aug. 2018.

[64] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1125–1134.

[65] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.

[66] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. ICLR*, Dec. 2014, pp. 1–15.

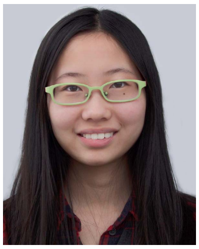
[67] A. Mittal, A. K. Moorthy, and A. C. Bovik, "Blind/Referenceless image spatial quality evaluator," in *Proc. Conf. Rec. 45th Asilomar Conf. Signals, Syst. Comput. (ASILOMAR)*, Nov. 2011, pp. 723–727.

[68] A. Mittal, R. Soundararajan, and A. C. Bovik, "Making a 'completely blind' image quality analyzer," *IEEE Signal Process. Lett.*, vol. 20, no. 3, pp. 209–212, Mar. 2012.

[69] N. Venkatanath, D. Praneeth, M. C. Bh, S. S. Channappayya, and S. S. Medasani, "Blind image quality evaluation using perception based features," in *Proc. Twenty 1st Nat. Conf. Commun. (NCC)*, Feb. 2015, pp. 1–6.

[70] V. Khrulkov and A. Babenko, "Neural side-by-side: Predicting human preferences for no-reference super-resolution evaluation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 4988–4997.

[71] A. Aakerberg, K. Nasrollahi, and T. B. Moeslund, "Real-world super-resolution of face-images from surveillance cameras," 2021, *arXiv:2102.03113*.



LUXI ZHAO received the B.A.S. degree in computer engineering from The University of British Columbia, in 2020, and the M.Sc. degree in applied computing from the University of Toronto, in 2022. She is currently a Machine Learning Engineer at Samsung AI Center-Toronto. Her research interests include computer vision, computational photography, and machine learning.

ABDELRAHMAN ABDELHAMED received the M.Sc. degrees from the National University of Singapore, Singapore, and Assiut University, Egypt, and the Ph.D. degree in computer science from York University, supervised by Prof. Michael S. Brown. He is currently a Research Scientist at Samsung AI Center-Toronto. His research interests include computer vision, computational imaging, and machine learning.

MICHAEL S. BROWN is currently a Professor and the Canada Research Chair in computer vision at York University, Toronto, Canada. He was also a part-time Senior Research Director at the Samsung AI Center-Toronto. His research interests include computer vision, image processing, and computer graphics. He has served as the Program Chair for WACV 2011/17/19 and 3DV 2015 and the General Chair for ACCV 2014 and CVPR 2018/21.