

Leveraging Color Naming for Image Enhancement

David Serrano-Lozano, Luis Herranz, Michael S. Brown, Javier Vazquez-Corral

Abstract—Enhancing images to make them visually appealing is a persistent challenge in computer vision. Many deep-learning methods train models on paired datasets to replicate expert editing styles. However, these approaches struggle with two key issues: (1) interpretability and (2) a parametrization suitable for user adjustments. To address these challenges, we present NamedCurves+, an approach inspired by the concept of Color Naming, a universal set of familiar colors widely used in software tools for intuitive editing. Our method integrates color names into a learning-based framework, enabling global adjustments for each named color through tone curves. To address local image variations, we incorporate a transformer block that captures spatial dependencies, enabling context-aware edits across the image. NamedCurves+ enhances the retouching process's interpretability and supports user interaction, allowing flexible modifications of individual tone curves to refine the retouched image according to personal preferences. Extensive experiments on tasks such as image retouching, tone mapping, and exposure correction demonstrate that NamedCurves+ outperforms state-of-the-art methods. Notably, our approach is both explainable, as the tone curves explicitly represent how each color name contributes to the enhancement, and interactive, allowing users to customize the retouching process and achieve results tailored to their liking. Source code and models will be publicly available at: <https://namedcurves.github.io>.

Index Terms—Image Enhancement, Image Retouching, Color Editing, Color Naming

I. INTRODUCTION

COLOR plays a vital role in photography, enhancing focal points, evoking emotions, and enriching storytelling. Whether through vibrant hues or subtle tones, understanding the importance of colors is crucial for photographers seeking to elicit specific responses. Despite significant advancements in camera technology, both amateur and professional photographers often resort to post-capture image enhancement to improve image quality. However, manual image editing poses challenges for those lacking expertise, time, or a well-developed aesthetic sense.

A potential solution to avoid manual adjustment lies in learning-based methods, where deep neural networks (DNN) model the editing style of skilled photographers or colorists. These models typically leverage paired datasets of raw and expert-retouched images. While effective in many scenarios, such approaches have critical limitations. Expert aesthetic sense may not always align with individual preferences or specific use cases, leaving certain users dissatisfied with the retouched results. Furthermore, existing methods often rely

on approaches such as weighting multiple 3D Look-Up Tables [1]–[4], or employing end-to-end U-Net-like architectures [5]–[7], which offer limited interpretability and prevent user interaction. As a result, users cannot easily adjust the output to suit their specific needs. Compounding the issue, most image-retouching datasets are captured in controlled environments by skilled professionals, while real-world images are often taken by amateurs under varying lighting conditions. This creates significant variability in the amount of light reaching the camera sensor, further complicating the enhancement process. To bridge the gap between the intensive manual adjustments required by software tools and the limited controllability of current automated methods, we propose an approach that mimics the expert's style while also allowing users to make subtle adjustments based on a predefined set of color curves.

It is interesting to consider the tools provided to the artists for performing the image editing. Many photo editing software applications (e.g., Adobe Photoshop [8]) provide users with the ability to manipulate the image based on a small set of fixed colors (e.g., red, green, yellow, orange, blue, purple). Interestingly, the predefined colors selected by software tools are similar to those psychologists and linguists have found to be universal across languages [9], a research topic often referred to as *color naming*.

This paper proposes leveraging pixel-wise color naming decomposition for image enhancement. Specifically, we introduce NamedCurves+, a learning-based framework comprising three main components. First, we employ a backbone model to standardize the input image into a canonical space. Next, using color naming, the image is decomposed into six color maps, each associated with a distinct set of Bezier-parametrized global tone curves. Finally, a transformer block combines the globally adjusted images to model local manipulations. Figure 1 (a) illustrates the framework of our approach. By associating specific tone curves with each color name, NamedCurves+ offers both interpretability and interactivity. Interpretability is achieved thanks to the computation of different curves for each color name. Interactivity because users can directly manipulate the curves to alter a specific color, tailoring the output to their preferences. In detail, by modifying the tone curves of each color term, the user can obtain different outputs as the model enhances each color region differently. See Figure 1 (b) and (c) to see different results when editing the pink-purple and green tone curves, respectively. We evaluate NamedCurves+ on multiple tasks, including image retouching, tone mapping, and exposure correction, using benchmark datasets such as MIT-Adobe-5K [10], PPR10K [11], MSEC [12], and SICE [13]. Our method achieves state-of-the-art performance both quantitatively and qualitatively, outperforming existing approaches.

This paper was produced by the IEEE Publication Technology Group. They are in Piscataway, NJ.

Manuscript received April 19, 2021; revised August 16, 2021.

DSL and JVC are with the Computer Vision Center, Barcelona, Spain and the Universitat Autònoma de Barcelona, Spain. LH is with the Universidad Politécnica de Madrid, Spain. MSB is with the York University, Toronto, Canada.

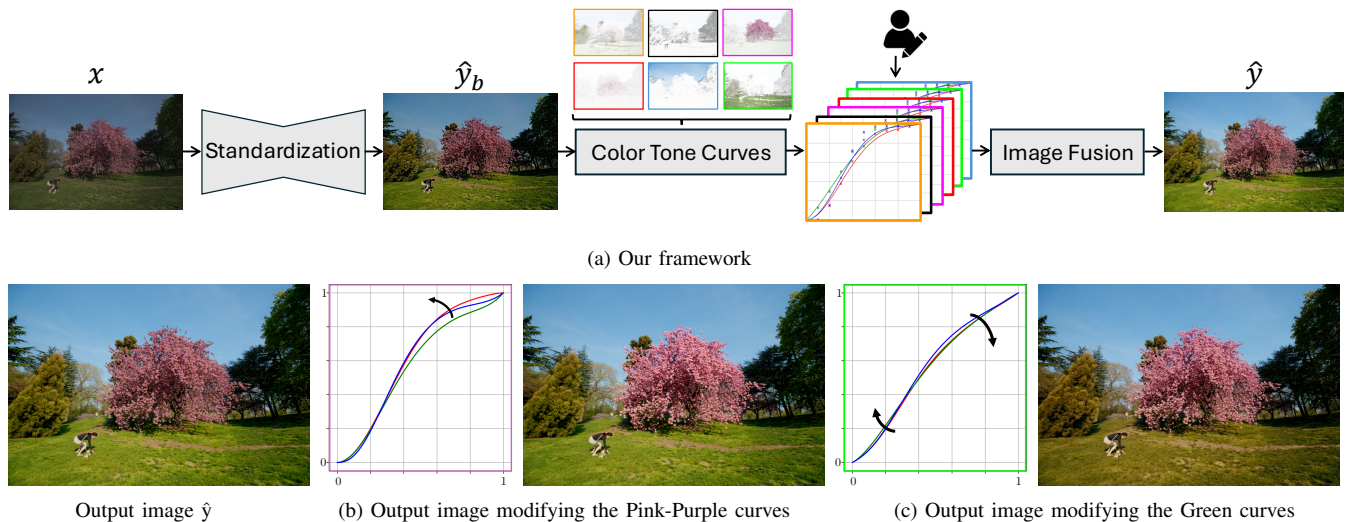


Fig. 1. Our framework (a) enhances images to achieve visually appealing results. Starting with the input image x , the image is first standardized by a UNet-like backbone y_b . Next, the framework estimates a set of tone curves corresponding to each color name probability map. These tone curves are applied to generate intermediate images, which are then fused by a transformer block to produce the final enhanced image $\hat{y}$. NamedCurves+ supports user interaction, allowing flexible adjustments to global tone curves for each color name. In (b), modifying the highlights section of the Pink-Purple curve alters the pink tree, while in (c), reducing the shadows and highlights of the Green curves shifts green regions to a more brownish tone. The arrows indicate the direction of the curve adjustments. Note that the curve adjustments modify the initial enhanced version $\hat{y}$ without altering the model weights, which remain frozen.

This work extends our previous conference paper [14], where we introduced a learning-based method that leverages color naming and tone curves for end-to-end image retouching. In this extended version, we present NamedCurves+ as a more general image enhancement framework with improved performance and efficiency across multiple tasks, including tone mapping and exposure correction. We also strengthen interpretability and interactivity by decomposing the image into a set of predefined color names and a color-conditioned tone curves that users can directly edit to steer the final result without retraining. In particular, we make the following contributions: (i) we redesign the image fusion component using an efficient transformer block, improving quality and reducing artifacts while maintaining fast inference; (ii) we extend the framework beyond retouching to tone mapping and exposure correction, achieving state-of-the-art results on standard benchmarks; and (iii) we demonstrate user control by modifying the tone curves associated with each color name to obtain customized enhancements. Finally, we address a key limitation of our previous architecture: its fusion mechanism could be semantically brittle when color-term probabilities overlap or change smoothly, leading to halo artifacts at color boundaries. Figure 2 highlights two representative failure cases of NamedCurves: in the first example, haloing appears near the top corners when pixels have high probability for multiple color terms, and in the second example, inconsistencies are visible in the cloud region. By contrast, NamedCurves+ mitigates these artifacts through the proposed end-to-end fusion design, producing cleaner and more visually consistent enhanced images.

II. RELATED WORK

Color Naming. Color naming is crucial for product design, photography, and vision research [15]–[19]. Berlin and Kay [9] conducted a study on the basic color lexicon across various

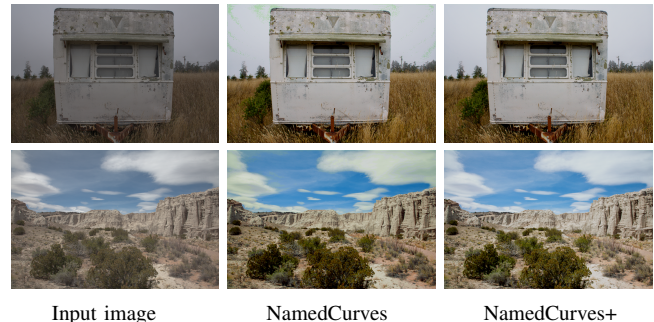


Fig. 2. Examples where NamedCurves struggled. It generates halo artifacts in color boundaries and brittleness when color naming probabilities change drastically.

languages and discovered universal semantics. Their seminal analysis showed that the evolution of basic color vocabularies is influenced by visual physiology, which limits the possible composite categories to a small number of those. The 11 color names found that most societies and cultures share are: *orange, brown, yellow, white, grey, black, pink, purple, red, green and blue*.

Following Berlin and Kay’s research, different studies (e.g., [20]–[23]) aimed at predicting the boundaries between each color name. For example, Figure 3 shows the standard Munsell color array using Van de Weijer et al. [20] color classification based on color naming.

These methods work as follows. Given an RGB value in the sRGB color space, color naming methods produce an 11-d vector corresponding to the probability of the RGB value belonging to the specific color names listed above. This is visualized in Figure 4, where we show an original image and the 11-probability maps coded with a color map to aid visualization.

Learned Image Enhancement. The need to provide users with tools to allow easy image enhancement has grown signif-

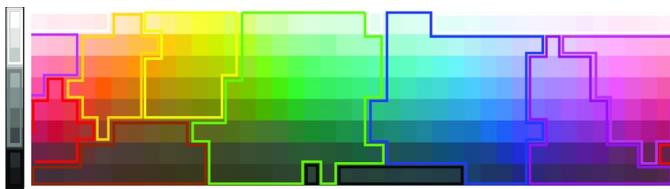


Fig. 3. Van de Weijer et al. [20] color names grouped in the Munsell color array. The color names are *white, grey, black, red, brown, orange, yellow, green, blue, purple and pink*.

icantly due to the ease of smartphone photo-taking. Initially, histogram equalization was a primary method for enhancing contrast in images [24], [25]. Subsequently, local operators [26], [27] and color correction techniques based on color constancy [28] were introduced. Since the introduction of the MIT-Adobe-5K dataset by Bychkovsky et al. [10], which contains 5,000 images retouched by five experts, data-driven methods have emerged as one of the preferred means to improve image quality.

Lookup tables (LUTs) represent a widely used method for image manipulation, typically manually tuned and fixed in camera imaging pipelines or photo editing tools. Zeng et al. [1] proposed 3DLUT, a method to learn these 3D LUTs from annotated data with a small convolutional neural network. Building upon this, Yang et al. [2] proposed a mechanism to achieve a more flexible sampling by learning the non-uniform sampling intervals, Yang et al. [3] proposed mixing 1DLUTs and 3DLUTs to obtain a good trade off between memory footprint and transformation capabilities, and Kim et al. [4] proposed including bilateral grids to incorporate spatial information. Wang et al. [29] also presented a modification of 3DLUT that incorporates spatial data extracted with a shallow UNet architecture.

Conversely, image-to-image methods estimate directly a mapping to modify the input images without intermediate steps. Generative adversarial networks (GANs) are frequently employed for such tasks. Chen et al. [5], Ni et al. [6], and Jiang et al. [7] proposed unpaired learning schemes using single GANs to estimate enhanced versions of input images directly.

However, the group of methods closer to our approach are the ones that involve estimating intermediate or physical parameters. DeepLPF [30] and RsfNet [31], inspired by retouching software tools, estimate elliptical, gradual, and polynomial filters for local image editing and region-specific filters, respectively. They showed promising results, but editing only locally the images introduced artifacts into the final image. Guo et al. [32] proposed Zero-DCE, the first method to formulate low-light image enhancement as a curve estimation problem. Their deep network estimates pixel-wise curves to modify the dynamic range of input images. This groundbreaking work influenced subsequent methods, such as CURL [33], which estimates piecewise linear curves for HSV, RGB, and CIELab color spaces; FlexiCurve [34], which estimates sets of piecewise curves and blends them via a Transformer, Bianco et al. [35] which estimates the global curves to retouch an image, and LTMNet [36] that learns a grid of tone curves to both locally and globally enhance an image. Many previous

methods [1]–[4] have also been evaluated on learned tone mapping using the widely used MIT-Adobe-5K dataset [10]. In contrast to image retouching, the images are in 16-bit CIE XYZ format and the outputs are sRGB images.

As in previous methods [33]–[36], we use tone curves to manipulate images. Our main novelty is that we leverage color naming decomposition to disentangle distinct colors and later employ a transformer-based fusion scheme to emulate the image editing style of an expert. While previous methods may appear to offer interpretability through tone curves, this is often not the case. Manipulating images in different spaces or concatenating various tone curve sets undermines interpretability. In contrast, our approach disentangles different color names, enabling a clear interpretation of how the model manipulates each color in the image.

Learned Exposure Correction. Incorrect exposure remains a significant source of errors in camera-based imaging, arising from excessively long exposure times (overexposure) or excessively short ones (underexposure). Existing methods have proposed various specialized approaches to address exposure correction. For instance, Afifi et al. [12] formulate the problem as a combination of color and detail enhancement using a two-step DNN. Nsambi et al. [37] introduce a deep feature matching loss to enable exposure-invariant feature representation, while Huang et al. [38] propose lightness reconstruction using two Fourier-based DNNs: one for amplitude and another for phase.

This paper extends from our conference paper [14], where we introduced the use of color naming and tone curves for end-to-end image retouching. NamedCurves+ addresses the broader challenge of developing a generic image enhancement framework. Specifically, we improve the architecture by incorporating a Transformer block to blend the globally adjusted images while modeling the spatial dependencies between color-specific adjustments. We evaluate our framework across three tasks: image retouching, tone mapping, and exposure correction, using four well-established datasets, demonstrating consistent improvements over state-of-the-art methods and our prior approach. Additionally, due to its parameterized design, NamedCurves+ allows users to fine-tune individual color tone curves for personalized control over the enhanced image. Finally, we thoroughly analyze the advantages of color naming, highlighting its role in improving both interpretability and user interaction in image enhancement.

III. METHOD

Figure 1 (a) shows an overview of our proposed method, NamedCurves+. Our method aims to enhance a low-quality RGB input image x , by a learned model that outputs an enhanced version $\hat{y}$. This resulting enhanced image is expected to be close to the expert-retouched image y , based on some objective function L .

Our method consists of four main components, which are detailed in the following sections, including the loss function used for optimizing the framework. The approach first applies a DNN backbone that standardizes the input image into a canonical latent space. Next, we use color naming



Fig. 4. Van de Weijer et al. [20] color naming method applied pixel-wise to the top-left image. The other 11 images show the 11 probability color names maps. Each color is displayed with a different map to aid visualization. Note that some linguistic color names share approximately the same hue and only differ in intensity—e.g., *pink* and *purple*. As tone curves are defined for all the intensity range we group: *orange-brown-yellow*, *pink-purple*, and *white-grey-black*. This grouping is represented by the boxes.

to decompose the image into six color maps. After color naming decomposition, a neural network learns a set of Bezier tone curves to manipulate each color map globally. Finally, a transformer module combines the edited images to achieve local editing effects.

A. Backbone

One challenge faced by learning-based image enhancement methods is that input images, x , can be captured using different cameras with different settings and under different lighting conditions. This may impact our ability for consistent color naming. Similar to the method by Moran et al. [30], we use a UNet-like backbone to standardize the input images.

Our backbone is inspired by LPIENet [39] architecture. We use MobileNet [40] layers (Conv-DWConv-eLU) and a CBAM module [41]—a combination of spatial and channel attention. The backbone consists of three encoder blocks and two decoder blocks connected by multi-resolution skip connections. Each encoder block consists of the following: two MobileNet layers, a CBAM attention block, and a max-pooling layer. The decoder blocks follow the same structure except for the max-pooling layers that are replaced by bilinear upsampling layers. The multi-resolution skip connections consist of three parallel branches of convolutional layers with different dilation rates. Two of the paths consist of two Conv-LeakyReLU blocks to extract local information, while the other path consists of three Conv-LeakyReLU-MaxPooling blocks, an AveragePooling and a LinearLayer to extract global information. As in [33], [42], [43], we found that skip connections at different resolutions improve the performance against backbones with simple skip connections.

B. Color Naming

We aim to decompose the standardized image $\hat{y}_b$ into a set of likelihood color maps to focus different branches of the model. Motivated by the importance of memory colors—the green of the grass or the blue of the sky—in aesthetics [44], [45] we decided to use Color Naming, a perceptually-based color decomposition.

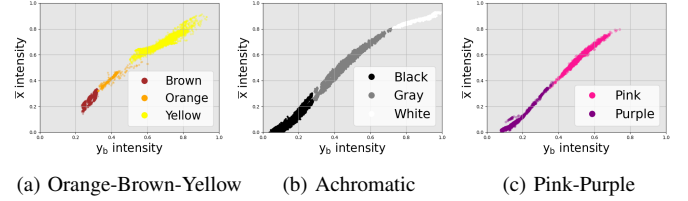


Fig. 5. Grouped color names with respect to the intensity value. Pixels with color name probability greater than 0.5 are plotted.

We used the color naming model from Van de Weijer et al. [20] to obtain the probability maps for each color name. This model inputs an sRGB color value and outputs the probability of this color belonging to each 11 color naming categories, namely *red*, *blue*, *green*, *yellow*, *pink*, *purple*, *orange*, *brown*, *white*, *grey*, *black*. When applied to an image, the model operates for each pixel, which returns a set of probability maps.

We note that some linguistic color names share similar hues but only differ in intensity. For example, orange and brown or pink and purple. As we manipulate the different colors for all intensity ranges, it is beneficial to group these colors together. To this end, we reduce the set of 11 probability maps to just 6 by grouping *orange-brown-yellow*, *pink-purple*, and *white-grey-black* (referring to this last one as *achromatic*). The combined map for these cases is just the addition of the individual maps, and therefore, they are still probabilities (the sum of all the maps for a specific pixel is 1).

In Figure 5, we visually show the aforementioned reason for reducing the number of color naming channels to just 6, i.e. these colors have similar hues at different intensities. We illustrate plots depicting the relationship between input and output intensity values for pixels with the probability of belonging to each specific color greater than 0.5. For example, in the case of orange-brown-yellow, brown is only present at low intensities, while orange dominates at mid-intensities and yellow at top intensities. The same analysis extends to the other joined color channels. Our method aims to learn a curve to be applied at all the intensity levels. Thus, incorporating information spanning all the intensity levels is beneficial. In Figure 4, the color names grouped are shown in boxes.

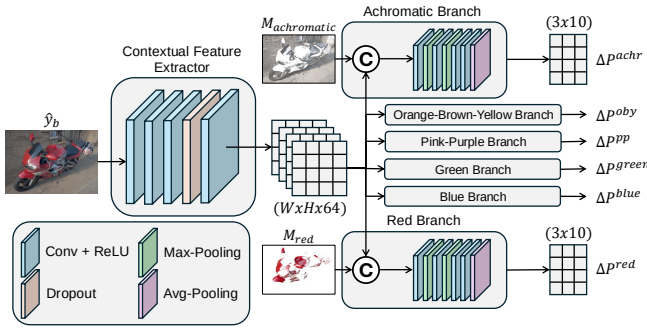


Fig. 6. Estimation of the Bezier tone curves. First, we extract 64-D convolutional feature maps from $\hat{y}_b$ using mainly 4 Conv-ReLU blocks. Then, each specific color name map is concatenated to the feature maps and passed through 4 Conv-ReLU-MaxPooling and a final AveragePooling layer. The module's output is ΔP , and the unnormalized control points increments.

C. Bezier Tone Curves Estimation

Similar to prior works [32]–[34], we leverage tone curves to remap the shadows, midtones, and highlights of the image $\hat{y}_b$ conditioned by the color naming probability maps. Tone curves represent global intensity adjustments from the input level to the output level. The curves are applied pixel-wise in each color channel as 1D Look-Up Tables. Bennet and Finlayson [46] demonstrated that tone adjustments are typically simple curves for a large dataset of enhanced images or can be well-approximated as such. We use Bezier curve parametrization to generate smooth and continuous tone curves from discrete control points.

Specifically, we aim to estimate one global curve for each RGB channel c and color name n . Each curve is parameterized by M control points. We evenly distribute these control points along the input axis, with the first control point fixed at $(0,0)$. Consequently, we only need to estimate the control points' output axis values instead of the two-point coordinates, resulting in $M-1$ parameters per curve. Thus, the Bezier formulation $B^{n,c}$ of a curve can be expressed as:

$$B^{n,c}(i) = \sum_{m=0}^{M-1} P_m^{n,c} \binom{M-1}{m} (1-i)^{(M-1-m)} i^m, \quad (1)$$

where $i \in [0,1]$ is an image channel pixel, $P_m^{n,c}$ denotes the m -th control point for the color name n and channel c , and M is the total number of control points.

Figure 6 illustrates the estimation of the control points that define the Bezier tone curves of an image, comprising two distinct blocks: the contextual feature extractor and 6 color naming branches. The contextual feature extractor primarily consists of 4 Conv-ReLU blocks and a Dropout, while the color naming branches consist of 4 Conv-ReLU-MaxPooling blocks and a final AveragePooling layer to manage the variable sizes of the images.

The contextual feature extractor computes a 64-d convolutional feature map from the standardized image $\hat{y}_b$. Each color naming branch receives as input a concatenation of the 64-d feature maps and the corresponding color naming probability map. The output of the branch for color name n

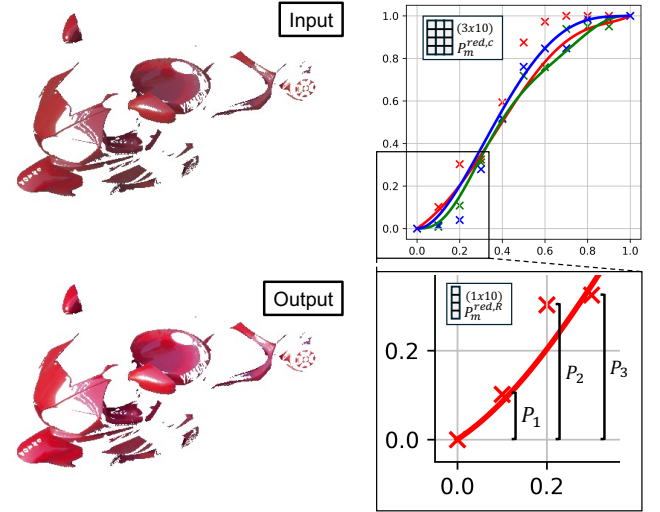


Fig. 7. An example of a Bezier curve (Red branch). The first column shows the input and edited image pixels in which the red color name branch focuses. The center plot shows the tone RGB curves learned for the red color name, while the right plot displays a zoomed-in view of the first four control points of the red channel.

consists of three sets of M increments $(\Delta P_m^{n,c})_{m=1}^M$, each set corresponding to a curve for a given color channel c . These increments $\Delta P_m^{n,c}$ do not directly represent the control points as we impose two different constraints. First, to make the curves monotonically increasing functions, we define $\Delta P_m^{n,c}$ as positive increments relative to the previous point. Second, we normalize $\Delta P_m^{n,c}$, ensuring the total increment between the first and last points is 1 and, thus, placing the last point at $(1,1)$. Consequently, we compute the control points $P_m^{n,c}$ as the accumulated sum of the normalized $\Delta P_m^{n,c}$. This can be formulated as:

$$P_m^{n,c} = \frac{1}{S^{n,c}} \sum_{k=1}^m \Delta P_k^{n,c}, \quad (2)$$

where $S^{n,c} = \sum_{k=1}^M \Delta P_k^{n,c}$.

Figure 7 illustrates an example of a Bezier curve. The left column shows the input and output image pixels with a higher red color name probability than 0.2. The top-right plot shows the tone curves learned for the red color name, while the bottom-right plot displays a zoomed-in view of the first four control points of the R channel. Note how the control points are fixed and evenly distributed along the input axis, while $P_m^{n,c}$ define the output axis value and, thus, the curvature of the tonal curve. The six sets of Bezier curves learned are applied pixel-wise to the entire standardized image $\hat{y}_b$, yielding six distinct globally adjusted images.

D. Transformer-based Image Fusion

Tone curves enable interpretable global color manipulation, but they are not sufficient to reproduce the localized, content-dependent adjustments typically performed by experts (e.g., selectively enhancing a subject while keeping the background unchanged). To capture these local editing effects, we introduce a transformer-based fusion block that combines the six globally adjusted images into a single enhanced output.

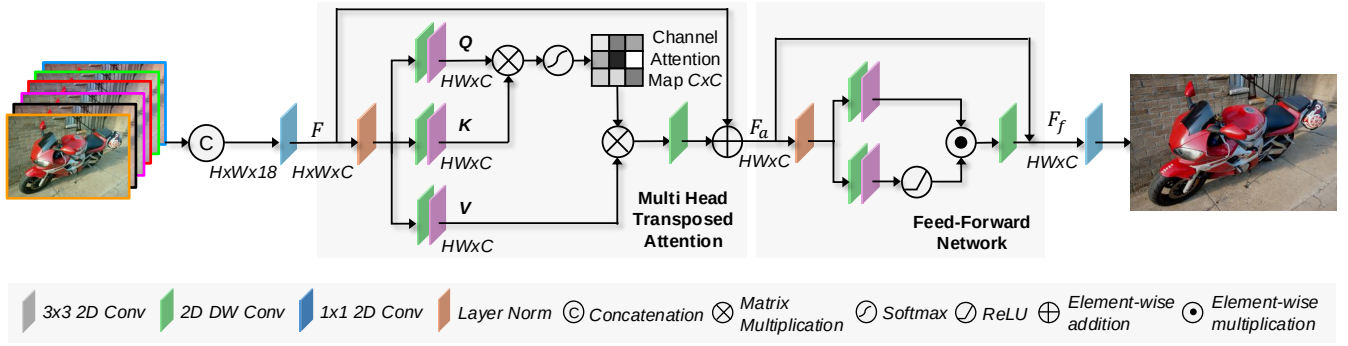


Fig. 8. Overview of the transformer block. The globally-adjusted images, each one conditioned by a color term, are concatenated and passed through a point-wise convolution layer. The resulting features, F , are then processed by a Multi-Head Transposed Attention module and refined by a Feed-Forward Network, culminating in a three-channel output that forms the final enhanced image.

Limitations of the previous fusion mechanism. In our previous approach, NamedCurves, we processed the six globally adjusted images independently with a spatial self-attention module and then averaged the results using the color naming probabilities as weights. Concretely, each globally adjusted image provided the *key* and *value*, while the standardized image acted as the *query*. We first reduced the spatial resolution with two Conv-ReLU-MaxPooling blocks, producing feature maps with a broad receptive field; in particular, $Q, K, V \in \mathbb{R}^{W/8 \times H/8 \times 16}$. The output of each attention module, $\hat{y}_i$, can be written as:

$$\hat{y}_i = \hat{y}_b + \text{Softmax}(QK^T)V. \quad (3)$$

Finally, we obtained the final image $\hat{y}$ as a weighted average of the attention outputs. This design has three major drawbacks. First, it is memory intensive because the attention maps are computed on relatively high-resolution features (only down-sampled by a factor of eight). Second, because each color term is processed independently, the module cannot model dependencies across color-specific adjustments. Third, the final weighted averaging can be brittle when color-term probabilities overlap at boundaries, often producing halo artifacts.

Proposed transformer-based fusion. Standard spatial self-attention is effective but suffers from quadratic time and memory complexity due to the key-query dot-product over spatial positions, which becomes prohibitive for high-resolution inputs. To address this, we adopt the transposed attention mechanism introduced by Zamir et al. [47], which computes cross-covariance across channels instead of across spatial locations. This yields an attention map whose cost scales linearly with image size. Moreover, rather than partitioning the image into tokens (patches), we operate directly on dense feature maps, which is more suitable for low-level vision. Transposed attention has been widely adopted in restoration and enhancement tasks due to its favorable efficiency/accuracy trade-off and its ability to capture long-range dependencies [48]–[50]. Figure 8 shows the architecture of our fusion block.

The fusion block consists of a Multi-Head Transposed Attention module followed by a Feed-Forward Network. We first concatenate the six globally adjusted RGB images to form an 18-channel input $I \in \mathbb{R}^{H \times W \times 18}$. A 3×3 convolution extracts

low-level features $F \in \mathbb{R}^{H \times W \times C}$. Following Zhang et al. [51], we compute the *query* (Q), *key* (K), and *value* (V) projections from F via Layer Normalization, a 1×1 convolution, and a 3×3 depth-wise convolution for each branch, producing $Q, K, V \in \mathbb{R}^{H \times W \times C}$. To build the transposed attention map A , we swap the usual roles of Q and K :

$$A = \text{Softmax}(K^T Q). \quad (4)$$

We then multiply A by V and refine the result with a depth-wise convolution. As in Zamir et al. [47], we split the channels into multiple heads, allowing the module to learn diverse attention patterns in parallel.

The output of the attention module is further refined by a Feed-Forward Network (FFN), which processes features independently at each spatial location while sharing parameters across the image. Specifically, we normalize the attention features F_a with Layer Normalization and process them through two parallel branches, each composed of a point-wise convolution, a depth-wise convolution, and a final 1×1 convolution. This FFN enriches local interactions while preserving efficiency. Finally, we apply a 3×3 convolution to map the features F_f to three channels, producing the final enhanced sRGB image. In all experiments, we set $C=18$, matching the number of input channels.

E. Loss Function

Our training loss function comprises three terms. The initial term calculates the L_2 loss between the standardized image $\hat{y}_b$ and the ground truth y , with a weighting factor α . The subsequent two terms compute the L_2 and $SSIM$ losses between the output of our model $\hat{y}$ and y . The primary objective of the first term is to obtain a good enough standardized output by the backbone. The other two terms are designed to assess the fidelity of the final output. In our experiments, we set α to 0.5. This value was determined to yield optimal performance and allows the image $\hat{y}_b$ to represent the scene colors accurately. The training loss is defined as:

$$L(\hat{y}_b, \hat{y}, y) = \alpha \|y - \hat{y}_b\|_2 + \|y - \hat{y}\|_2 + (1 - SSIM(y, \hat{y})). \quad (5)$$

TABLE I

IMAGE RETOUCHING. QUANTITATIVE COMPARISONS ON THE MIT-ADOBE-5K DATASET [10]. - MEANS THE SOURCE CODE OR MODELS ARE UNAVAILABLE, OR RESULTS FOR THE CORRESPONDING METRIC WERE NOT IN THE ORIGINAL PAPER.

Method	PSNR	SSIM	LPIPS	ΔE_{00}	ΔE_{ab}	Time (ms)
UPE [53]	21.88	0.853	-	-	10.80	-
DPE [5]	23.75	0.908	-	-	9.34	-
DeepLPP [30]	24.73	0.916	0.078	7.80	7.99	136
3DLUT [1]	25.29	0.923	0.043	6.76	7.55	13
SepLUT [3]	25.47	0.921	0.042	6.71	7.49	10
AdaInt [2]	25.49	0.926	0.041	6.69	7.47	13
RSFNet [31]	25.49	0.924	0.043	6.73	7.23	121
NamedCurves [14]	25.59	0.936	0.038	6.07	7.40	26
BGLUT [4]	25.66	0.930	0.038	6.06	7.29	15
NamedCurves+	25.75	0.940	0.035	6.01	7.27	19

IV. EXPERIMENTS

A. Implementation Details

We trained our model using Adam [52], an initial learning rate $1e-4$, reduced by 50% every 50 epochs. We use horizontal flips for augmenting the training data. We train the models for a fixed 200 epochs for all the datasets: MIT5K [10], PPR10K [11], MSEC [12], and SICE [13].

B. Quantitative Results

Image Retouching. We compare our method with state-of-the-art image retouching methods using the MIT-Adobe-5K dataset [10] and the PPR10K dataset [11]. MIT5K [10] consists of 5,000 images captured independently using several DSLR cameras and retouched by five artists. Following [5], [30], [33], [34], [53] we only use the Expert C retouched images as ground truth, and the pre-processing proposed by Ni et al. [6]. Specifically, we split the image into 4,500 images for training and 500 images for testing. PPR10K [11] is a portrait photo retouching dataset with 11,616 high-quality images retouched by three experts independently. We use the official splits, dividing the images into 8,875 and 2,286 for training and testing, respectively. Following [2], we conducted the experiments on the 360p augmented version, in which every image pair has five extra input versions with different manual settings. As in [1], [2], [11], we evaluate our method on the three expert-retouched versions of PPR10K.

In Table I, we report the PSNR, SSIM, LPIPS, ΔE_{00} , and ΔE_{ab} of the state-of-the-art methods and our approach for the MIT5K dataset. We also report the inference time for a 480p image on an AMD EPYC 7642 and an NVIDIA A40. Our full architecture outperforms contemporary image-to-image, curve estimation, and LUT-based methods. Table II reports the PSNR and ΔE_{ab} on the PPR10K dataset. Our method consistently outperforms the previous state-of-the-art methods across the three experts.

Tone Mapping. We compare our method with state-of-the-art tone mapping methods using the MIT-Adobe-5K dataset [10]. In contrast with image retouching, the input images are in 16-bit CIE XYZ format and the outputs are sRGB images. Following Kim et al. [4], in Table III we report PSNR, SSIM, and ΔE_{ab} of the previous state-of-the-art methods and our

TABLE II

IMAGE RETOUCHING. QUANTITATIVE COMPARISONS ON PPR10K.

Method	Expert A		Expert B		Expert C	
	PSNR	ΔE_{ab}	PSNR	ΔE_{ab}	PSNR	ΔE_{ab}
HDRNet [54]	23.93	8.70	23.96	8.84	24.08	8.87
3DLUT [1]	25.64	6.97	24.70	7.71	25.18	7.58
SepLUT [3]	26.28	6.59	25.23	7.49	25.59	7.51
AdaInt [2]	26.33	6.56	25.40	7.33	25.68	7.31
RSFNet [31]	25.58	6.92	24.81	7.58	25.52	7.47
BGLUT [4]	26.45	6.51	25.48	7.19	25.72	7.28
NamedCurves [14]	26.81	6.48	25.91	7.18	25.69	7.27
NamedCurves+	26.85	6.45	25.95	7.14	25.77	7.24

TABLE III

TONE MAPPING. QUANTITATIVE COMPARISONS ON MIT-ADOBE-5K

Method	PSNR $\uparrow$	SSIM $\uparrow$	ΔE_{ab} $\downarrow$
UPE [53]	21.56	0.837	12.29
DPE [5]	22.93	0.894	11.09
3DLUT [1]	25.07	0.920	7.55
SepLUT [3]	25.43	0.922	7.43
AdaInt [2]	25.28	0.925	7.48
BGLUT [4]	25.59	0.932	7.14
NamedCurves [14]	25.60	0.932	7.15
NamedCurves+ (ours)	25.66	0.933	7.12

approach. We outperform all the current methods across all the metrics, including our previous work NamedCurves [14].

Exposure Correction. We evaluate our approach against state-of-the-art methods and NamedCurves [14] on the ME [12] and SICE [13] datasets. The ME dataset contains over 24,000 images with the widest range of exposure values, paired with corresponding properly exposed reference images. This dataset was generated from the MIT-5K dataset using Adobe Camera RAW, which simulates different exposure values as a camera would apply them. Adobe Camera RAW uses metadata to ensure accurate emulation of nonlinear camera rendering procedures. The ME dataset is divided into 17,675 training images, 750 validation images, and 5,905 testing images. The SICE dataset, on the other hand, comprises 589 high-resolution multi-exposure sequences, totaling 4,413 images captured by seven types of consumer-grade cameras. Following Huang et al. [38], in Table IV, we report the PSNR and SSIM for underexposed images, overexposed images, and the average of all testing images. Our approach achieves the best PSNR and SSIM on both datasets, except for PSNR on overexposed images in the ME dataset and SSIM on overexposed images in the SICE dataset, where our method ranks second. These quantitative results highlight the effectiveness of our two-step approach. First, the backbone network recovers lost details, followed by the Bezier tone curves and transformer block, which enhance the image's color and overall quality.

C. Qualitative Results

Figures 9 and 10 present qualitative comparisons across the MIT5K [10], PPR10K [11], ME [12], and SICE [13] datasets, respectively. In Figure 9 (a) and (b), we compare our framework with BGLUT [4] and NamedCurves [14] for photo retouching. The results demonstrate that our framework

TABLE IV
EXPOSURE CORRECTION. QUANTITATIVE COMPARISONS ON THE MSEC AND SICE DATASETS.

Method	MSEC [12]						SICE [13]					
	Under		Over		Average		Under		Over		Average	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
CLAHE [55]	16.77	0.6211	14.45	0.5842	15.38	0.5990	12.69	0.5037	10.21	0.4878	11.45	0.4942
RetinexNet [56]	12.13	0.6209	10.47	0.5953	11.14	0.6048	12.94	0.5171	12.87	0.5252	12.90	0.5212
ZeroDCE [32]	14.55	0.5887	10.40	0.5142	12.06	0.5441	16.92	0.6330	7.10	0.4292	12.02	0.5311
MSEC [12]	20.52	0.8129	19.79	0.8156	20.35	0.8210	19.62	0.6512	17.59	0.6560	18.58	0.6536
CMEC [37]	22.23	0.8140	22.75	0.8336	22.54	0.8257	17.68	0.6592	18.17	0.6811	17.93	0.6702
I-SID [57]	22.59	0.8423	22.36	0.8519	22.45	0.8481	21.30	0.6645	19.63	0.6941	20.47	0.6793
I-DRBN-4 [57]	22.72	0.8544	22.11	0.8521	22.35	0.8530	21.77	0.7052	19.57	0.7267	20.67	0.7160
RSFNet [31]	22.61	0.8457	22.16	0.8443	22.39	0.8450	21.42	0.6724	19.03	0.6892	20.23	0.6808
FECNet [38]	22.96	0.8598	23.22	0.8748	23.12	0.8688	22.01	0.6737	19.91	0.6961	20.96	0.6849
NamedCurves [14]	23.11	0.8698	22.98	0.8771	23.06	0.8726	21.89	0.7154	21.95	0.7142	21.92	0.7148
NamedCurves+ (ours)	23.24	0.8710	23.09	0.8839	23.15	0.8780	22.07	0.7194	22.10	0.7194	22.08	0.7194

achieves visually appealing outcomes that most closely resemble the expert-retouched versions in terms of color fidelity. This can be observed in the blue sky in the first row, the white petals in the second row, the white dress in the third row, and the red dress in the fourth row. Figure 10 (a) and (b) compare our method against FECNet [38] and NamedCurves [14] for exposure correction. Our framework consistently achieves the most accurate exposure levels, as evidenced by its ability to properly balance details in the mountains in the first row, the sky in the second row, the grass in the third row, and the light filtering through the leaves in the last row. For exposure correction, NamedCurves+ is particularly well-suited, as the backbone first normalizes the dynamic range into a canonical space, after which the color-specific tone curves independently refine each chromatic region. This two-stage approach proves more stable than single-pass methods.

D. User Interactivity

Mimicking the style of an expert may not always align with individual user preferences. By associating tone curves with color names, NamedCurves+ provides both interpretability and interactivity. The proposed method enables users to adjust the estimated control points of the tone curves, thereby manipulating specific color regions in the final enhanced image. When a user modifies a tone curve, they directly alter all pixels with a high probability of belonging to the corresponding color region, as determined by the color naming method. Importantly, the model's weights remain frozen during inference, ensuring that the user interactions are limited to modifying the tone curves without affecting the learned parameters.

Figure 11 illustrates four examples of curve modifications applied to two different images. The first two columns show the input image and the enhanced version as output by our method, respectively. We display the modified curves alongside the corresponding altered output images on the right. Black arrows indicate the direction of the curve modifications applied. In the first image, adjusting the achromatic curves reduces the shadows and highlights of the asphalt and the gray wall behind the motorbike. However, if the highlights of the red curves are increased, the red parts of the motorbike are intensified without affecting the background. In the second

image, the grass or the sky is selectively intensified based on whether the green or blue curves are manipulated. We are committed to publicly releasing a demo to show our method's interactive and user-friendly nature.

Figure 12 demonstrates the model's ability to handle extreme interactive edits on the same image. The leftmost image shows the input, and the second image shows the model's automatic enhancement as a reference baseline. Then, the sequence shows progressive modifications where each step (left to right) increases all control points of the blue tone curve in the *Red* channel by 10%. With each successive edit, the red regions (most prominently the bridge) progressively shift toward a purplish hue. Notably, even under extreme and unrealistic edits, the model produces well-fused results.

E. Ablation Study on Module Contribution

We analyze the contribution of the main components of NamedCurves+ on two representative tasks: image retouching (MIT-Adobe-5K) and exposure correction (MSEC). Table V summarizes the effect of removing or simplifying each module.

Removing the backbone ("w/o Backbone") causes the largest degradation, highlighting that the standardization stage is critical to map the input to a canonical space where curve-based enhancement is effective. Using only one curve and without using color naming decomposition ("w/o naming") also decreases performance, showing that conditioning the tone curves on color terms is important to apply different adjustments to different chromatic regions. Removing the transformer ("w/o Transformer") hurts performance as well, indicating that local, context-aware fusion is necessary beyond global edits alone. Overall, the full NamedCurves+ pipeline achieves the best results, and the improvements over NamedCurves confirm the benefit of the proposed upgrades.

F. Color Naming Ablation Study

In this section, we perform an ablation study to assess the importance of color naming and its impact on the performance of our method. Table VI presents the results of all ablation experiments, where we report PSNR, SSIM, LPIPS, ΔE_{00} , and ΔE_{ab} . We compare the color naming methods proposed

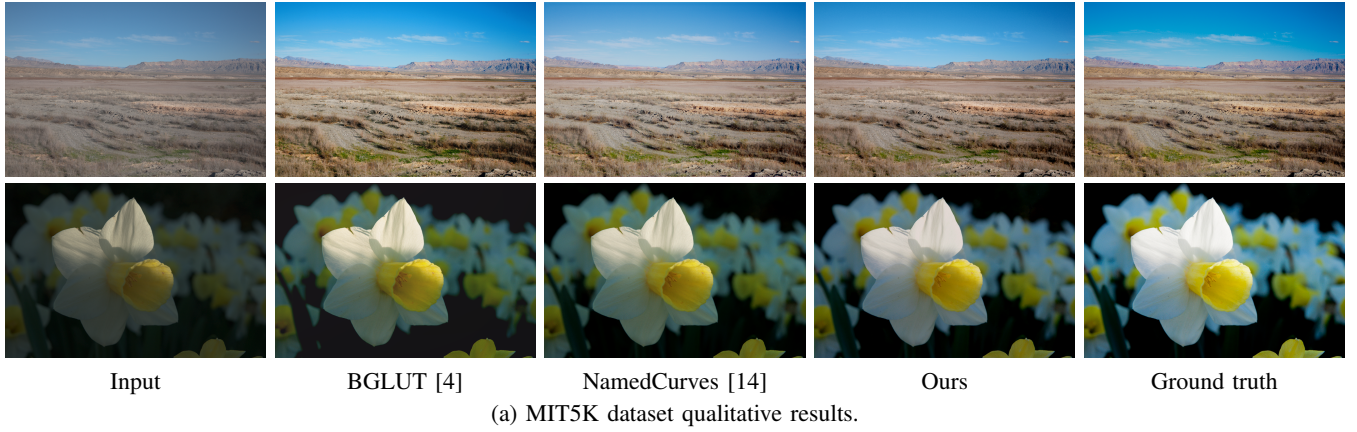
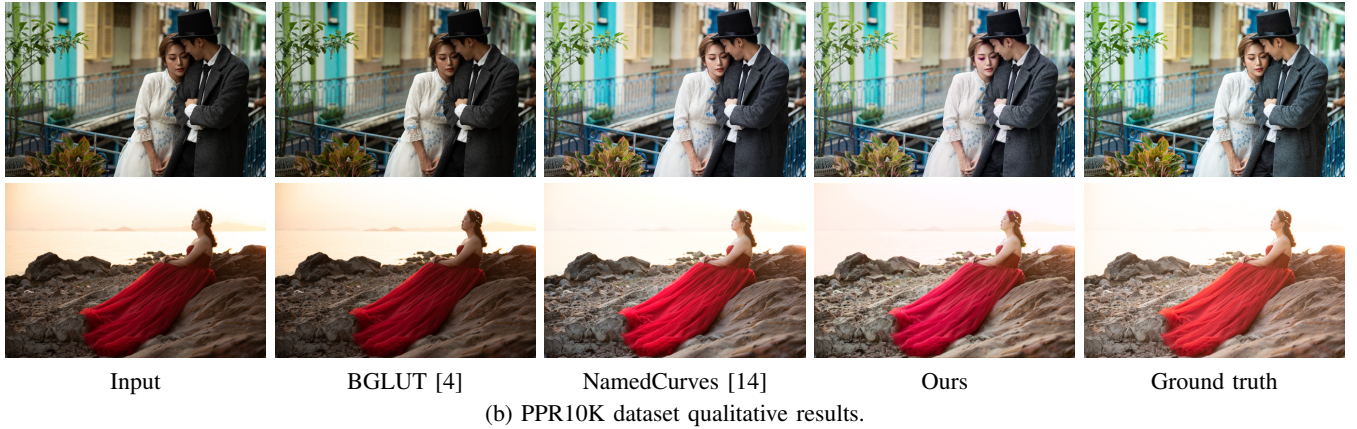


Fig. 9. Qualitative results on image-retouching: (a) MIT5K, and (b) PPR10K datasets.

TABLE V
MODULE CONTRIBUTION ABLATION STUDY ON
MIT-ADOBE-5K(RETOUCHING) AND MSEC (EXPOSURE CORRECTION).

Method	MIT-5K				MSEC	
	PSNR	SSIM	LPIPS	ΔE_{00}	PSNR	SSIM
w/o Backbone	24.30	0.916	0.052	7.16	20.98	0.848
w/o naming	25.13	0.922	0.045	6.55	22.36	0.861
w/o Transformer	25.41	0.929	0.041	6.47	22.72	0.863
NamedCurves	25.59	0.936	0.038	6.07	23.06	0.873
NamedCurves+	25.75	0.940	0.035	6.01	23.15	0.878

by Van de Weijer et al. [20] and Benavente et al. [21], both employing 11 color names, as well as our proposed grouping strategy to reduce them to six color categories. Additionally, we experimented with a simple baseline method in which a pixel probability assignment was based on the distance to the vertices of the RGB cube, resulting in eight channels. To further analyze the role of color grouping, we also generate six and eleven random classes using Gaussian Mixture Models (GMM) over the RGB color space. Finally, we evaluate our method without conditioning the curve estimation process with the color naming decomposition. The results show that, among the different partitioning methods evaluated (i.e., random GMM, RGB cube vertices), color naming is the most adequate to model enhancement operations since color names correlate with human perception, supporting our initial

TABLE VI
QUANTITATIVE COMPARISONS ON THE MIT-ADOBE-5K DATASET [10].

Method	Channels	PSNR	SSIM	LPIPS	ΔE_{00}	ΔE_{ab}
w/o color naming	1	24.91	0.927	0.056	6.67	7.74
w/o color naming	6	25.04	0.928	0.058	6.70	7.76
w/o color naming	11	24.83	0.927	0.055	6.67	7.69
RGB cube vertices	8	24.96	0.928	0.059	6.74	7.81
Random GMM	11	24.09	0.931	0.054	6.59	7.63
Random GMM	6	24.12	0.931	0.052	6.56	7.62
Benavente et al. [21]	11	25.32	0.930	0.049	6.38	7.48
Benavente et al. [21]	6	25.57	0.935	0.039	6.18	7.39
Van de Weijer [20]	11	25.49	0.932	0.041	6.28	7.46
Van de Weijer [20]	6	25.75	0.940	0.035	6.01	7.27

hypothesis. Among color naming methods, Van de Weijer et al. [20] outperforms Benavente et al. [21]. Moreover, reducing the number of color terms to six yields better performance than using all 11 color terms individually.

G. Backbone Analysis

The backbone produces the standardized image $\hat{y}_b$, which serves as the starting point for the subsequent curve-based global editing and transformer-based local fusion. To better understand the impact of this standardization stage, we replace the backbone with several representative architectures

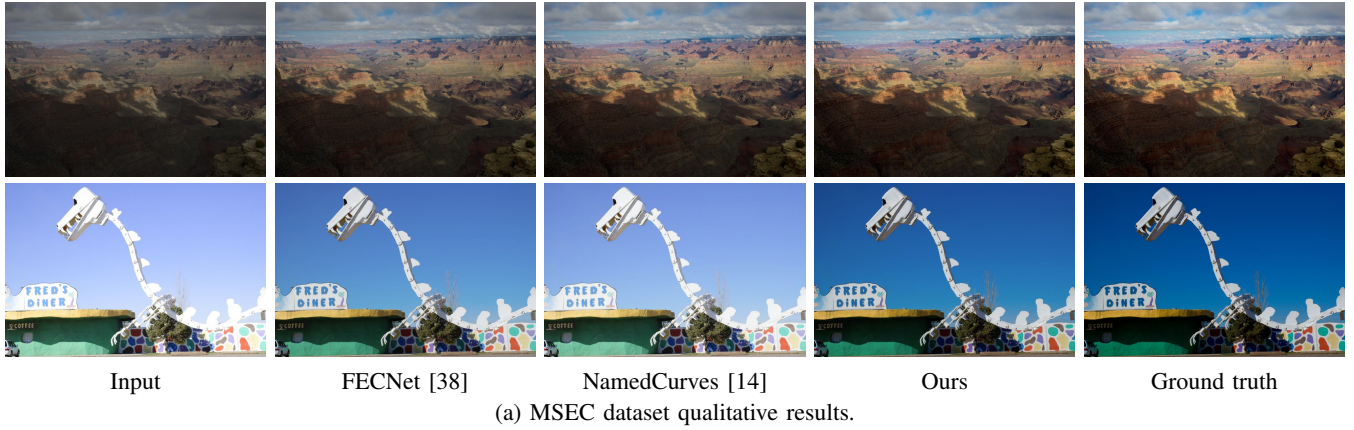
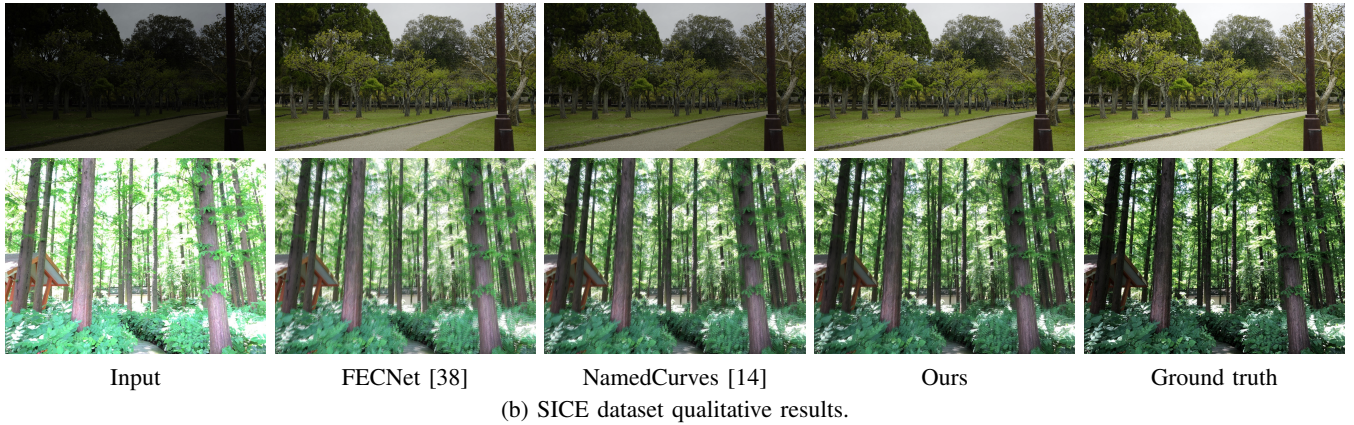


Fig. 10. Qualitative results on exposure correction: (a) ME, and (b) SICE (d) datasets.

TABLE VII
QUANTITATIVE BACKBONE ANALYSIS.

Method	MIT-5K				MSEC	
	PSNR	SSIM	LPIPS	ΔE_{00}	PSNR	SSIM
Conv-UNet	25.17	0.931	0.041	7.05	22.21	0.864
TED	25.22	0.934	0.039	6.87	22.45	0.868
Restormer	25.47	0.938	0.036	6.33	22.94	0.873
LPIENet	25.51	0.938	0.035	6.29	22.98	0.874
Ours	25.75	0.940	0.035	6.01	23.15	0.878

TABLE VIII
COMPUTATIONAL COST COMPARISON.

Fusion Module	Inf. time	Num. Params	VRAM	FLOPs
NamedCurves	26ms	993K	2.44GB	814G
NamedCurves+	19ms	1015K	3.40GB	766G

and evaluate the overall pipeline while keeping the rest of the framework unchanged. Table VII reports the results on MIT-Adobe-5K and MSEC. We observe that stronger restoration-oriented backbones consistently lead to better final enhancement quality, especially on the more challenging exposure correction setting (MSEC), where recovering details in shadows/highlights is essential. While a Conv-UNet already provides a reasonable baseline, transformer-based or hybrid designs (e.g., Restormer and LPIENet) improve perceptual similarity and reduce color error. Our backbone achieves

TABLE IX
STUDY ON THE NUMBER OF
CONTROL POINTS. MIT5K

N	PSNR	SSIM	ΔE_{00}
7	25.70	0.938	6.04
11	25.75	0.940	6.01
16	25.59	0.935	6.15

TABLE X
STUDY ON α OF THE LOSS
FUNCTION. MIT5K

α	PSNR	SSIM	ΔE_{00}
0	25.41	0.930	6.29
0.5	25.75	0.940	6.01
1	25.58	0.937	6.21

the best overall performance across datasets and metrics, indicating that accurate standardization is a key enabler: it reduces the burden on the curve estimation module (which can then focus on interpretable, color-conditioned global edits) and provides cleaner inputs for the transformer fusion module to model local, context-dependent corrections.

In Figure 13, we show two examples of NamedCurves+, and the intermediate standardized image $\hat{y}_b$. By imposing an additional constraint on the loss term with $\hat{y}_b$, we impose that the standardized image is a first enhanced version. As can be seen in Figure 13, the standardized image corrects the exposure of the input image. Then the color naming curves and fusion module refine it to obtain closer results to the ground truth.

H. Fusion Mechanism Computational Cost

Table VIII presents a detailed computational analysis comparing our new fusion module with the previous approach. We report: (i) average inference time over 100 images from the

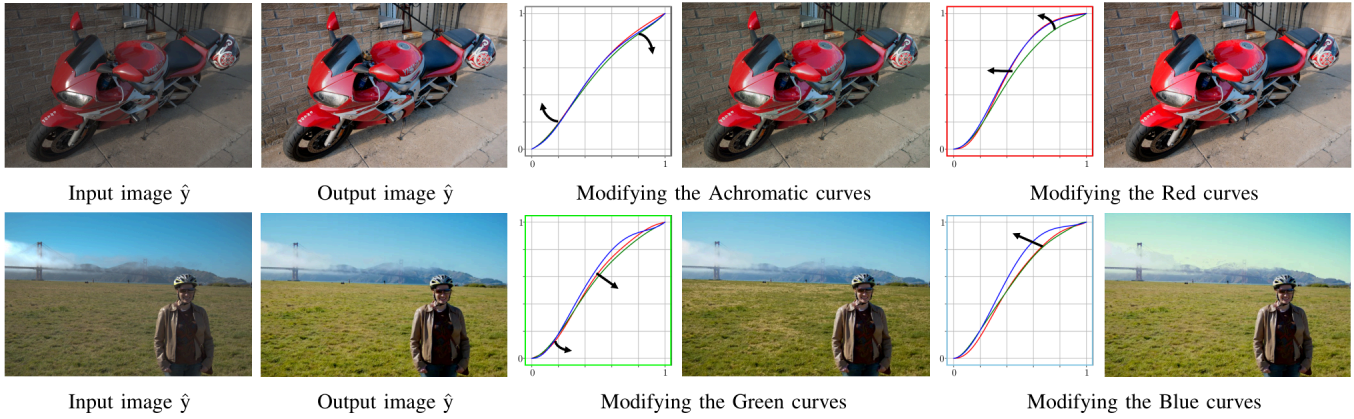


Fig. 11. Examples demonstrating the interactivity of our method. The first column displays the input image followed by the output image given by our end-to-end model. On the right, we present two distinct examples of curve modifications for each image, with black arrows indicating the direction of the adjustments.



Fig. 12. Progressive interactive editing of a single image. From left to right: input, NamedCurves+ automatic enhancement, and successive edits increasing all blue tone curve control points of the color name *Red* by 10% per step (when possible).



Fig. 13. Examples of NamedCurves+ and the intermediate standardized image.

MIT5K dataset on an AMD EPYC 7642 CPU and NVIDIA A40 GPU, (ii) total number of parameters for both complete models, (iii) peak VRAM consumption, and (iv) FLOPs for processing a 1024×1024 image. The new Transformer fusion mechanism adds minimal overhead in terms of parameters (+22K, or 2.2% increase) while requiring moderately higher VRAM (+0.96GB) due to full-resolution processing. Importantly, despite these modest increases, the approach achieves both faster inference (27% reduction from 26ms to 19ms) and reduced computational cost (6% fewer FLOPs, from 814G to 766G), while simultaneously improving results both quantitatively and qualitatively. The efficiency gains stem from replacing six independent attention modules with a single unified Transformer block that leverages GPU-optimized transposed attention operations.

I. Hyperparameter Analysis

We conducted additional analysis to evaluate the impact of the number of control points, N , and the parameter α in the loss function. Specifically, we tested our method with 7, 11, and 16 control points for each Bézier curve. Table IX reports the PSNR, SSIM, and ΔE_{00} for each configuration. Our experiments show that 11 control points, spaced evenly every 0.1 along the input axis, yield the best performance. Additionally, we examined the influence of the α parameter, which balances the contribution of the backbone and the final image component in the loss function. We tested three settings: removing the backbone component ($\alpha=0$), assigning it half the weight of the final image component ($\alpha=0.5$), and giving it equal weight ($\alpha=1$). As shown in Table X, we found that $\alpha=0.5$ provides the best results for our method.

J. User Study

We compared NamedCurves+ against NamedCurves and BGLUT following a two-alternative forced choice (2AFC), performed in a completely black room with a monitor set to sRGB. We selected the same 25 images from both MIT5K and PPR10K datasets as in [14] (that were selected randomly). 15 observers took part, all tested for colorblindness with the Ishihara test. Results analyzed using Thurstone Case V (larger means better) were: NamedCurves+: 1.0672; BGLUT: -0.3793; NamedCurves: -0.6878. Our method is statistically significantly better than the other two —95% confidence interval is 0.33. Figure 14 shows the results.

V. LIMITATIONS

Despite the strong quantitative results and the interpretability afforded by color-specific tone curves, our approach has

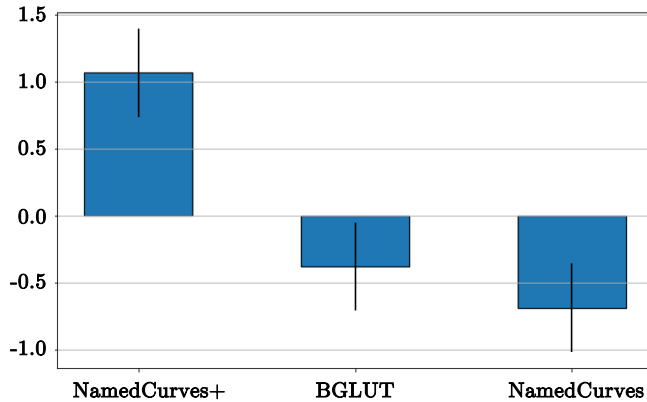


Fig. 14. Two-alternative forced choice (2AFC) on 25 images from MIT5K and PPR10K datasets. 15 observers took part.

some limitations. First, the global editing space is constrained by the expressiveness of tone curves applied per color term. This is well suited to global color and contrast changes, but it cannot directly represent highly localized or semantic operations (e.g., object-specific relighting, instance-level recoloring, or edits that depend on high-level scene understanding). The transformer fusion alleviates this limitation by incorporating local context, but it still operates on a fixed set of globally manipulated candidates, and thus remains bounded by them. Second, the approach depends on the quality of the color naming decomposition. When color term probabilities are uncertain (e.g., at chromatic boundaries, saturated highlights, or under unusual illuminants), edits may partially leak across regions. Finally, a limitation shared by all retouching methods that train a model to mimic the style of an *expert* is the difficulty of generalizing to the preferences of all users. We partially address this by allowing users to interactively modify the color-specific tone curves; nevertheless, the automatic output will inherently reflect the aesthetic style of the training data.

VI. CONCLUSION

In this paper, we introduce NamedCurves+, a novel approach to efficiently enhance images in three scenarios: photo retouching, tone mapping, and exposure correction. NamedCurves+ incorporates significant advancements, including a transformer block for enhanced image fusion, support for additional tasks with more efficient computation, and analysis of interactivity. This interactivity enables users to fine-tune image colors independently by adjusting the tone curves, allowing them to customize the results to match their preferences better.

REFERENCES

- [1] 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 Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 4, pp. 2058–2073, 2020.
- [2] C. Yang, M. Jin, X. Jia, Y. Xu, and Y. Chen, "Adaint: Learning adaptive intervals for 3d lookup tables on real-time image enhancement," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 17 522–17 531.

- [3] C. Yang, M. Jin, Y. Xu, R. Zhang, Y. Chen, and H. Liu, "Seplut: Separable image-adaptive lookup tables for real-time image enhancement," in *European Conference on Computer Vision*. Springer, 2022, pp. 201–217.
- [4] W. Kim and N. I. Cho, "Image-adaptive 3d lookup tables for real-time image enhancement with bilateral grids," in *European Conference on Computer Vision*. Springer, 2025, pp. 91–108.
- [5] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.
- [6] Z. Ni, W. Yang, S. Wang, L. Ma, and S. Kwong, "Towards unsupervised deep image enhancement with generative adversarial network," *IEEE Transactions on Image Processing*, vol. 29, pp. 9140–9151, 2020.
- [7] Y. Jiang, X. Gong, D. Liu, Y. Cheng, C. Fang, X. Shen, J. Yang, P. Zhou, and Z. Wang, "Enlightengan: Deep light enhancement without paired supervision," *IEEE Transactions on Image Processing*, vol. 30, pp. 2340–2349, 2021.
- [8] Adobe Systems, "Adobe photoshop," Computer software. [Online]. Available: photoshop.adobe.com
- [9] B. Berlin and P. Kay, *Basic color terms: Their universality and evolution*. Univ of California Press, 1991.
- [10] V. Bychkovsky, S. Paris, E. Chan, and F. Durand, "Learning photographic global tonal adjustment with a database of input/output image pairs," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2011, pp. 97–104.
- [11] J. Liang, H. Zeng, M. Cui, X. Xie, and L. Zhang, "Ppr10k: A large-scale portrait photo retouching dataset with human-region mask and group-level consistency," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 653–661.
- [12] M. Afifi, K. G. Derpanis, B. Ommer, and M. S. Brown, "Learning multi-scale photo exposure correction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 9157–9167.
- [13] J. Cai, S. Gu, and L. Zhang, "Learning a deep single image contrast enhancer from multi-exposure images," *IEEE Transactions on Image Processing*, vol. 27, no. 4, pp. 2049–2062, 2018.
- [14] D. Serrano-Lozano, L. Herranz, M. S. Brown, and J. Vazquez-Corral, "Namedcurves: Learned image enhancement via color naming," in *European Conference on Computer Vision*, 2025, pp. 92–108.
- [15] D. Xue, J. V. Corral, L. Herranz, Y. Zhang, and M. S. Brown, "Integrating high-level features for consistent palette-based multi-image recoloring," in *Computer Graphics Forum*, vol. 42, no. 7. Wiley Online Library, 2023, p. e14964.
- [16] D. Xue, J. Vazquez-Corral, L. Herranz, Y. Zhang, and M. S. Brown, "Palette-based color harmonization via color naming," *IEEE Signal Processing Letters*, vol. 31, pp. 1474–1478, 2024.
- [17] T. Regier, P. Kay, and N. Khetarpal, "Color naming reflects optimal partitions of color space," *Proceedings of the National Academy of Sciences*, vol. 104, no. 4, pp. 1436–1441, 2007.
- [18] D. A. Szafrir, "Modeling color difference for visualization design," *IEEE Transactions on Visualization and Computer Graphics*, vol. 24, no. 1, pp. 392–401, 2017.
- [19] H. Bahng, S. Yoo, W. Cho, D. K. Park, Z. Wu, X. Ma, and J. Choo, "Coloring with words: Guiding image colorization through text-based palette generation," in *European Conference on Computer Vision*, 2018, pp. 431–447.
- [20] J. Van De Weijer, C. Schmid, J. Verbeek, and D. Larlus, "Learning color names for real-world applications," *IEEE Transactions on Image Processing*, vol. 18, no. 7, pp. 1512–1523, 2009.
- [21] R. Benavente, M. Vanrell, and R. Baldrich, "Parametric fuzzy sets for automatic color naming," *Journal of the Optical Society of America A*, vol. 25, no. 10, pp. 2582–2593, 2008.
- [22] L. Yu, Y. Cheng, and J. van de Weijer, "Weakly supervised domain-specific color naming based on attention," in *International Conference on Pattern Recognition*. IEEE, 2018.
- [23] C. A. Parraga and A. Akbarinia, "Nice: A computational solution to close the gap from colour perception to colour categorization," *PloS one*, vol. 11, no. 3, p. e0149538, 2016.
- [24] J. A. Stark, "Adaptive image contrast enhancement using generalizations of histogram equalization," *IEEE Transactions on Image Processing*, vol. 9, no. 5, pp. 889–896, 2000.
- [25] A. M. Reza, "Realization of the contrast limited adaptive histogram equalization (clahe) for real-time image enhancement," *Journal of VLSI Signal Processing-Systems for Signal, Image, and Video Technology*, vol. 38, pp. 35–44, 2004.

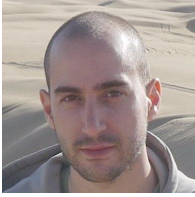
- [26] M. Aubry, S. Paris, S. W. Hasinoff, J. Kautz, and F. Durand, "Fast local laplacian filters: Theory and applications," *ACM Transactions on Graphics*, vol. 33, no. 5, pp. 1–14, 2014.
- [27] F. Durand and J. Dorsey, "Fast bilateral filtering for the display of high-dynamic-range images," in *Conference on Computer Graphics and Interactive Techniques*, 2002.
- [28] J. Van De Weijer, T. Gevers, and A. Gijsenij, "Edge-based color constancy," *IEEE Transactions on Image Processing*, vol. 16, no. 9, pp. 2207–2214, 2007.
- [29] T. Wang, Y. Li, J. Peng, Y. Ma, X. Wang, F. Song, and Y. Yan, "Real-time image enhancer via learnable spatial-aware 3d lookup tables," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021.
- [30] S. Moran, P. Marza, S. McDonagh, S. Parisot, and G. Slabaugh, "Deepplf: Deep local parametric filters for image enhancement," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 12 826–12 835.
- [31] W. Ouyang, Y. Dong, X. Kang, P. Ren, X. Xu, and X. Xie, "Rsfnets: A white-box image retouching approach using region-specific color filters," in *ICCV*, 2023.
- [32] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020, pp. 1780–1789.
- [33] S. Moran, S. McDonagh, and G. Slabaugh, "Curl: Neural curve layers for global image enhancement," in *International Conference on Pattern Recognition*, 2021, pp. 9796–9803.
- [34] C. Li, C. Guo, S. Zhou, Q. Ai, R. Feng, and C. C. Loy, "Flexicurve: Flexible piecewise curves estimation for photo retouching," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 1092–1101.
- [35] S. Bianco, C. Cusano, F. Piccoli, and R. Schettini, "Personalized image enhancement using neural spline color transforms," *IEEE Transactions on Image Processing*, vol. 29, pp. 6223–6236, 2020.
- [36] L. Zhao, A. Abdelhamed, and M. S. Brown, "Learning tone curves for local image enhancement," *IEEE Access*, vol. 10, pp. 60 099–60 113, 2022.
- [37] N. E. Nsambi, Z. Hu, and Q. Wang, "Learning exposure correction via consistency modeling," in *British Machine Vision Conference*, 2021, p. 12.
- [38] J. Huang, Y. Liu, F. Zhao, K. Yan, J. Zhang, Y. Huang, M. Zhou, and Z. Xiong, "Deep fourier-based exposure correction network with spatial-frequency interaction," in *European Conference on Computer Vision*. Springer, 2022, pp. 163–180.
- [39] M. V. Conde, F. Vasluianu, J. Vazquez-Corral, and R. Timofte, "Perceptual image enhancement for smartphone real-time applications," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2023, pp. 1848–1858.
- [40] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.
- [41] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "Cbam: Convolutional block attention module," in *European Conference on Computer Vision*, 2018, pp. 3–19.
- [42] D. Marnerides, T. Bashford-Rogers, J. Hatchett, and K. DeBattista, "Expandnet: A deep convolutional neural network for high dynamic range expansion from low dynamic range content," in *Computer Graphics Forum*, vol. 37, no. 2. Wiley Online Library, 2018, pp. 37–49.
- [43] M. Gharbi, J. Chen, J. T. Barron, S. W. Hasinoff, and F. Durand, "Deep bilateral learning for real-time image enhancement," *ACM Transactions on Graphics*, vol. 36, no. 4, pp. 1–12, 2017.
- [44] G. Shing-Sheng and H. Po-Sung, "Influences of psychological factors on image color preferences evaluation," *Color Research & Application*, vol. 35, no. 3, pp. 213–232, 2010.
- [45] K. Topfer and R. Cookingham, "The quantitative aspects of color rendering for memory colors," in *IS&T's PICS*, 2000.
- [46] J. Bennett and G. D. Finlayson, "Simplifying tone curves for image enhancement," *Color and Imaging Conference*, vol. 31, no. 1, pp. 108–108, 2023.
- [47] S. W. Zamir, A. Arora, S. Khan, M. Hayat, F. S. Khan, and M.-H. Yang, "Restormer: Efficient transformer for high-resolution image restoration," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 5728–5739.
- [48] Y. Cai, H. Bian, J. Lin, H. Wang, R. Timofte, and Y. Zhang, "Retinex-former: One-stage retinex-based transformer for low-light image enhancement," in *ICCV*, 2023.
- [49] X. Chen, Z. Li, Y. Pu, Y. Liu, J. Zhou, Y. Qiao, and C. Dong, "A comparative study of image restoration networks for general backbone network design," in *ECCV*, 2024.
- [50] D. Serrano-Lozano, A. Arora, L. Herranz, K. G. Derpanis, M. S. Brown, and J. Vazquez-Corral, "Revisiting image fusion for multi-illuminant white-balance correction," in *ICCV*, 2025.
- [51] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena, "Self-attention generative adversarial networks," in *International Conference on Machine Learning*. PMLR, 2019, pp. 7354–7363.
- [52] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *International Conference on Learning Representations*, 2015.
- [53] R. Wang, Q. Zhang, C.-W. Fu, X. Shen, W.-S. Zheng, and J. Jia, "Underexposed photo enhancement using deep illumination estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.
- [54] J. Li and P. Fang, "Hdrnet: Single-image-based hdr reconstruction using channel attention cnn," in *International Conference on Multimedia Systems and Signal Processing*, 2019, pp. 119–124.
- [55] 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," *Computer Vision, Graphics, and Image Processing*, vol. 39, no. 3, pp. 355–368, 1987.
- [56] C. Wei, W. Wang, W. Yang, and J. Liu, "Deep retinex decomposition for low-light enhancement," in *British Machine Vision Conference*, 2018, p. 155.
- [57] J. Huang, Y. Liu, X. Fu, M. Zhou, Y. Wang, F. Zhao, and Z. Xiong, "Exposure normalization and compensation for multiple-exposure correction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 6043–6052.

VII. ACKNOWLEDGEMENTS

This work was supported by Grant PID2021-128178OB-I00, , PID2024-162555OB-I00 funded by MCIN/AEI/10.13039/ 501100011033 and by ERDF "A way of making Europe", the grant Catedra ENIA UAB-Cruilla (TSI-100929-2023-2) from the Ministry of Economic Affairs and Digital Transformation of Spain, and by the Generalitat de Catalunya CERCA Program. DSL also acknowledges the FPI grant from Spanish Ministry of Science and Innovation (PRE2022-101525). LH was also supported by the Ramon y Cajal grant RYC2019-027020-I. The CFREF (VISTA) program, an NSERC Discovery Grant, and the Canada Research Chair program supported MSB.



David Serrano-Lozano is a PhD student at the Computer Vision Center and Universitat Autònoma de Barcelona. He received his B.Sc. degree from Universitat Politècnica de Catalunya and his M.Sc. degree from Universitat Autònoma de Barcelona. His research interests include low-level computer vision, generative models, and computational imaging.



Luis Herranz is a researcher at the Universidad Autónoma de Madrid. He received his Ph.D. degree from the Universidad Autónoma de Madrid in 2010. He has worked at the Computer Vision Center (Barcelona), the Institute of Computing Technology (Chinese Academy of Sciences), and Mitsubishi Electric Research and Development. His research interests include deep learning applied to various topics in computer vision and multimedia.



Michael S. Brown is a Professor and a Canada Research Chair of Computer Vision with York University, Toronto. He also holds a part-time position as a 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, 2017, 2019) and 3DV (2015), as well as General Chair for ACCV (2014) and CVPR (2018, 2021, 2023). He is a Fellow of the IEEE.



Javier Vazquez-Corral is an Associate Professor at the Universitat Autònoma de Barcelona and a researcher at the Computer Vision Center, Barcelona. He received his Ph.D. degree from the Universitat Autònoma de Barcelona. He has worked at the University of East Anglia, the École Polytechnique Fédérale de Lausanne, and the Universitat Pompeu Fabra. His research interests include the use of color in computer vision problems and bridging the gap between color perception in the human brain and its application in computer vision.