

CURL: Neural Curve Layers for Global Image Enhancement

Sean Moran^{*,1}, Steven McDonagh², Gregory Slabaugh^{*,3}

¹JP Morgan Chase & Co., ²Huawei Noah's Ark Lab, ³Queen Mary University of London
sean.j.moran@jpmchase.com, steven.mcdonagh@huawei.com, g.slabaugh@qmul.ac.uk

Abstract—We present a novel approach to adjust global image properties such as colour, saturation, and luminance using human-interpretable image enhancement curves, inspired by the Photoshop curves tool. Our method, dubbed neural CURve Layers (CURL), is designed as a multi-colour space neural retouching block trained jointly in three different colour spaces (HSV, CIELab, RGB) guided by a novel multi-colour space loss. The curves are fully differentiable and are trained end-to-end for different computer vision problems including photo enhancement (RGB-to-RGB) and as part of the image signal processing pipeline for image formation (RAW-to-RGB). To demonstrate the effectiveness of CURL we combine this global image transformation block with a pixel-level (local) image multi-scale encoder-decoder backbone network. In an extensive experimental evaluation we show that CURL produces state-of-the-art image quality versus recently proposed deep learning approaches in both objective and perceptual metrics, setting new state-of-the-art performance on multiple public datasets. Our code is publicly available at: <https://github.com/sjmoran/CURL>.

I. INTRODUCTION

Image quality is of fundamental importance in any imaging system, especially DSLR and smartphone cameras. Modern smartphones for example, provide multiple camera sensors and sophisticated image signal processor (ISP) pipelines to produce images with good contrast, detail, colours, and dynamic range while at the same time mitigating against degradations. However, despite great advances in imaging hardware and ISP pipelines, there remains substantial room for improvement. Even professional photographers using DSLR will spend significant time using photo editing software to produce a quality digital photograph.

This paper is inspired by the Photoshop curves tool¹, which professional photographers use to modify global properties of an image through manual design of adjustment curves. However, most casual users lack artistic skills (or software) to retouch their photos in such a manner. In this context, several interesting research questions emerge: *Is it possible to automatically estimate, and apply, adjustment curves through multiple colour spaces to improve image quality? What collection of curves are necessary? What order should image adjustments be applied?* This paper not only precisely poses these research questions, but answers them using new neural

CURve Layers dubbed **CURL**. We arrange a collection of such curve adjustments in a defined sequence operating in multiple colour spaces, resulting in a **CURL block**.

CURL can be used along with a deep backbone feature extractor to enhance images in an **RGB-to-RGB** mapping. An example is shown in Figure 1, where CURL adjusts an underexposed image to produce a result with better contrast and more pleasing colours than other methods such as the state-of-the-art DeepUPE [1] approach.

Alternatively, a CURL block can be used as part of the **RAW-to-RGB** transformation modeling the entire ISP pipeline. In this configuration, first a pixel level block (modeled as an encoder/decoder network) is applied to perform local operations such as denoising and demosaicing, the latter estimating RGB colours at each pixel from the RAW data collected using a colour filter array. Then, a CURL block is used to transform the brightness, colour, and saturation to produce the final image. Modeling the entire ISP in this way, as a single neural network trained end-to-end, is very appealing as standard ISP pipelines consist of many different modules (e.g. denoising, demosaicing, automatic white balance, tone mapping, etc.) that are sequentially applied. Such ISPs are difficult and expensive to design, develop, and tune due to the complex dependencies between the modules.

Our contributions in this paper are three-fold:

- **Multi-colour space neural retouching block:** We introduce **CURL**, a neural retouching block, that learns a set of piece-wise linear scaling curves to globally adjust image properties in a human-interpretable manner, inspired by the Photoshop curves tool.
- **Multi-colour space loss function:** We apply sequential differentiable transformations of the image in different colour spaces (Lab, RGB, HSV) guided by a novel multi-colour space loss function.
- **Improved Encoder/Decoder for image enhancement:** CURL modifies images initially enhanced by a backbone network. We explore an effective backbone encoder/decoder architecture. Our Transformed Encoder-Decoder backbone (dubbed **TED**) has all but one U-Net skip connection removed, with the remaining skip connection endowed with a multi-scale neural processing block that enriches the information available to the decoder.

*Work done while at Huawei Noah's Ark Lab.

¹<https://www.cambridgeincolor.com/tutorials/photoshop-curves.htm>



Fig. 1. **Underexposed image enhancement.** Given a poorly exposed image, our method produces an image with pleasing contrast and colour better matching the groundtruth compared to the state-of-the-art DeepUPE model [1].

II. RELATED WORK

The CURL block, illustrated in Figure 2, is useful in two image enhancement scenarios, both of which we explore in this paper: a) *Photo Enhancement (RGB-to-RGB mapping)*: taking an input RGB image and mapping that image to a visually pleasing output RGB image and b) *RAW-to-RGB mapping*: modelling the full ISP pipeline, taking RAW sensor input and producing an RGB output image.

Photo enhancement typically relates to brightness and colour adjustment. The state-of-the-art DeepUPE [1] approach learns a luminance adjustment matrix that improves the exposure of an image. Hu *et al.* [2] design a photo retouching approach (White-Box) using reinforcement learning and GANs to apply a sequence of filters to an image. Deep Photo Enhancer (DPE) [3] is a GAN-based architecture with a U-Net-style generator for RGB image enhancement that produces state-of-the-art results on the popular MIT-Adobe5K benchmark dataset. Recently [4] demonstrated an image enhancement method that does not require reference training images, and [5] presented a content-preserving tone adjustment method. Arguably, the highest quality in digital photography is achieved with DSLR which, due to the large aperture and lenses, can produce higher quality photographs than a smartphone camera. Related work enhances smartphone images by learning a mapping between smartphone images and DSLR [6], [7] photographs using deep learning. Our proposed CURL block differs from these methods in its exploitation of multiple colour spaces and the loss function which drives curve learning jointly in each colour space, outperforming [1], [3], [6] on the RGB-to-RGB task.

Further combination of multiple ISP tasks (*e.g.* denoising, demosaicing, photo enhancement *etc.*) ultimately leads to modeling the entire ISP RAW-to-RGB transformation. Currently the literature is limited for deep learning methods that replace the full pipeline. The most relevant related work is DeepISP [8], which incorporates a low level network that performs local adjustment of the image including joint denoising and demosaicing, and a high level network that performs global image enhancement. Other recent work includes the Self-Guided Network (SGN) [9] which relies extensively on pixel shuffling [10] operations, and CameraNet [11] that decomposes the problem into two subproblems of restoration and enhancement. Another related work is [12] which also

learns a RAW-to-RGB mapping, however for extremely dark images using a simple U-Net-style architecture [13]. Nam and Kim [14] present a deep model that mimicks the RAW-to-RGB ISP for different cameras. Despite this progress, it is an open question how far image quality can advance using deep RAW-to-RGB networks. Similar to [12], our proposed pixel-level block has an encoder/decoder structure, however we replace standard skip connections with novel multi-scale contextual awareness (MSCA) connections, providing enriched features from the encoder to the decoder that boost performance. CURL includes global image transformations to adjust colours and brightness, similar to [8], however, we introduce neural curve layers, which more expressively adjust image properties in a controlled but human-interpretable way. Unlike [12] or [8], the CURL block uses multiple colour spaces, producing results that outperform [8] and [12] on the RAW-to-RGB mapping.

III. TED + CURL FOR IMAGE ENHANCEMENT

CURL receives and processes convolutional features from a backbone network (Figure 2). The backbone network, dubbed **Transformed Encoder-Decoder (TED)**, is a multi-scale encoder-decoder neural network (Section III-A) for local pixel processing. The output of TED is passed to the CURL block (Section III-B), employing novel *neural curve layers* that globally adjust image properties. We argue that, as in previous work [3], [8], effective image enhancement requires both local (Section III-A) and global (Section III-B) adjustment. In Section III-A we describe TED, used for local pixel processing, while in Section III-B we describe CURL. The combination of both architectures is referred to as TED+CURL.

A. Transformed Encoder-Decoder (TED) for Local Image Adjustment

Despite its popularity, in our experimental evaluation (Table I), we find that the standard U-Net backbone [3], [12], [13] is not entirely suitable for the image translation task, as it is important to consider global and mid-level context when making local pixel adjustments to reduce spatial inconsistencies in the predicted image. To address this, we redesign the encoder / decoder backbone with a skip connection (Figure 3) that we call *multi-scale contextual awareness* (MSCA) connection. Our **Transformed Encoder-Decoder** is dubbed **TED**. The MSCA-connection fuses multiple different contextual image features, combining global and mid-level

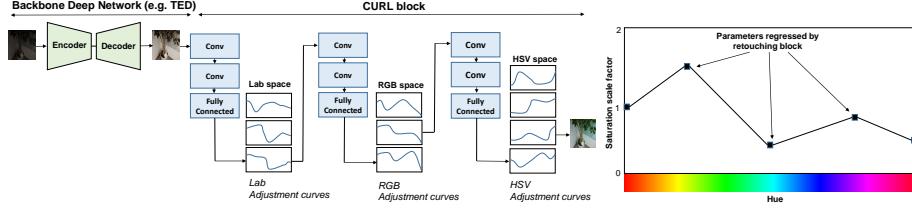


Fig. 2. **Left:** Placement of the CURL block with regards to an encoder-decoder backbone. The CURL block leverages *neural curve layers* for artist-inspired image refinement operations. The input to the backbone network is a RAW image and the output from the CURL block is a high quality RGB image with correct colour and brightness. **Right:** Illustration of a piece-wise linear neural curve from the CURL block that predicts a scaling factor that adjusts saturation based on hue.

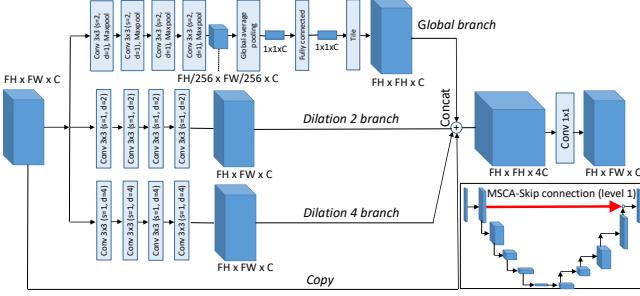


Fig. 3. **Transformed encoder/decoder (TED) backbone** fusing multiple levels of image context to deliver more contextually relevant features for the expanding path (Section III-A). **Main image:** The multi-scale contextual awareness skip connection. **Bottom right:** the connection is placed at the first level skip connection. All other standard U-Net skip connections are removed.

features to enable cross-talk between image content at different scales. The MSCA-connection uses convolutional layers with dilation rate 2 and 4 to gain a larger receptive field and capture mid-level image context from the input. These mid-level feature maps are at the same spatial resolution as the input tensor to the block. Global image features are extracted using a series of convolutional layers with a stride 2, followed by a Leaky ReLU activation and then a max pooling operation. These layers are then followed by global average pooling and a fully connected layer, which outputs a fixed dimensional feature vector, tiled across the height and width dimensions of the input. The resulting feature maps are concatenated to the input and fused by a 1x1 convolution to produce a tensor with much fewer channels. This merges the *local*, *mid-level* and *global* contextual information for concatenation to the feature maps of the upsampling path at that particular level. We find that a MSCA-skip connection at level one of the encoder/decoder provides a good trade-off between parameter complexity and image quality, with all other skip connections removed (Figure 3). Multi-scale neural blocks are not new, having been suggested in the literature before [12], [15], [16]. However, to the best of our knowledge, we are the first to show that *endowing a skip connection* with a multi-scale processing block can dramatically reduce the number of parameters required while increasing the image quality (Table I, Figure 4) for RAW-to-RGB and RGB-to-RGB mapping.

Different to previous work, we evaluate our approach on

TABLE I
BACKBONE NETWORK (TED) WITH MSCA-SKIP CONNECTION, IMPROVES IMAGE QUALITY WITH FEWER PARAMETERS (SAMSUNG S7 DATASET).

Architecture	PSNR	SSIM	# Parameters
TED (MSCA-skip, level 1)	26.56	0.781	1.3 M
TED (MSCA-skip, all levels)	26.39	0.793	3.3 M
U-Net	25.78	0.771	1.4 M
U-Net (large)	25.37	0.788	5.1 M

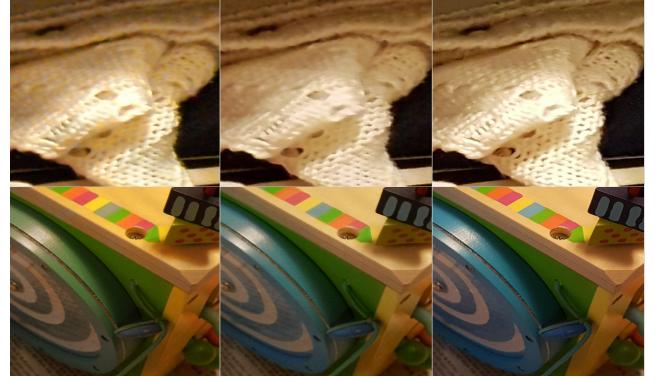


Fig. 4. **Left:** U-Net. **Middle:** TED. **Right:** Groundtruth. TED improves colour reproduction.

both RAW-to-RGB and photo enhancement (RGB-to-RGB). The backbone encoder/decoder differs depending on the type of input data (RAW or RGB). Both architecture diagrams are presented in the supplementary material. For RAW-to-RGB experiments we modify the encoder architecture to facilitate the processing of RAW images represented by a colour filter array (CFA). The CFA tensor $H \times W \times 1$ is shuffled into a packed form using a pixel shuffle layer [10] to give a $(H/r)(W/r) \times r^2$ tensor, where we set $r=2$ in this work. This tensor is fed into the downsampling path of the backbone network. The output from the expanding path is a set of feature maps of half the width and height of the full RGB image. These feature maps are added to the input packed image (replicated $4 \times$ on channel dimension) through a long skip connection. A pixel shuffle upsampling operation [10] produces a result with output shape $H \times W \times C$, where C is the number of feature maps, and is passed to the retouching block for further processing. Our RGB-to-RGB network is broadly similar to the RAW-to-RGB

network, however the pixel shuffling operations are removed. The long skip connection remains, but in this case the 3 channel input image is added to the output, rather than a shuffled version.

B. CURL: Neural Curve Layers for Global Image Adjustment

This section describes CURL, a pluggable global image colour and luminance retouching block which forms the central contribution of this paper. The first three channels of the $H \times W \times C$ tensor from the pixel-level block are treated as the image to be globally adjusted, and the remaining channels serve as feature maps that are used as input to each *neural curve layer*. Our proposed neural curve layer block is shown in Figure 5 and consists of, for each of the three colour spaces, a global feature extraction block followed by a fully connected layer that regresses the knot points of a piecewise-linear curve (Figure 2). The curve adjusts the predicted image ($\hat{I}_i \in [0, 1]$) by scaling pixels with the formula presented in Equation 1.

$$\mathcal{S}(\hat{I}_i^{jl}) = k_0 + \sum_{m=0}^{M-1} (k_{m+1} - k_m) \delta(M\hat{I}_i^{jl} - m), \quad (1)$$

$$\text{where } \delta(x) = \begin{cases} 0 & x < 0 \\ x & 0 \leq x \leq 1 \\ 1 & x > 1 \end{cases}$$

where M is the number of predicted knot points, \hat{I}_i^{jl} is the j -th pixel value in the l -th colour channel of the i -th image, k_m is the value of the knot point m . The neural curve outputs scale factors, so to apply the curve is a simple matter of multiplication of a pixel's value with its scale factor indicated by the curve. Example curves learnt by an instance of our model are shown in Figure 6. CURL regresses expressive curves, used to *scale* rather than *remap* colors in comparison to existing approaches, e.g. [2], [4], [5].² The feature extraction block of the neural curve layer (Figure 5) accepts a $H \times W \times C$ feature map, passing it to a group of blocks each consisting of a convolutional layer with 3×3 kernels of stride 2, and 2×2 maxpooling. We place a global average pooling layer and fully connected layer at the end. The neural curves are learnt in several colour spaces (RGB, Lab, HSV).

In Figure 5, we arrange the neural curve layers in a particular sequence, adjusting firstly luminance and the a, b chrominance channels (using three curves respectively) in CIELab space. Afterwards, we adjust the red, green, blue channels (using three curves respectively) in RGB space. Lastly hue is scaled based on hue, saturation based on saturation, saturation based on hue, and value based on value (using four curves respectively) in HSV space. This ordering of the colour spaces is found to be optimal based on a sweep on a validation dataset³. The input to each neural curve layer consists of the concatenation of a $H \times W \times 3$ image converted to the given

²Thanks to the scaling, CURL can easily mix representations, for example mapping saturation as a function of hue, whereas previous approaches are restricted to mapping like to like (e.g. red channel to red channel).

³An experimental study on the ordering of the colour spaces is presented in Section IV.

colour space, and a $H \times W \times C'$ subset of the features from the pixel-level block ($C' = C - 3$). For the luminance neural curve layer, the fully connected layer regresses the parameters of the L, a and b channel scaling curves. The scaling curves scale the pixel values in the L, a, b channels using Equation 1. The adjusted CIELab image is then converted back to RGB. This $H \times W \times 3$ RGB image is concatenated with the pixel-level block $H \times W \times C'$ feature map and fed into the second neural curve layer that learns three more curves, one for each channel of the RGB image. These curves are applied to the R, G, B channels to adjust the colours. Lastly, the $H \times W \times 3$ RGB image is converted to HSV space. HSV space separates the hue, saturation and value properties of an image. The HSV image is concatenated with the $H \times W \times C'$ feature map and used by the final curve layer to predict the four HSV space adjustment curves. The four HSV curves are for saturation-to-saturation mapping, hue-to-hue, value-to-value and hue-to-saturation mapping. The hue-to-saturation curve is particularly interesting as it permits the precise adjustment of the saturation of individual hues. These curves are applied to the HSV image and the HSV image is converted back to RGB space via a differentiable HSV to RGB conversion. A long skip connection links input and output. The differentiable colour spaces transformations required in the CURL block are readily implementable using e.g. the Kornia library [17].

C. The CURL Loss Function

The CURL loss function consists of three colour space-specific terms which seek to optimise different aspects of the predicted image, including the hue, saturation, luminance and chrominance. The loss is designed to control each of the colour-space specific transformations in CURL. The loss is minimised over a set of N image pairs $\{(I_i, \hat{I}_i)\}_{i=1}^N$, where I_i is the reference image and \hat{I}_i is the predicted image. The CURL loss is presented in Equation 2

$$\mathcal{L} = \sum_{i=1}^N \mathcal{L}_{hsv}^i + \mathcal{L}_{lab}^i + \mathcal{L}_{rgb}^i + \mathcal{L}_{reg}^i \quad (2)$$

where \mathcal{L}_{rgb} , \mathcal{L}_{lab} , \mathcal{L}_{hsv} are the various colour space loss terms, and \mathcal{L}_{reg} is a curve regularization loss. These terms are defined in more detail below.

HSV loss, \mathcal{L}_{hsv} : given the hue (angle) $H_i \in [0, 2\pi]$, saturation $S_i \in [0, 1]$ and value $V_i \in [0, 1]$ for image I_i , we compute \mathcal{L}_{hsv}^i in the conical HSV colour space,

$$\mathcal{L}_{hsv}^i = \omega_{hsv} (\|\hat{S}_i \hat{V}_i \cos(\hat{H}_i) - S_i V_i \cos(H_i)\|_1 + \|\hat{S}_i \hat{V}_i \sin(\hat{H}_i) - S_i V_i \sin(H_i)\|_1) \quad (3)$$

HSV is advantageous as it separates colour into useful components (hue, saturation, intensity). We believe this is one of the first applications of a differentiable HSV transform and loss in a deep network for the purposes of image enhancement.

CIELab loss, \mathcal{L}_{lab} : we compute the L_1 distance between the Lab values of the groundtruth and predicted images (Equation

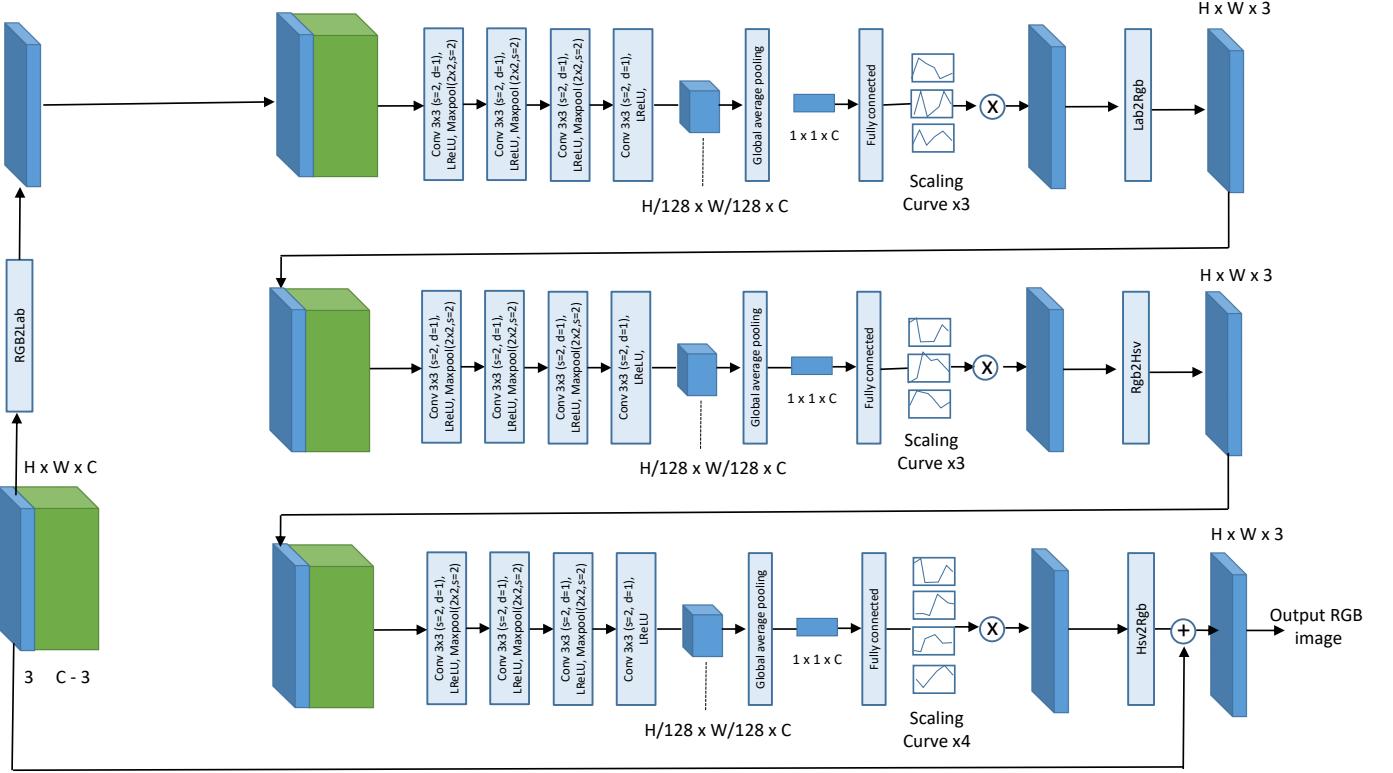


Fig. 5. **CURL block** refines the image from the backbone network to adjust luminance, colour and saturation. The image has luminance adjusted first, color (RGB) second, and saturation third using three piecewise linear curves. The output of this block is the final RGB image. C is number of feature maps, s stride, d dilation rate, H image height, W image width.

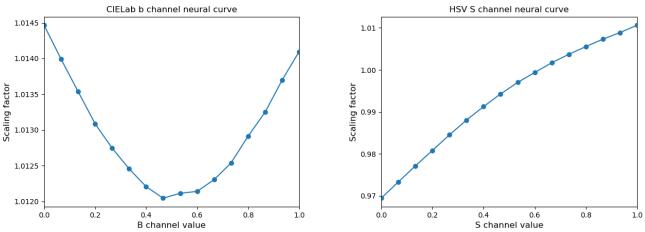


Fig. 6. Examples of learnt neural global adjustment curves for different colour spaces (CIELab and HSV). Retouching is often a subtle operation, mildly adjusting colour and luminance properties of an image produced by the backbone network. A unique adjustment curve is predicted per image, with the magnitude of the adjustment depending on how much retouching is required for that image.

4). In particular, the multi-scale structural similarity (MS-SSIM) [18] between the luminance channels of the ground truth and predicted images enforces a reproduction of the contrast, luminance and structure of the target image [8].

$$\mathcal{L}_{lab}^i = \omega_{lab} \|Lab(\hat{I}_i) - Lab(I_i)\|_1 + \omega_{ms-ssim} MS-SSIM(L(\hat{I}_i), L(I_i)) \quad (4)$$

where $Lab(\cdot)$ is a function that returns the CIELab Lab channels corresponding the RGB channels of the input image

and $L(\cdot)$ is a function that returns the L channel of the image in CIELab colour space.

RGB loss, \mathcal{L}_{rgb} : this term consists of L_1 distance on RGB pixels between the predicted and groundtruth images and a cosine distance between RGB pixel vectors (where I_i^j is three-element vector representing the RGB components of pixel j in the i -th image)

$$\mathcal{L}_{rgb}^i = \omega_{rgb} \|\hat{I}_i - I_i\|_1 + \omega_{cosine} \left(1 - \frac{1}{HW} \sum_{j=1}^{HW} \left(\frac{\hat{I}_i^j \cdot I_i^j}{\|\hat{I}_i^j\|_2 \|\hat{I}_i^j\|_2} \right) \right) \quad (5)$$

Curve regularization loss, \mathcal{L}_{reg} : to mitigate overfitting we regularize by penalizing the curvature:

$$\mathcal{L}_{reg}^i = \omega_{reg} \sum_{r=0}^{R-1} \sum_{m=0}^{M-3} (\Delta k_m^r - \Delta k_{m+1}^r)^2 \quad (6)$$

where Δk_m is the gradient of the line segment defined by knot points k_m, k_{m+1} , M is the number of knot points, and R denotes the number of curves.

Loss term weights: each loss term has an associated weight hyperparameter: $\omega_{rgb}, \omega_{hsv}, \omega_{lab}, \omega_{ms-ssim}, \omega_{reg}, \omega_{cosine}$. We empirically find that only the $\omega_{ms-ssim}$ term is sensitive to the particular dataset. The supplementary material presents details on our training configuration. Our experimental evaluation



Fig. 7. Qualitative effect of different combinations of terms in the CURL loss function on image quality. $\mathcal{L}_{rgb,no-cos}$ is the RGB loss without the cosine distance term. All terms are most effective, with obvious artefacts and colour distortions removed. See Section III-C for more detail.

(Section IV) presents evidence towards the necessity and contribution of each term, in relation to output image quality.

IV. EXPERIMENTAL RESULTS

Datasets: We evaluate CURL on three publicly available datasets: (i) **Samsung S7** [8] consists of 110, 12M pixel images of short-to-medium exposure RAW, RGB image pairs and medium-to-medium exposure RAW, RGB pairs. Following [8] we divide the dataset into 90 images for training, 10 for validation and 10 for testing. (ii) **MIT-Adobe5k-DPE** [3] contains 5,000 images taken on DSLR cameras and subsequently adjusted by an artist (Artist C). We follow the dataset pre-processing procedure of DeepPhotoEnhancer (DPE) [3]. The training dataset consists of 2,250 RGB image pairs. The RAW input images are processed into RGB by Lightroom. The groundtruth RGB images are generated by applying the adjustments of Artist C to the input. The dataset is divided into 2,250 training images and 500 test images. We randomly sample 500 validation images from the 2,250 training images. The images are re-sized to have a long-edge of 500 pixels. (iii) **MIT-Adobe5k-UPE** [1] MIT-Adobe5k dataset pre-processed as described in [1] (DeepUPE). The training dataset consists of 4,000 images of RGB image input, groundtruth pairs. The groundtruth are from Artist C. The dataset is divided into 4,500 training images and 500 test images. We randomly sample 500 images to serve as our validation dataset. The images are not re-sized and range from 6M pixel to 25M pixel resolution. **Evaluation Metrics:** Our image quality metrics are PSNR, SSIM and the perceptual quality aware metric LPIPS [19]. Hyperparameters were tuned on the held-out validation portion of each benchmark dataset. See supplementary material. **Backbone:** For the ablation studies in Tables IV-A we use the parameter heavier TED backbone network with an MSCA-skip connection at every level. This variant mostly leads to the highest image quality. For the comparison to the state-of-the-art in Section IV-B we use the parameter-light variant of TED with one MSCA-skip connection at the first level.

A. CURL Ablation Studies

Colour Spaces of CURL: Tables II, III present ablation studies on the inclusion of individual colour spaces and on the ordering of the colour spaces within the retouching block. In Table II we find that inclusion of all colour spaces (HSV, RGB, Lab) in the colour block leads to the best PSNR and SSIM, versus variants of the model that have only a single

TABLE II
ABLATION STUDY ON THE CURL GLOBAL IMAGE RETOUCHING LAYER. ALL COLOUR SPACES ARE NECESSARY IN CURL FOR BEST IMAGE QUALITY.

Architecture	PSNR↑	SSIM↑	LPIPS↓
TED+CURL (RGB only)	26.74	0.790	0.340
TED+CURL (HSV only)	25.88	0.780	0.308
TED+CURL (CIELab only)	26.98	0.788	0.323
TED+CURL (HSV, RGB, CIELab)	27.09	0.793	0.312

TABLE III
RESULTS FOR ALL PERMUTATIONS OF THE COLOUR SPACES IN OUR CURL BLOCK.

Ordering	PSNR (test)↑	SSIM (test)↑	PSNR (valid)↑	PSNR (valid)↑
HSV→RGB→LAB	26.20	0.779	26.48	0.765
RGB→HSV→LAB	26.83	0.786	26.23	0.770
LAB→RGB→HSV	27.09	0.793	26.56	0.771
LAB→HSV→RGB	26.37	0.784	26.44	0.757
RGB→LAB→HSV	25.32	0.761	26.31	0.760
HSV→LAB→RGB	26.53	0.787	26.67	0.769

colour space (Figure 8). Table III suggests that the ordering of the colour spaces is important, with the highest image quality attained with a Lab, RGB and then HSV adjustment.

Loss function terms: Equation 2 presents the CURL loss function. We perform an ablation study in Table IV on the various terms in the loss function. The highest image quality is attained with all loss function terms, demonstrating the need to constrain each of the three colour spaces appropriately with a dedicated loss term. Interestingly we find that coupling the the L_1 RGB loss with a cosine distance term gives a significant boost in image quality ($25.45 \rightarrow 27.09$ dB, $0.756 \rightarrow 0.793$ SSIM) compared to the using just the L_1 RGB loss term. In addition the regularization term (\mathcal{L}_{reg}) is important for highest image quality due to its role in constraining the flexibility of the neural retouching curves.

B. Comparison to state-of-the-art methods

We evaluate our method against competitive baseline models. Specifically our main state-of-the-art baselines are (i) **U-Net**: [3], [12] we use a U-Net architecture without the MSCA-connections, and broadly following the design of the generator architecture of DPE [3]. (ii) **DeepISP** [8]: we follow the architecture and experimental procedure of [8]. The thirty element RGB colour transformation matrix is initialized



Fig. 8. Qualitative effect of **different formulations of the CURL block**. We keep only one colour space (either CIELab, RGB, HSV) in the CURL block and compare the output versus keeping all three colour spaces.

TABLE IV

ABLATION STUDY ON THE **SAMSUNG S7** DATASET [8] FOR THE VARIOUS TERMS IN THE CURL LOSS FUNCTION. $rgb, no-\cos$ IS EQUATION 5 *without* THE COSINE TERM.

CURL Loss Terms	PSNR	SSIM
TED+CURL ($\mathcal{L}_{lab} + \mathcal{L}_{reg}$)	26.19	0.777
TED+CURL ($\mathcal{L}_{rgb} + \mathcal{L}_{reg}$)	26.52	0.785
TED+CURL ($\mathcal{L}_{rgb,no-\cos} + \mathcal{L}_{reg}$)	25.65	0.768
TED+CURL ($\mathcal{L}_{hsb} + \mathcal{L}_{reg}$)	25.90	0.777
TED+CURL ($\mathcal{L}_{hsb} + \mathcal{L}_{rgb} + \mathcal{L}_{reg}$)	26.61	0.792
TED+CURL ($\mathcal{L}_{lab} + \mathcal{L}_{hsb} + \mathcal{L}_{reg}$)	26.35	0.781
TED+CURL ($\mathcal{L}_{lab} + \mathcal{L}_{rgb} + \mathcal{L}_{reg}$)	25.88	0.767
TED+CURL ($\mathcal{L}_{lab} + \mathcal{L}_{hsb} + \mathcal{L}_{rgb}$)	26.32	0.768
TED+CURL ($\mathcal{L}_{lab} + \mathcal{L}_{hsb} + \mathcal{L}_{rgb,no-\cos} + \mathcal{L}_{reg}$)	25.45	0.756
TED+CURL ($\mathcal{L}_{lab} + \mathcal{L}_{hsb} + \mathcal{L}_{rgb} + \mathcal{L}_{reg}$)	27.09	0.793

TABLE V

MEDIUM-TO-MEDIUM EXPOSURE RAW TO RGB MAPPING RESULTS ON THE HELD-OUT TEST IMAGES OF THE **SAMSUNG S7** DATASET [8].

Architecture	PSNR↑	SSIM↑	LPIPS↓	# Params
TED+ CURL	27.04	0.794	0.320	1.4 M
TED	26.56	0.781	0.339	1.3 M
U-Net [13]	25.90	0.783	0.340	5.1 M
DeepISP [8]	26.51	0.794	0.326	3.9 M

using linear regression. (iii) **DPE**: [3] we evaluate against the supervised (paired data) version of DPE. (iv) **DeepUPE** [1] (v) **HDRNet** [15] (vi) **White-Box** [2].

Quantitative Comparison: (i) **Samsung S7 dataset**: We evaluate TED+CURL on the RAW-to-RGB mapping task for long-to-long exposure image pairs. TED+CURL is compared to U-Net [13] and DeepISP [8]. Results are presented in Table V. TED+CURL produces higher-quality images for both exposure settings compared to U-Net and outperforms DeepISP in terms of PSNR and LPIPS metrics. This suggests TED+CURL is a competitive model for replacing the traditional RAW-to-RGB ISP pipeline. (ii) **MIT-Adobe5k (DPE)**: Table VI presents the results on this dataset. TED+CURL uses $\sim 2.5 \times$ fewer parameters than DPE, yet is able to maintain the same SSIM score while boosting PSNR by $+0.17\text{dB}$. (iii) **MIT-Adobe5k (UPE)**: Table VII demonstrates that CURL is competitive with DeepUPE on this challenging high-resolution dataset. TED+CURL obtains a substantial $+1.16\text{dB}$ boost in PSNR, a reduction in LPIPS (lower is better) while retaining a

TABLE VI

PREDICTION QUALITY OF PHOTOGRAPHER C RETOUCHINGS FOR THE **MIT-ADOBEE 5K** [3] TEST IMAGES. RESULTS FOR OTHER ARCHITECTURES ARE EXTRACTED FROM [3].

Architecture	PSNR↑	SSIM↑	LPIPS↓	# Params
TED+CURL	24.04	0.900	0.583	1.4 M
DPED [6]	21.76	0.871	–	–
8RESBLK [20], [21]	23.42	0.875	–	–
FCN [22]	20.66	0.849	–	–
CRN [23]	22.38	0.877	–	–
U-Net [13]	22.13	0.879	–	–
DPE [3]	23.80	0.900	0.587	3.3 M

TABLE VII

AVERAGE PSNR, SSIM RESULTS FOR PREDICTING THE RETOUCHING OF PHOTOGRAPHERS ON THE 500 TESTING IMAGES FROM THE **MIT-ADOBEE 5K** DATASET. DATASET PRE-PROCESSED ACCORDING TO THE DEEPUPE PAPER [1]. THE PSNR AND SSIM OF OTHER ARCHITECTURES ARE EXTRACTED FROM [1].

Architecture	PSNR↑	SSIM↑	LPIPS↓	# Params
TED+CURL	24.20	0.880	0.108	1.4 M
HDRNet [15]	21.96	0.866	–	–
DPE [3]	22.15	0.850	–	3.3 M
White-Box [2]	18.57	0.701	–	–
Distort-and-Recover [24]	20.97	0.841	–	–
DeepUPE [1]	23.04	0.893	0.158	1.0 M

competitive SSIM. **Visual Comparison:** results showing our method output in comparison to DeepISP and DeepUPE are shown in Figures 9-10. CURL produces images with more pleasing colour and luminance compared to DeepISP and DeepUPE. Additional visual examples are presented in the supplementary material.

V. CONCLUSIONS

This paper introduced **CURL (CURve Layers)**, a novel neural block for image enhancement. CURL takes inspiration from artists/photographers and retouches images based on global image adjustment curves. The retouching curves are learnt automatically during training to adjust image properties by exploiting image representation in three different colours spaces (CIELab, HSV, RGB). The adjustments applied by these curves is moderated by novel multi-colour space loss function. In our experimental evaluation a encoder/decoder



Fig. 9. Examples images produced by **DeepISP** and **TED+CURL** on the Samsung S7 *Medium Exposure* dataset.

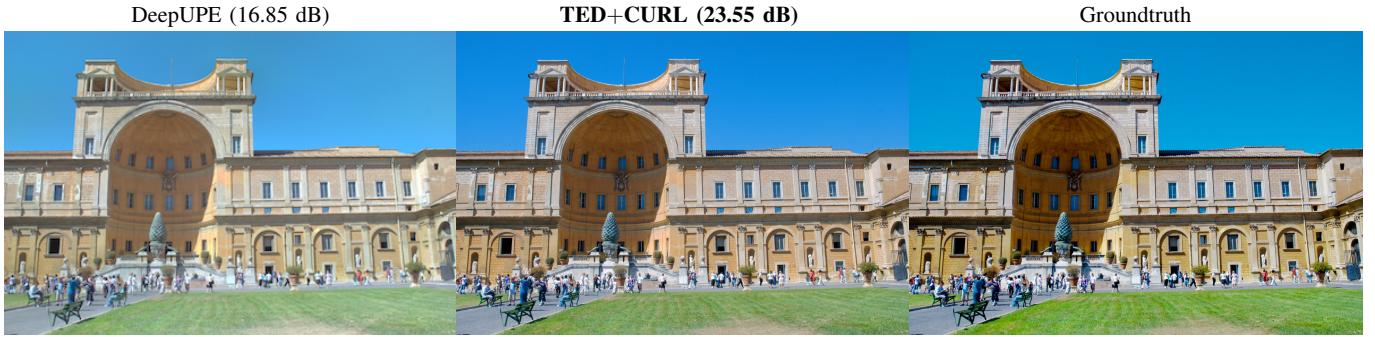


Fig. 10. Examples images produced by **DeepUPE** and **TED+CURL** on the *MIT-Adobe-UPE* dataset.

backbone augmented with the pluggable CURL block significantly outperformed the state-of-the-art across a suite of benchmark datasets. Future research will investigate per-image adaptive ordering of adjustments in the CURL block.

REFERENCES

- [1] R. Wang, Q. Zhang, C.-W. Fu, X. Shen, W.-S. Zheng, and J. Jia, “Underexposed photo enhancement using deep illumination estimation,” in *CVPR*, 2019.
- [2] 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. 26:1–17, May 2018.
- [3] 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,” June 2018.
- [4] 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 *CVPR*, 2020.
- [5] S. Bianco, C. Cusao, F. Piccoli, and R. Schettini, “Content-preserving tone adjustment for image enhancement,” in *CVPR Workshops*, 2019.
- [6] A. Ignatov, N. Kobyshev, R. Timofte, K. Vanhoey, and L. V. Gool, “Dslr-quality photos on mobile devices with deep convolutional networks,” in *ICCV*, 2017.
- [7] X. Zhang, Q. Chen, R. Ng, and V. Koltun, “Zoom to learn, learn to zoom,” in *CVPR*, 2019.
- [8] E. Schwartz, R. Giryes, and A. M. Bronstein, “DeepISP: Towards Learning an End-to-End Image Processing Pipeline.” *IEEE Transactions on Image Processing*, vol. 28, no. 2, pp. 912–923, 2019.
- [9] S. Gu, Y. Li, L. V. Gool, and R. Timofte, “Self-guided network for fast image denoising,” in *ICCV*, October 2019.
- [10] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Cunningham, A. Acosta, A. P. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi, “Photo-realistic single image super-resolution using a generative adversarial network,” in *CVPR*, 2017, pp. 105–114.
- [11] Z. Liang, J. Cai, Z. Cao, and L. Zhang, “Cameranet: A two-stage framework for effective camera isp learning,” in *ArXiv*, 2019.
- [12] C. Chen, Q. Chen, J. Xu, and V. Koltun, “Learning to see in the dark,” in *CVPR*, 2018.
- [13] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” in *MICCAI*, 2015, pp. 234–241.
- [14] S. Nam and S. J. Kim, “Modelling the scene dependent imaging in cameras with a deep neural network,” *ICCV*, pp. 1726–1734, 2017.
- [15] 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 (TOG)*, vol. 36, no. 4, p. 118, 2017.
- [16] 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,” *Computer Graphics Forum*, vol. 37, 2018.
- [17] E. Riba, D. Mishkin, D. Ponsa, E. Rublee, and G. Bradski, “Kornia: an open source differentiable computer vision library for pytorch,” in *WACV*, March 2020.
- [18] Z. Wang, E. Simoncelli, and A. Bovik, “Multiscale Structural Similarity for Image Quality Assessment,” in *The Thirtieth Asilomar Conference on Signals, Systems and Computers*, 2003.
- [19] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, “The unreasonable effectiveness of deep features as a perceptual metric,” in *CVPR*, 2018.
- [20] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, “Unpaired image-to-image translation using cycle-consistent adversarial networks,” in *ICCV*, 2017.
- [21] M.-Y. Liu, T. Breuel, and J. Kautz, “Unsupervised image-to-image translation networks,” in *NeurIPS*, 2017, pp. 700–708.
- [22] Q. Chen, J. Xu, and V. Koltun, “Fast image processing with fully-convolutional networks,” in *ICCV*, 2017, pp. 2497–2506.
- [23] Q. Chen and V. Koltun, “Photographic image synthesis with cascaded refinement networks,” in *ICCV*, 2017, pp. 1511–1520.
- [24] J. Park, J.-Y. Lee, D. Yoo, and I. So Kweon, “Distort-and-recover: Color enhancement using deep reinforcement learning,” in *ICCV*, 2018.