DIFAR: Deep Image Formation and Retouching

Sean Moran, Gregory Slabaugh
Huawei Noah’s Ark Lab
{sean.moran, gregory.slabaugh}@huawei.com

Figure 1: Underexposed image enhancement. Given (a) poorly exposed image, DIFAR (c) produces an image with pleasing contrast and colour better matching the groundtruth (d) compared to the state-of-the-art DeepUPE model [42] (b).

Abstract

We present a novel neural network architecture for the image signal processing (ISP) pipeline. In a camera system, the ISP is a critical component that forms a high quality RGB image from RAW camera sensor data. Typical ISP pipelines sequentially apply a complex set of traditional image processing modules, such as demosaicing, denoising, tone mapping, etc. We introduce a new deep network that replaces all these modules, dubbed Deep Image Formation And Retouching (DIFAR). DIFAR introduces a multi-scale context-aware pixel-level block for local denoising/demosaicing operations and a retouching block for global refinement of image colour, luminance and saturation. DIFAR can also be trained for RGB to RGB image enhancement. DIFAR is parameter-efficient and outperforms recently proposed deep learning approaches in both objective and perceptual metrics, setting new state-of-the-art performance on multiple datasets including Samsung S7 [38] and MIT-Adobe 5k [6].

1. Introduction

Image quality is of fundamental importance in any imaging system, including DSLR and smartphone cameras. At the imaging sensor, RAW data is normally captured on a color filter array (such as the well-known Bayer pattern) where at each pixel, only a red, green, or blue color is available. This mosaiced RAW data suffers from noise, vignetting, lack of white balance, and many other defects and additionally has a high dynamic range.

The camera’s image signal processing (ISP) pipeline is responsible for forming a high quality RGB image with minimal noise, pleasing colors, sharp detail, and good contrast from the degraded RAW data. In most cases, the ISP is realised as a modular sequence of traditional image signal processing algorithms (Figure 2) each responsible for a single well-defined image operation (e.g. denoising, demosaicing, tone mapping). The modules are numerous and highly engineered, easily exceeding twenty in modern ISPs. As a result, it is difficult and expensive to design, develop, and tune the ISP pipeline due to the complex dependencies between the modules.

A single neural network trained end-to-end that directly transforms RAW data into an RGB image has numerous advantages including simplicity, efficiency, and most importantly, quality. Currently the state-of-the-art in many of the individual modules in an ISP (such as image denoising, automatic white balance, dynamic range compression) is set by deep neural networks. However, concatenating a large set of deep networks is impractical for efficient inference on a device like a smartphone, and end-to-end training with many modules is complicated; instead, a single network transforming from RAW to RGB is preferred. Efficient on-board neural image processing software can correct for lim-
iterations of the sensor, producing an RGB image that has rich colour, texture, contrast and detail. This is particularly important for smartphones, which must include smaller and less capable imaging sensors due to form factors.

Given the importance of the ISP on image quality, surprisingly, the problem of RAW to RGB transformation using a deep neural network has only received limited attention in the literature, rendering it an under-researched topic. Recently the DeepISP [38] approach first demonstrated proof-of-concept for a deep network to replace the entire ISP, and work by Chen et al. [7] addressed RAW to RGB transformation in low light (less than 5 lux). However, it is an open question how far image quality can advance using deep RAW to RGB networks.

Drawing inspiration from this related work, we propose DIFAR (Deep Image Formation And Retouching), a novel neural image processing pipeline for image formation and retouching. DIFAR includes a pixel-level block for low level operations like denoising and demosaicing. Similar to [7], our 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. Similar to [38], DIFAR includes global image transformations to adjust colors and brightness. However, unlike [38], DIFAR introduces neural curve layers, which more expressively adjust image properties in a controlled but human-interpretable way. Unlike [7] or [38], DIFAR leverages multiple color spaces to boost performance. Although DIFAR was designed to handle the RAW to RGB mapping, it also excels at RGB to RGB image enhancement. Our contributions in this paper are four-fold:

- **State-of-the-art end-to-end deep neural network for the entire ISP**: A new neural network architecture incorporating a pixel-level block for local processing including demosaicing, denoising, brightness, and luminance correction and a retouching block that globally adjusts the image luminance, saturation and colour. DIFAR produces state-of-the-art results on several public datasets.

- **MSCA connections** in the pixel level block: Our pixel-level block has an encoder/decoder structure but replaces standard skip connections with a novel multi-scale context awareness connection, enriching the information available to the decoder.

- **Retouching block**: We introduce a neural network curve layer, that learns a piece-wise linear scaling curve to globally adjust image properties in a human-interpretable manner, inspired by the Photoshop curves tool.

- **Multiple colour spaces**: We apply sequential differentiable transformations of the image in different colour spaces (Lab, RGB, HSV) guided by a novel multi-colour space loss function.

2. Related Work

**Single-stage ISP operations.** Image denoising is a fundamental problem in the ISP and remains an open problem. A detailed review of denoising methods can be found in [3]. Traditional methods [15, 33, 41, 5, 13] seek noise removal while preserving edges. More recent work casts denoising as a regression problem addressed with deep learning [46, 30, 40, 27, 47, 8, 1] showing much promise to de-noise using convolutional neural networks (CNNs). Image demosaicing seeks to recover a full colour RGB image from an incomplete set of colour samples arranged on a colour filter array. The task of demosaicing is essentially one of interpolation. Often coupled with denoising, the state-of-the-art in demosaicing is set by deep learning using CNNs [16, 25] and conditional GANs [14]. Other fundamental ISP operations include automatic white balance [20, 2, 35, 4] and dynamic range compression [17, 44].

**Combined operations.** A recent trend in the literature is to combine tasks to solve joint problems. In addition to joint denoising and demosaicing mentioned in the previous paragraph, authors have explored solutions for joint demosaicing and super-resolution [37], demosaicing, denoising, and super-resolution [34], super-resolution and tone mapping [24], demosaicing and deblurring [12].

**Modeling the full ISP.** Further combination of multiple ISP tasks ultimately leads to modeling the entire ISP. Currently the literature is sparse on deep learning methods that specifically aim to replace the full pipeline. The most relevant related work is the DeepISP model of [38], 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) [18] which relies extensively on pixel shuffling [26] operations, and CameraNet [28] that decomposes...
the problem into two subproblems of restoration and enhancement. Another related work is [7] which also learns a RAW to RGB mapping, however for extremely dark images using a simple UNet-style architecture [36].

Image enhancement. In an ISP, image enhancement typically relates to brightness and colour adjustment. Hu et al. [19] design a photo retouching approach (WhiteBox) using reinforcement learning and GANs to apply a sequence of filters to an image. Deep Photo Enhancer (DPE) [11] is a GAN-based architecture with a UNet-style generator for RGB image enhancement that produces state-of-the-art results on the popular MIT-Adobe5K benchmark dataset. Arguably, the state-of-the-art in digital photography is achieved with DSLR, which due to the large aperture and lenses can produce higher quality photographs than what is achievable with a smartphone camera. Related work enhances smartphone images by learning a mapping between smartphone images and DSLR [21, 22, 49] photographs using deep learning.

3. The DIFAR Neural Network

DIFAR is an end-to-end trainable neural network that consists of two blocks as shown in Figure 3. The pixel-level block is an encoder-decoder neural network (Section 3.1) for local processing. The output of the pixel level block is passed to the retouching block (Section 3.2), employing a novel neural curve layer that globally adjusts image properties.

3.1. Pixel-Level Block: Localised Pixel Processing

The DIFAR pixel-level block is based on a UNet [36] backbone. The pixel-level block performs local pixel processing including the tasks of denoising and demosaicing. We adopt an encoder-decoder architecture with symmetric skip connections. The UNet has proven to be a capable network for image enhancement tasks as demonstrated by recent related work [11, 7].

However, DIFAR proposes a new type of skip connection (Figure 4) we call multi-scale contextual awareness (MSCA) connection, inspired by [31]. The MSCA-connection fuses multiple different contextual image features, combining global and mid-level 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. 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. As argued by [17, 7, 31] it is important to consider global and mid-level context when making local pixel adjustments to reduce spatial inconsistencies in the predicted image. MSCA-connections reduce colour inconsistency artifacts, see Figure 5) while also reducing the number of trainable parameters.

The backbone U-Net differs depending on the type of input data (RAW or RGB). For our RAW to RGB experiments we modify the standard U-Net 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 [26] 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 U-Net 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 up-
Conv 3x3 (s=2, d=1), LReLU, Maxpool
Conv 3x3 (s=2, d=1), LReLU, Maxpool
Conv 3x3 (s=2, d=1), LReLU, Maxpool
Conv 3x3 (s=2, d=1), LReLU, Maxpool
Global average pooling
Fully connected

Figure 4: **Multi-scale contextual awareness connection** that fuses multiple levels of image context (global, mid, local) to deliver more contextually relevant features for the expanding path (Section 3.1).

Figure 5: **Left:** U-Net. **Right:** U-Net+MSCA connection. Adding an MSCA connection to a vanilla U-Net model can reduce colour bleeding artifacts. Please zoom in for details.

sampling operation [26] produces a result with output shape $H \times W \times C$, where $C$ is the number of feature maps, and is passed over to the retouching block for further processing. Our RGB to RGB U-Net network is similar to the RAW to RGB network, with the exception that we remove the pixel shuffling operations. The long skip connection remains, but in this case the 3 channel input image is added to the output, rather than a shuffled version.

### 3.2. Retouching Block: Global Image Adjustment

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 a feature maps that are used as input to each neural curve layer. Our proposed neural curve layer block is shown in Figure 6 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 3). The curve adjusts the predicted image ($\hat{I}_i \in [0, 1]$) by scaling pixels with the formula presented in Equation 1.

$$S(\hat{I}_{ijl}) = k_0 + \sum_{m=0}^{M-1} (k_{m+1} - k_m)\delta(M\hat{I}_{ijl} - m)$$

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, $\hat{I}_{ijl}$ is the $j$-th pixel value in the $l$-th color 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\(^2\). Examples of neural curves learnt by one instance of our model are shown in Figure 7. The feature extraction block of the neural curve layer (Figure 6) accepts a $W \times H \times C$ feature map, passing it to a group of blocks each consisting of a convolutional layer with 3x3 kernels of stride 2, and 2x2 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 6, 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.

\(^2\)We set $M=16$ in this paper based on a parameter sweep on a validation dataset.
Figure 6: **Global retouching block** of DIFAR. The retouching block refines the image from the pixel-level network to adjust luminance, colour and saturation using our neural curve formulation. The image has luminance adjusted first, colour (RGB) second, and saturation third using three piecewise linear curves. The output of this block is the final RGB image, completing the RAW to RGB mapping. The supplementary material investigates different arrangements of this general architecture.

Figure 7: Examples of learnt neural global adjustment curves for different colour spaces (CIELab and HSV). (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 dataset\(^3\). The input to each neural curve layer consists of the concatenation of a \(H \times W \times 3\) image converted to the given 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 \(L, a, b\) 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 which is used by the final curve layer to predict the four HSV space adjustment curves. 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.

3.3. Loss Function

The DIFAR 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 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 DIFAR loss is
Figure 8: Qualitative effect of different combinations of terms in the DIFAR loss function on image quality. $L_{rgb,nc}$ is the RGB loss without the cosine distance term. All terms are most effective, with obvious artefacts and colour distortions removed. See Section 3.3 for more detail.

Presented in Equation 2

$$L = \sum_{i=1}^{N} L_{rgb}^i + L_{lab}^i + L_{hsv}^i + L_{reg}^i$$  \hspace{1cm} (2)

where $L_{rgb}$, $L_{lab}$, $L_{hsv}$, $L_{reg}$ are the various color space loss terms, and $L_{reg}$ is a curve regularization loss. These terms are defined in more detail below.

**HSV loss, $L_{hsv}$:** 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 $L_{hsv}^i$ in the conical HSV colour space,

$$L_{hsv}^i = \omega_{hsv} ||S_i \hat{V} \cos(H_i) - S_i V \sin(H_i)||$$  \hspace{1cm} (3)

This loss is adapted from the colour similarity function of VisualSeek [39]. HSV is advantageous as it separates colour into useful components (hue, saturation, intensity).

**CIELab loss, $L_{lab}$:** we follow [38] and compute the $L_1$ distance between the Lab values of the groundtruth and predicted images (Equation 4). In particular, the multi-scale structural similarity (MS-SSIM) [43] between the luminance channels of the groundtruth and predicted images enforces a reproduction of the contrast, luminance and structure of the target image [38],

$$L_{lab}^i = \omega_{lab} ||\hat{L}(I_i) - L(I_i)||$$  \hspace{1cm} (4)

$$+ \omega_{ms-ssim} MS-SSIM(L(I_i), \hat{L}(I_i))$$

where $L(\cdot)$ is a function that returns the CIELab Lab channels corresponding the RGB channels of the input image and $\hat{L}(\cdot)$ is a function that returns the L channel of the image in CIELab colour space.

**RGB loss, $L_{rgb}$:** this term follows [31] and consists of $L_1$ distance on RGB pixels between the predicted and groundtruth images and a cosine distance between RGB pixel vectors (where $I^j_i$ is three-element vector representing the RGB components of pixel $j$ in the $i$-th image)

$$L_{rgb}^i = \omega_{rgb} ||\hat{I}^j - I^j|| + \omega_{cosine}(1 - \frac{1}{K} \sum_{j=1}^{K} (\frac{\hat{I}^j I^j}{||\hat{I}^j||_2 ||I^j||_2}))$$  \hspace{1cm} (5)

**Curve regularization loss, $L_{reg}$:** to mitigate overfitting we regularize the neural curves with a smoothness prior by penalizing the squared difference between each of the $L$ piecewise linear segments, each denoted by $k_j$.

$$L_{reg}^i = \omega_{reg} \sum_{j=0}^{M-1} (k_{j+1} - k_j)^2$$  \hspace{1cm} (6)

**Loss term weights:** each loss term has an associated weight hyperparameter: $\omega_{rgb}$, $\omega_{hsv}$, $\omega_{lab}$, $\omega_{ms-ssim}$, $\omega_{reg}$, $\omega_{cosine}$. In our experimental evaluation we empirically find that only the $\omega_{ms-ssim}$ term is sensitive to the particular dataset. The remaining terms are largely dataset insensitive and we set $\omega_{rgb}=1$, $\omega_{cosine}=1$, $\omega_{lab}=1$, $\omega_{hsv}=1$, $\omega_{reg}=10^{-5}$ in all of our experiments across all benchmark datasets. In our experimental evaluation we find that all terms are necessary to attain the highest quality output images.

4. Experimental Results

**Datasets:** We evaluate DIFAR on three publicly available benchmark datasets: (i) **Samsung S7** [38] consists of 110, 12M pixel images of short-to-medium exposure RAW, RGB image pairs and medium-to-medium exposure RAW, RGB pairs. Following [38] we divide the dataset into 90 images for training, 10 for validation and 10 for testing. (ii) **MIT-Adobe5k-DPE** [38] contains 5,000 images taken on a variety of DLSR cameras and subsequently adjusted by an artist (Artist C). We follow the dataset pre-processing procedure of DeepPhotoEnhancer (DPE) [11]. 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 resized to have a long-edge of 500 pixels. (iii) **MIT-Adobe5k-UPe** [42] MIT-Adobe5k dataset pre-processed as described in [42] (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 [48]. Hyperparameters were tuned on the held-out validation portion of each benchmark dataset.

### 4.1. Comparison to state-of-the-art methods

We evaluate DIFAR against competitive baseline models on the RAW to RGB and RGB to RGB image enhancement tasks. Specifically our main state-of-the-art baselines are (i) **U-Net:** [11, 7] we use a U-Net architecture without the MSCA-connections, and broadly following the design of the generator architecture of DPE [11]. (ii) **DeepISP** [38]: we follow the architecture and experimental procedure of [38]. The thirty element RGB colour transformation matrix is initialised using linear regression. (iii) **DPE:** [11] we evaluate against the supervised (paired data) version of DPE. (iv) **DeepUPE** [42] (v) **HDRNet** [17] (vi) **White-Box** [19].

**Quantitative Comparison:** (i) **Samsung S7 dataset:** We evaluate DIFAR on the RAW to RGB mapping task for both short-to-long and long-to-long exposure image pairs. DIFAR is compared to the U-Net [36] and DeepISP [38] baselines. Results are presented in Tables 1-2. DIFAR produces higher-quality images for both exposure settings compared to U-Net and outperforms DeepISP in terms of the PSNR and LPIPS metrics. This suggests DIFAR is a competitive model for replacing the traditional RAW to RGB ISP pipeline. (ii) **MIT-Adobe5k (DPE):** Table 3 presents the results on the MIT-Adobe5K (DPE) dataset. DIFAR uses almost \(2.5\times\) less parameters than DPE, but is able to maintain the same SSIM score while boosting the PSNR by \(+0.17\)dB. (iii) **MIT-Adobe5k (UPE):** Table 4 presents the results on the MIT-Adobe5K (UPE) dataset. PSNR by \(+1.16\)dB boost in PSNR, a \(-0.053\) change in the LPIPS metric (lower is better) while remaining competitive in terms of SSIM.

**Visual Comparison:** We show visual results from the output of DIFAR and compared to the output of DeepISP and DeepUPE in Figures 9-10. DIFAR produces images with more pleasing colour and luminance compared to the DeepISP and DeepUPE baselines. Additional visual examples are presented in the supplementary material.

### 4.2. Discussion

**Ablation Studies** (i) **Colour blocks:** In Table 5 we evaluate the importance of the three colour blocks in the retouching layer. We compare the DIFAR model with all three colours blocks (RGB, HSV, Lab) in the retouching layer versus variants of the model with just one of the available blocks. A loss function that operates in only one colour space in e.g. RGB space is suboptimal. (ii) **Loss function terms:** Equation 2 presents the DIFAR loss function. We perform an ablation study (full results in the supplementary material, Table 7) 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

---

**Table 1:** Medium-to-medium exposure RAW to RGB mapping results on the held-out test images of the **Samsung S7 dataset** [38].

| Architecture                     | PSNR↑ | SSIM↑ | LPIPS↓ | # Parameters |
|----------------------------------|-------|-------|--------|--------------|
| DIFAR (MSCA, level 1)            | 27.04 | 0.794 | 0.320  | 1.3 M        |
| DIFAR                            | 27.09 | 0.793 | 0.312  | 3.3 M        |
| U-Net (large) [36]               | 25.00 | 0.783 | 0.340  | 5.1 M        |
| DeepISP (small) [38]             | 26.50 | 0.795 | 0.335  | 630 K        |
| DeepISP (large) [38]             | 26.51 | 0.794 | 0.326  | 3.9 M        |

**Table 2:** Short-to-medium exposure RAW to RGB mapping results on the held-out test images of the **Samsung S7 dataset** [38].

| Architecture                     | PSNR↑ | SSIM↑ | LPIPS↓ | # Parameters |
|----------------------------------|-------|-------|--------|--------------|
| DIFAR (MSCA, level 1)            | 26.52 | 0.788 | 0.353  | 1.3 M        |
| DIFAR                            | 26.73 | 0.790 | 0.362  | 3.3 M        |
| U-Net (large) [36]               | 26.01 | 0.782 | 0.395  | 5.1 M        |
| DeepISP (small) [38]             | 26.11 | 0.789 | 0.372  | 630 K        |
| DeepISP (large) [38]             | 26.37 | 0.792 | 0.360  | 3.9 M        |

**Table 3:** Results for predicting the retouching of photographer C on the 500 testing images from the **MIT-Adobe5K dataset**. Dataset pre-processed according to the DPE paper [11].

| Architecture                     | PSNR↑ | SSIM↑ | LPIPS↓ | # Parameters |
|----------------------------------|-------|-------|--------|--------------|
| DIFAR (MSCA, level 1)            | 23.97 | 0.900 | 0.583  | 1.3 M        |
| DIFAR                            | 23.71 | 0.900 | 0.587  | 3.3 M        |
| DPE [38]                         | 21.76 | 0.871 | –      | –            |
| 8RESBLK [50, 29]                 | 23.42 | 0.875 | –      | –            |
| FCRN [10]                       | 20.66 | 0.849 | –      | –            |
| CRN [9]                          | 22.38 | 0.877 | –      | –            |
| U-Net [36]                       | 22.13 | 0.879 | –      | –            |
| DPE [11]                         | 23.80 | 0.900 | 0.587  | 3.3 M        |

**Table 4:** Average PSNR/SSIM results for predicting the retouching of photographers on the 500 testing images from the **MIT-Adobe5K dataset**. Dataset pre-processed according to the DeepUPE paper [42].

| Architecture                     | PSNR↑ | SSIM↑ | LPIPS↓ | # Parameters |
|----------------------------------|-------|-------|--------|--------------|
| DIFAR (MSCA, level 1)            | 24.20 | 0.880 | 0.108  | 1.3 M        |
| DIFAR                            | 23.97 | 0.879 | 0.105  | 3.3 M        |
| HDRNet [17]                      | 21.96 | 0.866 | –      | –            |
| DPE [11]                         | 22.15 | 0.850 | –      | –            |
| White-Box [19]                   | 18.57 | 0.701 | –      | –            |
| Distort-and-Recover [32]         | 20.97 | 0.841 | –      | –            |
| DeepUPE [42]                     | 23.04 | 0.893 | 0.158  | 1.0 M        |
Table 5: Ablation study on the retouching layer of DIFAR.

| Architecture | PSNR↑ | SSIM↑ | LPIPS↓ |
|--------------|-------|-------|--------|
| DIFAR (with RGB, w/o HSV, w/o CIELab) | 26.74 | 0.790 | 0.340 |
| DIFAR (with HSV, w/o RGB, w/o CIELab) | 25.88 | 0.780 | **0.308** |
| DIFAR (with CIELab, w/o RGB, w/o HSV) | 26.98 | 0.788 | 0.323 |
| DIFAR (with HSV, with RGB, with CIELab) (MSCA, level 1) | 27.04 | **0.794** | 0.320 |
| DIFAR (with HSV, with RGB, with CIELab) | 27.09 | 0.793 | 0.312 |

5. Conclusions

This paper introduced DIFAR (Deep Image Formation And Retouching), a novel deep architecture for the ISP or image enhancement. DIFAR introduces a multi-scale contextual awareness (MSCA) skip connection that markedly improves the image quality obtained from a U-Net backbone network, while also drastically reducing the required number of trainable parameters. DIFAR takes inspiration from artists/photographers and applies a retouching layer to refine images based on learnt 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 adjustment applied by these curves is moderated by novel multi-colour space loss function. In our experimental evaluation DIFAR significantly outperformed the state-of-the-art across a suite of benchmark datasets. Future research will investigate weakly supervised methods that could ease the burden of collecting paired training data. A natural step in this direction would instead allowing training with unpaired image sets [51].

References

[1] S. Anwar and N. Barnes. Real image denoising with feature attention. In The IEEE International Conference on Computer Vision (ICCV), October 2019.
[2] J. T. Barron and Y.-T. Tsai. Fast fourier color constancy. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2

[3] M. Bertalmio. Denoising of Photographic Images and Video: Fundamentals, Open Challenges and New Trends. Springer, 2018. 2

[4] S. Bianco and C. Cusano. Quasi-unsupervised color constancy. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

[5] A. Buades. A non-local algorithm for image denoising. In *CVPR*, 2005. 2

[6] V. Bychkovsky, S. Paris, E. Chan, and F. Durand. Learning photographic global tonal adjustment with a database of input / output image pairs. In *The Twenty-Fourth IEEE Conference on Computer Vision and Pattern Recognition*, 2011. 1

[7] C. Chen, Q. Chen, J. Xu, and V. Koltun. Learning to see in the dark. In *CVPR*, 2018. 2, 3, 7

[8] C. Chen, Z. Xiong, X. Tian, Z.-J. Zha, and F. Wu. Real-world Image Denoising with Deep Boosting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019. 2

[9] Q. Chen and V. Koltun. Photographic image synthesis with cascaded refinement networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1511–1520, 2017. 7

[10] Q. Chen, J. Xu, and V. Koltun. Fast image processing with fully-convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2497–2506, 2017. 7

[11] 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 *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 3, 6, 7, 11, 15, 23, 24

[12] Z. Chi, X. Shu, and X. Wu. Joint demosaicking and blind deblurring using deep convolutional neural network. In *International Conference on Image Processing*, 2019. 2

[13] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian. Image denoising by sparse 3d transform-domain collaborative filtering. In *IEEE Transactions on Image Processing*, IEEE, 2007. 2

[14] W. Dong, M. Yuan, X. Li, and G. Shi. Joint demosaicing and denoising with perceptual optimization on a generative adversarial network. *arXiv preprint arXiv:1802.04723*, 2018. 2

[15] D. L. Donoho. Denoising via soft thresholding. *IEEE Transactions on Information Theory*, 41(3), 1995. 2

[16] M. Gharbi, G. Chaurasia, S. Paris, and F. Durand. Deep Joint Demosaicking and Denoising. *ACM Transactions on Graphics (TOG)*, 35(8), 2016. 2

[17] 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)*, 36(4):118, 2017. 2, 3, 7

[18] S. Gu, Y. Li, L. V. Gool, and R. Timofte. Self-guided network for fast image denoising. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019. 2

[19] Y. Hu, H. He, C. Xu, B. Wang, and S. Lin. Exposure: A white-box photo post-processing framework. *arXiv preprint arXiv:1709.09602*, 2018. 3, 7

[20] Y. Hu, B. Wang, and S. Lin. Fc4: Fully convolutional color constancy with confidence-weighted pooling. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2

[21] A. Ignatov, N. Kobyshchev, R. Timofte, K. Vanhoey, and L. V. Gool. Dslr-quality photos on mobile devices with deep convolutional networks. In *ICCV*, 2017. 3

[22] A. Ignatov, N. Kobyshchev, R. Timofte, K. Vanhoey, and L. Van Gool. Wesp: Weakly supervised photo enhancer for digital cameras. *arXiv preprint arXiv:1709.01118*, 2017. 3

[23] A. Ignatov, N. Kobyshchev, R. Timofte, K. Vanhoey, and L. Van Gool. Dslr-quality photos on mobile devices with deep convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3277–3285, 2017. 7, 23, 24

[24] S. Y. Kim, J. Oh, and M. Kim. Deep sr-itm: Joint learning of super-resolution and inverse tone-mapping for 4k uhd hdr applications. In *International Conference on Computer Vision (ICCV)*, 2019. 2

[25] F. Kokkinos and S. Lefkimmiatis. Iterative Joint Image Demosaicking and Denoising using a Residual Denoising Network. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(8), 2019. 2

[26] 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 *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 105–114, 2017. 2, 3

[27] S. Lefkimmiatis. Universal denoising networks: A novel cnn architecture for image denoising. In *CVPR*, 2018. 2

[28] Z. Liang, J. Cai, Z. Cao, and L. Zhang. Cameranet: A two-stage framework for effective camera isp learning. In *ArXiv*, 2019. 2

[29] M.-Y. Liu, T. Breuel, and J. Kautz. Unsupervised image-to-image translation networks. In *Advances in neural information processing systems*, pages 700–708, 2017. 7

[30] X. Mao, C. Shen, and Y.-B. Yang. Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections. In *Advances in neural information processing systems*, pages 2802–2810, 2016. 2

[31] 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. *CoRR*, abs/1803.02266, 2018. 3, 6
[32] J. Park, J.-Y. Lee, D. Yoo, and I. So Kweon. Distort-and-recover: Color enhancement using deep reinforcement learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5928–5936, 2018. 7

[33] P. Perona and J. Malik. Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(7), 1990. 2

[34] G. Qian, J. Gu, J. S. Ren, C. Dong, F. Zhao, and J. Lin. Trinity of pixel enhancement: a joint solution for demosaicking, denoising and super-resolution. In *ArXiv*, 2019. 2

[35] Y. Qian, J.-K. Kamarainen, J. Nikkanen, and J. Matas. On finding gray pixels. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

[36] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Imaging Computing and Computer-Assisted Intervention (MICCAI)*, volume 9351 of LNCS, pages 234–241. Springer, 2015. (available on arXiv:1505.04597 [cs.CV]). 3, 7, 15

[37] Ruofan, R. Achanta, and S. Susstrunk. Deep residual network for joint demosaicing and super-resolution. In *Color and Imaging Conference, 26th Color and Imaging Conference Final Program and Proceedings*, 2018. 2

[38] E. Schwartz, R. Giryes, and A. M. Bronstein. DeepISP: Towards Learning an End-to-End Image Processing Pipeline. *IEEE Transactions on Image Processing*, 28(2):912–923, 2019. 1, 2, 6, 7, 14, 15, 17, 18

[39] J. R. Smith and S. fu Chang. Visualseek: a fully automated content-based image query system. pages 87–98, 1996. 6

[40] Y. Tai, J. Yang, X. Liu, and C. Xu. Memnet: A persistent memory network for image restoration. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4539–4547, 2017. 2

[41] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. In *ICCV*, 1998. 2

[42] R. Wang, Q. Zhang, C.-W. Fu, X. Shen, W.-S. Zheng, and J. Jia. Underexposed photo enhancement using deep illumination estimation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 1, 6, 7, 15, 21, 22

[43] Z. Wang, E. Simoncelli, and A. Bovik. Multiscale Structural Similarity for Image Quality Assessment. In *The Thirty-Seventh Asilomar Conference on Signals, Systems and Computers*, 2003. 6

[44] Q. Yan, D. Gong, Q. Shi, A. van den Hengel, C. Shen, I. Reid, and Y. Zhang. Attention-guided network for ghost-free high dynamic range imaging. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

[45] Y. Yu, M. Abadi, P. Barham, E. Brevdo, M. Burrows, A. Davis, J. Dean, S. Ghemawat, T. Harley, P. Hawkins, M. Isard, M. Kudlur, R. Monga, D. Murray, and X. Zheng. Dynamic control flow in large-scale machine learning. In *Proceedings of the Thirteenth EuroSys Conference*, EuroSys ’18, pages 18:1–18:15, New York, NY, USA, 2018. ACM. 14

[46] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE Transactions on Image Processing*, 26(7):3142–3155, 2017. 2

[47] K. Zhang, W. Zuo, and L. Zhang. FFDNet: Toward a Fast and Flexible Solution for CNN-Based Image Denoising. *IEEE Transactions on Image Processing*, 27(9), 2018. 2

[48] 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. 7

[49] X. Zhang, Q. Chen, R. Ng, and V. Koltun. Zoom to learn, learn to zoom. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 3

[50] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017. 7

[51] J. Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, 2017. 8
Figure 11: Effect of the number of knot points in a neural curve on the image quality (PSNR/SSIM). Maximum quality reached at 16 knot points. Results measured on the Samsung held-out validation dataset.

6. Supplementary material

This document provides additional material on DIFAR. In particular, we cover:

- DIFAR Pixel-Level Block Architecture (Section 6.1)
- MSCA-skip connection ablation study (Section 6.2)
- Loss function ablation study (Section 6.3)
- Number of Knot Points for the Neural Curves (Section 6.4)
- Variations of the retouching block architecture (Section 6.5)
- Differentiability of the piecewise linear scaling curve formulation (Section 6.6)
- Differentiability of the colour space conversion operators (Section 6.7)
- Limitations of DIFAR (Section 6.10)
- Additional qualitative visual examples (Section 6.11)

6.1. DIFAR Pixel-Level Block Architecture

The RAW to RGB and RGB to RGB pixel-level block of DIFAR is modelled as a U-Net, broadly following the specifications in [11]. For the RAW to RGB mapping we make modifications designed for ingesting RAW data formatted as a colour filter array (CFA). The RAW to RGB backbone for DIFAR is shown in Figure 13. Additionally, in Figure 12 we show how precisely the U-Net architecture is amended with MSCA-skip connections.

6.2. MSCA-skip connection ablation study

In Table 6 we present the full ablation study on the DIFAR MSCA-skip connection. For a parameter efficient model that produces images of high quality we find it is only necessary to place an MSCA-skip connection at level 1 of the U-Net (Figure 12).

Table 6: Ablation study on the MSCA-skip connection of DIFAR.

| Architecture                      | PSNR | SSIM | # Parameters |
|-----------------------------------|------|------|--------------|
| U-Net+MSCA, layer 4               | 25.09| 0.757| 2.7 M        |
| U-Net+MSCA, layer 3               | 26.00| 0.769| 2.7 M        |
| U-Net+MSCA, layer 2               | 24.90| 0.734| 1.8 M        |
| U-Net+MSCA, layer 1               | 26.56| 0.783| 1.3 M        |
| U-Net+MSCA, all layers            | 26.39| 0.793| 3.3 M        |
| U-Net                             | 25.78| 0.771| 1.4 M        |
| U-Net (large)                     | 23.37| 0.788| 5.1 M        |
| U-Net (L₁ Lab, MS-SSIM L)         | 25.75| 0.771| 1.3 M        |
| U-Net (large) (L₁ Lab, MS-SSIM L) | 25.90| 0.783| 5.1 M        |

6.3. Loss function ablation study

In Table 7 we present the full ablation study on the DIFAR loss function. We find that all loss terms are conducive to best image quality.

Table 7: Ablation study on the Samsung dataset for the various loss terms of DIFAR. \( L_{\text{lab}} \) is the loss term containing the \( L_1 \) on the Lab channels and MS-SSIM on the L channel. \( L_{\text{rgb}} \) is the \( L_1 \) on the RGB channels and the cosine distance on the RGB vectors. \( L_{\text{rgb,nc}} \) is the \( L_1 \) on the RGB channels (with no cosine distance term). \( L_{\text{hsv}} \) is the loss in HSV space. \( L_{\text{reg}} \) is the curve regularisation term. For best image quality it is important to have a loss to regularise the transformation in each colour space.

| Architecture                  | PSNR | SSIM |
|------------------------------|------|------|
| DIFAR (\( L_{\text{lab}}+L_{\text{reg}} \)) | 26.19| 0.777|
| DIFAR (\( L_{\text{rgb}}+L_{\text{reg}} \)) | 26.52| 0.785|
| DIFAR (\( L_{\text{rgb,nc}} \)) | 25.65| 0.768|
| DIFAR (\( L_{\text{hsv}}+L_{\text{reg}} \)) | 25.90| 0.777|
| DIFAR (\( L_{\text{rgb}}+L_{\text{reg}} \)) | 26.61| 0.792|
| DIFAR (\( L_{\text{hsv}}+L_{\text{reg}} \)) | 26.35| 0.781|
| DIFAR (\( L_{\text{rgb}}+L_{\text{reg}} \)) | 25.88| 0.767|
| DIFAR (\( L_{\text{rgb,nc}} \)) | 26.32| 0.768|
| DIFAR (\( L_{\text{hsv}}+L_{\text{reg}} \)) | 25.45| 0.756|
| DIFAR (\( L_{\text{hsv}}+L_{\text{reg}} \)) | 27.09| 0.793|
Figure 12: Variants of the **U-Net backbone with MSCA skip connections** at various levels. Note that the normal U-Net skip connections are removed entirely, with only one being replaced with an MSCA skip connection. This saves on parameters and leads to better performance.

Figure 13: **Pixel-level block of DIFAR.** The block is modelled as a U-Net. The raw image data is input on the left and the RGB image is output on the right. The skip connections denote our novel multi-scale contextual fusion blocks (MCF-skip).

### 6.4. Number of Knot Points for the Neural Curves

In Figure 11 we present an ablation study on how the number of knot points in the neural curve formulation affects the resulting image quality. This study is conducted on the held-out validation dataset of the medium exposure Samsung S7 dataset. We find that 16 knot points is optimal in terms of PSNR and SSIM, with quality falling for more and less knot points per curve.

### 6.5. Variations of the Retouching Block

In this section we test minor variations of the retouching block architecture. Specifically we examine the following:

- **Multiple feature extraction blocks:** In the our suggested retouching block architecture there are three feature extraction block with no parameter sharing. In Table 8 we explore a multi-task alternative with one feature extraction block (Figure 14) to determine whether having separate feature extraction blocks for each colour space is important. Our results suggest have three independent feature extraction blocks in the retouching layer is important for best image quality.

- **Ordering of colour spaces:** In the original retouching block the CIELab colour space adjustments are applied first, followed by the RGB colour space adjustments and then the HSV colour space adjustments. In Table 9 we explore all possible permutations of the Lab, HSV and RGB colour spaces to determine if the ordering is important. Our results suggest that CIELab followed by RGB and then HSV leads to the best image quality.

- **Visual examples:** We show in Figure 15 and Figure 17 examples of the output from DIFAR with all colour blocks (CIELab, RGB, HSV) versus variants of the model with just one of the available colour blocks.

### 6.6. Piecewise Linear Scaling Curve

The piecewise linear scaling curve as in Equation 7. This curve defines the mapping from a pixel property (e.g. intensity) to a scale factor that is to be applied to that pixel value.
where $M$ is the number of predicted knot points, $\hat{I}_{jl}^i$ is the $j$-th pixel value of the $l$-th channel of the $i$-th image, $k_m$ is the value of the knot point $m$. This mapping function is differentiable with respect to $\hat{I}_{jl}^i$ and the curve knot points $\{k_m\}$. The derivatives with respect to $\hat{I}_{jl}^i$ are defined almost everywhere except at the knot points, and is given in Equation 8.

$$
\frac{dS(\hat{I}_{jl}^i)}{d\hat{I}_{jl}^i} = \begin{cases} 
M(k_{m+1} - k_m) & -1 \leq m - M\hat{I}_{jl}^i \leq 1 \\
0 & \text{otherwise}
\end{cases}
$$

(8)

The derivatives for the knot points $\{k_m\}$ are defined in Equation 9.

$$
\frac{dS(\hat{I}_{jl}^i)}{dk_m} = \begin{cases} 
1 - \delta(M\hat{I}_{jl}^i - m) & m = 0 \\
\delta(M\hat{I}_{jl}^i - m + 1) - \delta(M\hat{I}_{jl}^i - m) & 0 < m < M \\
\delta(M\hat{I}_{jl}^i - m) & m = M
\end{cases}
$$

(9)

As the derivatives exist the piecewise linear scaling curve in Equation 7 is amenable for optimisation in a neural network.
Figure 15: Qualitative effect of different formulations of the retouching block. We keep only one colour space (either CIELab, RGB, HSV) in the retouching block and compare the output versus keeping all three colour space blocks. See Section 6.5 for more detail.

6.7. Colour Space Transformations

DIFAR relies on differentiable RGB→HSV, HSV→RGB, RGB→CIELab, CIELab→RGB colour space conversions to permit end-to-end learning via stochastic gradient descent and backpropagation. The colour space conversion formulae are all listed on the OpenCV website\(^4\). The colour space conversion functions consist of differentiable operations mixed with piecewise continuous functions that are also differentiable except at certain points. Non-differentiability at certain points in a domain still permits the function to be optimised by the backpropagation algorithm e.g. consider the ReLU non-linearity \(g(z) = \max\{0, z\} \) at \(z = 0\).

6.8. RGB→CIELab and CIELab→RGB

The Lab→RGB and RGB→Lab operators can be seen as a composition of elementary differentiable operations, except with the hard conditioning (thresholding) operations which are both piecewise continuous functions that are non-differentiable at certain points. The RGB→Lab operator has been used in previous work to define a loss function in Lab space that computes the \(L\) channel using the MS-SSIM metric on the \(L\) channel [38].

6.9. RGB→HSV and HSV→RGB

RGB→HSV consists of minimum and maximum operators which are both differentiable. If \(f(x_1, x_2, ..., x_n) = \max(x_1, x_2, ..., x_n)\) then the derivative is defined as in Equation 10.

\[
\frac{df}{dx_i} = \begin{cases} 
1 & \text{if } \arg\max x_i(x_i) = i \\
0 & \text{otherwise}
\end{cases} \tag{10}
\]

If \(f(x_1, x_2, ..., x_n) = \min(x_1, x_2, ..., x_n)\) then the derivative is defined as in Equation 11.

\[
\frac{df}{dx_i} = \begin{cases} 
1 & \text{if } \arg\min x_i(x_i) = i \\
0 & \text{otherwise}
\end{cases} \tag{11}
\]

In both cases the gradient is set to zero for every element except the maximum or minimum. The RGB→HSV operator also involves conditional (if-then-else) statements, conditioned on the value of \(R, G\) and \(B\). These are naturally handled by dynamic control flow that is possible in modern deep learning frameworks such as Tensorflow [45].

A HSV→RGB conversion has been little explored in the deep learning literature. The HSV→RGB conversion functions can be specified as in Equations 12-14 based on Figure 16.

\[
\mathcal{R}(\hat{I}_i) = v_j - ((v_j s_j)/60) \delta(360h_j - 60) + ((v_j s_j)/60) \delta(360h_j - 240) \tag{12}
\]

\[
\mathcal{G}(\hat{I}_i) = v_j (1 - s_j) - ((v_j s_{ij})/60) \delta(360h_j - 60) + ((v_j s_j)/60) \delta(360h_j - 180) \tag{13}
\]

\[
\mathcal{B}(\hat{I}_i) = v_j (1 - s_j) - ((v^j s^j_i)/60) \delta(360h_j - 120) + (v_j s_j)/60) \delta(360h_j - 300) \tag{14}
\]

where we define \(\delta(\cdot)\) as in Equation 15:

\[
\delta(x) = \begin{cases} 
0 & x < 0 \\
1 & 0 \leq x \leq 60 \\
60 & x > 60
\end{cases} \tag{15}
\]

with \(h_j, s_j, v_j \in [0, 1]\), the hue, saturation and value of pixel \(j\) of image \(I_i\). Equations 12-14 are piecewise linear curves defined by linear segments and knot points and they have defined gradients between the knot points. It is also possible to formulate a smooth approximation to the HSV→RGB transformation using suitable scaled and offset cosine waves (Equations 16-18).
Figure 16: Piecewise linear function used to convert HSV values to RGB colour space. This figure provides a graphical representation of RGB coordinates given values for HSV. The equation $V(1-S)=V-VS$ shows the origin of the marked vertical axis values. Figure source: en:user:Goffrie-en:Image:HSV_RGB_Comparison.svg, CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=1116423.

$$R(\hat{I}_j) = v_j (1 + s_j (\cos(360h_j) - 1))$$  \hspace{1cm} (16)  
$$G(\hat{I}_j) = v_j (1 + s_j (\cos(360h_j - 120) - 1))$$  \hspace{1cm} (17)  
$$B(\hat{I}_j) = v_j (1 + s_j (\cos(360h_j + 120) - 1))$$  \hspace{1cm} (18)

with $h_j$, $s_j$, $v_j \in [0, 1]$. These equations, being smooth approximations, lead to errors (i.e. small differences to the real RGB values) in the conversion from HSV to RGB. In future work it would be interesting to explore the effect of this smooth approximation versus the full piecewise linear formulation on the convergence of DIFAR and the quality of the resulting images.

### 6.10. Limitations of DIFAR

The retouching block of DIFAR uses multiple conversions for the CIELab, RGB and HSV colour spaces. Each of these conversions may lead to a loss of image fidelity. We mitigate this issue by a long skip connection between the input and output of the retouching layer, which we found is important for model training and image quality.

### 6.11. Additional Visual Examples

We provide additional examples of output images from DIFAR and the baseline models on the Samsung and Adobe datasets in Figures 18-19. The following visual examples are provided:

- Figure 17: Comparing a retouching network with just RGB, versus a retouching network with all three colour spaces (CIELab, RGB, HSV).
- Figure 18: Examples images produced by Deep-ISP [38] and DIFAR on the Samsung S7 Medium Exposure dataset.
- Figure 19: Examples images produced by Deep-ISP [38] and DIFAR on the Samsung S7 Short Exposure dataset.
- Figure 20: Examples images produced by Unet (large) [36] and DIFAR on the Samsung S7 Medium Exposure dataset.
- Figure 21: Examples images produced by Unet (large) [36] and DIFAR on the Samsung S7 Short Exposure dataset.
- Figures 22-23: Examples images produced by Deep-UPE [42] and DIFAR on the Adobe-UPE dataset.
- Figures 24-25: Examples images produced by DPE [11] and DIFAR on the Adobe-DPE dataset.

### References

[52] Y. Gao, H. Hu, and B. Li, and Q. Guo. Naturalness Preserved Non-Uniform Illumination Estimation for Image Enhancement Based on Retinex. *IEEE Transactions on Multimedia*, 2017. 23, 24

[53] M. Aubry, S. Paris, S. Hasinoff, J. Kautz, and F. Durand. Fast Local Laplacian Filters: Theory and Applications. *ACM Trans. Graph.*, 2014. 23, 24

[54] S. Wang, W. Cho, J. Jang, M. Abidi, and J. Paik. Contrast-dependent saturation adjustment for outdoor image enhancement. *Journal of the Optical Society of America A*, 2017. 23, 24
Figure 17: Comparing a retouching network with just RGB, versus a retouching network with all three colour spaces (CIELab, RGB, HSV). Examples images from the the Samsung S7 Medium Exposure dataset.
Figure 18: Examples images produced by DeepISP [38] and DIFAR on the Samsung S7 Medium Exposure dataset.
Figure 19: Examples images produced by DeepISP [38] and DIFAR on the Samsung S7 Short Exposure dataset.
Figure 20: Examples images produced by Unet (large) and DIFAR on the Samsung S7 Medium Exposure dataset.
Figure 21: Examples images produced by **Unet (large)** and **DIFAR** on the Samsung S7 **Short Exposure** dataset.
Figure 22: Examples images produced by DeepUPE [42] and DIFAR on the Adobe-UPE dataset.
Figure 23: Examples images produced by DeepUPE [42] and DIFAR on the Adobe-UPE dataset.
Figure 24: Examples images produced by DPE [11], CLHE [54], DPED (iphone7) [23], NPEA [52], FLLF [53] and DIFAR on the Adobe-DPE dataset.
Figure 25: Examples images produced by DPE [11], CLHE [54], DPED (iphone7) [23], NPEA [52], FLLF [53] and DIFAR on the Adobe-DPE dataset.