

# DeepLPF: Deep Local Parametric Filters for Image Enhancement

Sean Moran<sup>1</sup>, Pierre Marza<sup>\*,1,2</sup>, Steven McDonagh<sup>1</sup>, Sarah Parisot<sup>1,3</sup>, Gregory Slabaugh<sup>1</sup>

sean.j.moran@gmail.com, pierre.marza@gmail.com,

{steven.mcdonagh, sarah.parisot, gregory.slabaugh}@huawei.com,

<sup>1</sup>Huawei Noah's Ark Lab

<sup>2</sup>INSA Lyon

<sup>3</sup>Mila Montréal

## Abstract

Digital artists often improve the aesthetic quality of digital photographs through manual retouching. Beyond global adjustments, professional image editing programs provide local adjustment tools operating on specific parts of an image. Options include parametric (graduated, radial filters) and unconstrained brush tools. These highly expressive tools enable a diverse set of local image enhancements. However, their use can be time consuming, and requires artistic capability. State-of-the-art automated image enhancement approaches typically focus on learning pixel-level or global enhancements. The former can be noisy and lack interpretability, while the latter can fail to capture fine-grained adjustments. In this paper, we introduce a novel approach to automatically enhance images using learned spatially local filters of three different types (Elliptical Filter, Graduated Filter, Polynomial Filter). We introduce a deep neural network, dubbed Deep Local Parametric Filters (DeepLPF), which regresses the parameters of these spatially localized filters that are then automatically applied to enhance the image. DeepLPF provides a natural form of model regularization and enables interpretable, intuitive adjustments that lead to visually pleasing results. We report on multiple benchmarks and show that DeepLPF produces state-of-the-art performance on two variants of the MIT-Adobe 5k [3] dataset, often using a fraction of the parameters required for competing methods.

## 1. Introduction

Digital photography has progressed dramatically in recent years due to sustained improvements in camera sensors and image signal processing pipelines. Yet despite this progress, captured photographs may still lack quality due to varying factors including scene condition, poor illumination, or photographer skill. Human image retouchers often

\*Work done during an internship at Huawei Noah's Ark Lab.

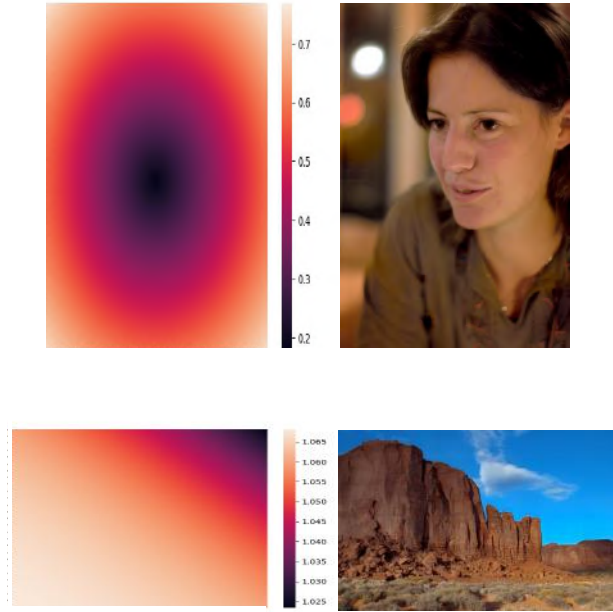


Figure 1: DeepLPF for parametric local image enhancement. **Left:** Examples of estimated filters. **Right:** The produced output images. **Top:** Adjustment of the image red channel with a single Elliptical Filter. **Bottom:** Adjustment of the image red channel with a single Graduated Filter.

improve the aesthetic quality of digital photographs through manual adjustments. Professional-grade software (e.g. Photoshop, Lightroom) allows application of a variety of modifications through both interactive and semi-automated tools.

In addition to elementary global adjustments such as contrast enhancement and brightening, advanced editing functionality is also available through local image adjustments, such as the examples shown in Fig 1. However, manual enhancement remains challenging for non-experts who may lack appropriate skills, time, or aesthetic judgment to improve their images effectively.

These observations motivate the development of fully

automatic photo enhancement tools that can replace non-expert user work or provide an improved manual-editing starting point for professional artists. Photographers often retouch images using a combination of different *local filters* that only affect limited spatial regions of the image. For example, a photographer might want to adjust the darkness of the sky using a graduated filter while increasing the brightness of a face using an appropriately sized elliptical filter, and retouching small fine image detail using a brush tool.

Inspired by this manual workflow, we propose a novel approach that learns parametric filters for local image enhancement. We draw influence from digital photo editing software, but model and emulate local editing tools using a deep neural network. Given (input, enhanced) pairs as training examples, we reproduce local, mid-level adjustments through learned graduated and elliptical filters and a learned brush tool. By constraining our model to learn how to utilize tools that are similar to those found in a digital artists' toolbox, we provide a natural form of model regularization and enable interpretable, intuitive adjustments that lead to visually pleasing results. Extensive experiments on multiple public datasets show that we outperform state-of-the-art [8, 25, 4] results with a fraction of the neural network weight capacity.

Our **Contributions** can be summarised as follows:

- **Local Parametric Filters:** We propose a method for automatic estimation of parametric filters for local image enhancement. We instantiate our idea using Elliptical, Graduated, Polynomial filters. Our formulation provides intuitively interpretable and intrinsically regularised filters that ensure weight efficiency (capacity frugality) and mitigate overfitting.
- **Multiple Filter Fusion Block:** We present a principled strategy for the fusion of multiple learned parametric image filters. Our novel plug-and-play neural block is capable of fusing multiple independent parameter filter outputs and provides a flexible layer that can be integrated with common network backbones for image quality enhancement.
- **State-Of-The-Art Image Enhancement Quality:** DeepLPF provides state-of-the-art image quality enhancement on two challenging benchmarks.

## 2. Related work

Digital photo enhancement has a rich history in both the image processing and computer vision communities. Early automated enhancement focused primarily on image contrast [20, 24, 29], while recent work has employed data-driven methods to learn image adjustments for improving contrast, colour, brightness and saturation [11, 13, 9, 8, 10, 19]. The related image enhancement work can be broadly

divided into methods that operate globally or locally on an image, or propose models that operate over both scales.

**Global image enhancement:** Bychkovsky *et al.* [3] collected the popular MIT-Adobe-5K dataset<sup>1</sup> that consists of 5,000 photographs and their retouching by five different artists. The authors propose a regression based approach to learn the artists' photographic adjustments from image pairs. To automate colour enhancement, [27] propose a learning-to-rank approach over ten popular global colour controls. In [7] an FCN is used to learn approximations to various global image processing operators such as photographic style, non-local dehazing and pencil drawing. Photo post-processing is performed by a white-box framework in [10] where global retouching curves are predicted in RGB space. A Reinforcement Learning (RL) approach enables the image adjustment process in [10] and Deep RL is used to define an ordering for enhancement adjustments in [19], enabling application of global image modifications (eg. contrast, saturation).

**Local image enhancement:** Aubry *et al.* [1] propose fast local Laplacian filtering for enhancing image detail. Hwang *et al.* [12] propose a method for local image enhancement that searches for the best local match for an image in a training database of (input, enhanced) image pairs. Semantic maps are constructed in [28] to achieve semantic-aware enhancement and learn local adjustments. Underexposed photo enhancement [25] (DeepUPE) learns a scaling luminance map using an encoder-decoder setup however no global adjustment is performed. The learned mapping has high complexity and crucially depends on the regularization strategy employed. Chen *et al.* [7] propose a method for fast learning of image operators using a multi-scale context aggregation network. Enhancement via image-to-image translation using both cycle consistency [32] and unsupervised learning [18] have also been proposed in recent years.

**Global and local image enhancement:** Chen *et al.* [8] develop Deep Photo Enhancer (DPE), a deep model for enhancement based on two-way generative adversarial networks (GANs). As well as local pixel-level adjustments, DPE introduces a global feature extraction layer to capture global scene context. Ignatov *et al.* [14] design a weakly-supervised image-to-image GAN-based network, removing the need for pixel-wise alignment of image pairs. Chen *et al.* [4] propose a low-light enhancement model that operates directly on raw sensor data and propose a fully-convolutional approach to learn short-exposure, long-exposure mappings using their low-light dataset. Recent work makes use of multitask learning [16] for real-time image processing with various image operators. Bilateral guided joint upsampling enables an encoder/decoder architecture for local as well as global image processing. HDR-Net [9] learns global and local image adjustments by lever-

<sup>1</sup><https://data.csail.mit.edu/graphics/fivek/>

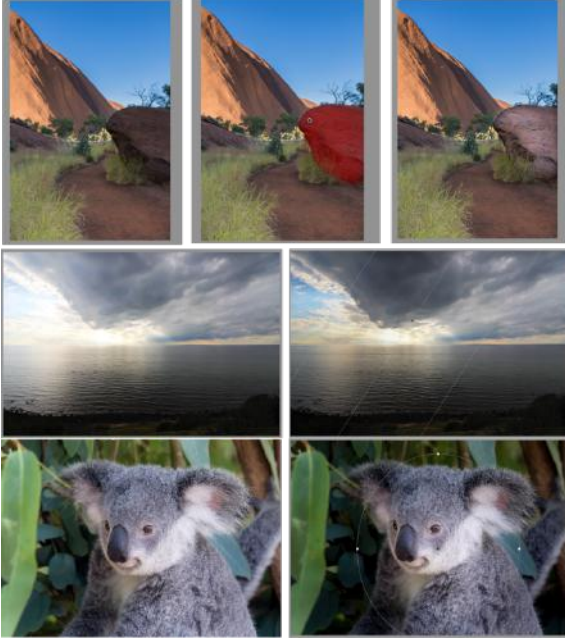


Figure 2: Examples of local filter usage in Lightroom for the brush tool (top row), graduated (middle row) and radial (bottom row) filters. Images are shown before (left) and after filter based enhancement by a human artist (right).

aging a two stream convolutional architecture. The local stream extracts local features that are used to predict the coefficients of local affine transforms, while the global stream extracts global features that permit an understanding of scene category, average intensity *etc.* In [13] DSLR-quality photos are produced for mobile devices using residual networks to improve both colour and sharpness.

In contrast with prior work, we propose to frame the enhancement problem by learning parametric filters, operating locally on an image. Learnable, parametrized filters align well with the intuitive, well-understood human artistic tools often employed for enhancement and this naturally produces appealing results that possess a degree of familiarity to human observers. Additionally, the use of our filter parameterization strategy serves to both constrain model capacity and regularize the learning process, mitigating overfitting and resulting in moderate model capacity cost.

### 3. Deep Local Parametric Filters (DeepLPF)

DeepLPF defines a novel approach for local image enhancement, introducing a deep fusion architecture capable of combining the output from learned, spatially local parametric image filters that are designed to emulate the combined application of analogous manual filters. In this work, we instantiate three of the most popular local image filters

(Elliptical, Graduated, Polynomial). Figure 2 illustrates usage, and resulting effects of, comparable manual Lightroom filters. In Section 3.1 we present our global DeepLPF architecture, designed to learn and apply sets of different parametric filters. Section 3.2 then provides detail on the design of the three considered local parametric filters, and describes the parameter prediction block used to estimate filter parameters. Finally, Sections 3.3 and 3.4 explain how multiple filters are fused together and provide detail on our training loss function, respectively.

#### 3.1. DeepLPF Architecture

The architecture of DeepLPF is shown in Figure 3. Given a low quality RGB input image  $I$  and its corresponding high quality enhanced target image  $Y$ , DeepLPF is trained to learn a transformation  $f_\theta$  such that  $\hat{Y} = f_\theta(I)$  is close to  $Y$  as determined by an objective function based on image quality. Our model combines a single-stream network architecture for fine-grained enhancement, followed by a two-stream architecture for higher-level, local enhancement. We first employ a standard CNN backbone (*e.g.* ResNet, UNet) to estimate a  $H \times W \times C$  dimensional feature map. The first three channels of the feature map represent the image to be adjusted, while the remaining  $C' = C - 3$  channels represent additional features that feed into three filter parameter prediction blocks. The first *single stream path* estimates the parameters of a polynomial filter that is subsequently applied to the pixels of the backbone enhanced image  $\hat{Y}_1$  input image. This block emulates a brush tool which adjusts images at the pixel level with a smoothness constraint imposed by the brush’s shape. The image enhanced by the polynomial filter,  $\hat{Y}_2$ , is concatenated with the  $C'$  backbone features, and serves as input to the *two stream path*, which learns and applies more constrained, local enhancements in the form of elliptical and graduated filters. The adjustment maps of the elliptical and graduated filters are estimated using two parallel regression blocks. Elliptical and graduated maps are fused using simple addition, although more involved schemes could be employed *e.g.* weighted combination. This fusion step results in a scaling map  $\hat{S}$  that is element-wise multiplied to  $\hat{Y}_2$  to give image  $\hat{Y}_3$ , effectively applying the elliptical and graduated adjustments to the image after polynomial enhancement. The image  $\hat{Y}_1$ , enhanced by the backbone network, is finally added through a long residual connection to  $\hat{Y}_3$  producing the final output image  $\hat{Y}$ .

#### 3.2. Local Parametric Filters

We provide detail on three examples of local parametric filter: the Graduated Filter (3.2.2), Elliptical Filter (3.2.3) and Polynomial Filter (3.2.4). Filters permit different types of local image adjustment, governed by their parametric form, and parameter values specify the exact image effect

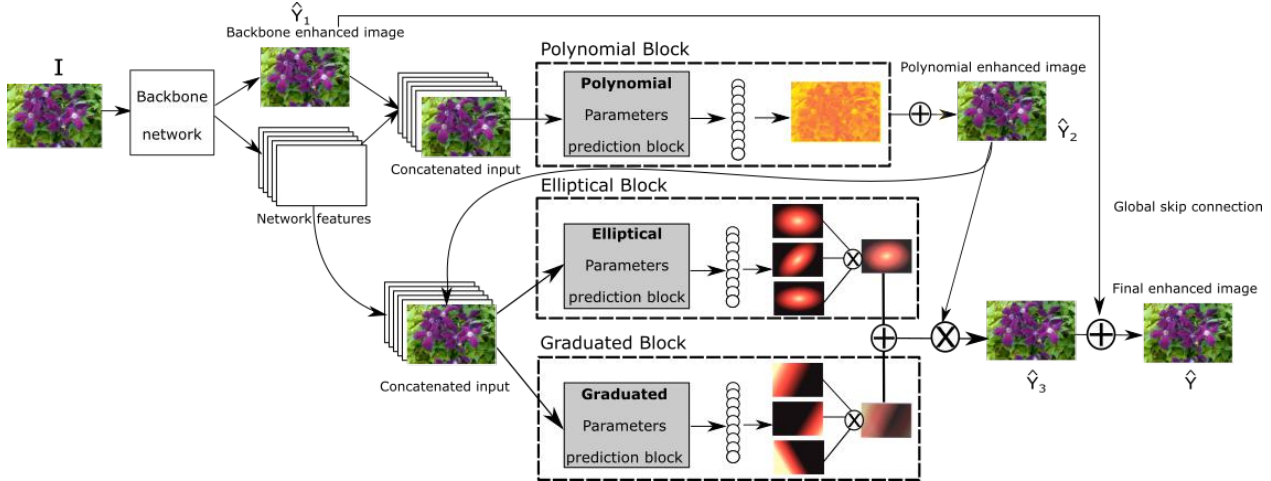


Figure 3: Architecture diagram illustrating our approach to combine the different filter types (polynomial, elliptical, graduated) in a single end-to-end trainable neural network. The architecture combines a *single stream path* for initial enhancement with the polynomial filter and a *two stream path* for further refinement with the graduated and elliptical filters.

in each case. Filter parameter values are image specific and are predicted using supervised CNN regression (3.2.1).

### 3.2.1 Filter Parameter Prediction Network

Our parameter prediction block is a lightweight CNN that accepts a feature set from a backbone network and regresses filter parameters individually. The network block alternates between a series of convolutional and max pooling layers that gradually downsample the feature map resolution. Following these layers there is a global average pooling layer and a fully connected layer which is responsible for predicting the filter parameters. The global average pooling layer ensures that the network is agnostic to the resolution of the input feature set. Activation functions are Leaky ReLUs and dropout (50%, both train and test) is applied to the fully connected layer. Further architectural details are provided in our supplementary material.

For the three example filters considered; the only difference between respective network regressors is the number of output nodes in the final fully connected layer. This corresponds to the number of parameters that define the respective filter (Table 1). In comparison to contemporary local pixel level enhancement methods [8], the number of parameters required is greatly reduced. Our network output size can be altered to estimate the parameters of multiple instances of the same filter type. Our methodology for predicting the parameters of an image transformation is aligned with previous work [9, 5, 23] that shows that learning a parameterised transformation is often more effective and simpler than directly predicting an enhanced image directly. Importantly, the regression network is agnostic to the

Table 1: Parameters used in localized filters

Filter	# Parameters	Parameters
Graduated	$G=8$	$s_g^R, s_g^G, s_g^B, m, c, o_1, o_2, g_{inv}$
Elliptical	$E=8$	$s_e^R, s_e^G, s_e^B, h, k, \theta, a, b$
Cubic-10	$P=30$	$\{A \cdots J\}$ per colour channel
Cubic-20	$P=60$	$\{A \cdots T\}$ per colour channel

specific implementation of backbone model allowing image filters to be used to enhance the output of any image translation network.

### 3.2.2 Graduated Filter

Graduated filters are commonly used in photo editing software to adjust images with high contrast planar regions such as an overexposed sky. Our graduated image filter, illustrated in Figure 4a, is parametrised by three parallel lines. The central line defines the filter location and orientation, taking the form  $y=mx+c$  with slope  $m$  and intercept  $c$ , providing a linear parameterisation in standard fashion. Offsets  $o_1$  and  $o_2$  provide two additional parameters such that each (channel-wise) adjustment map is composed of four distinct regions that form a heatmap  $s(x, y)$ . In the the 100% area all pixels are multiplied by scaling parameter  $s_g$ . Then, in the 100-50% area, the applied scaling factor linearly decreases from  $s_g$  to  $\frac{s_g}{2}$ . Inside the 50-0% area, the scaling value is further decreased linearly until reaching the 0% area where pixels are not adjusted. Mathematically, the graduated filter is described in Equations 1-3:



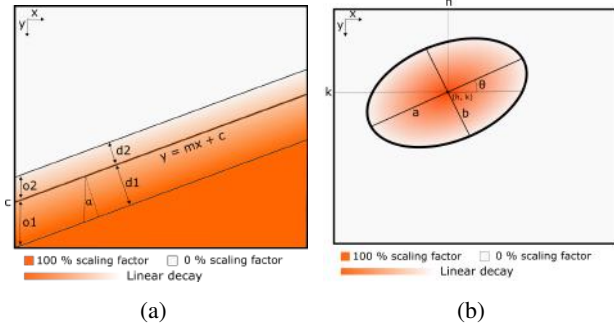


Figure 4: Parametrisation and heatmap of the graduated (a) and elliptical (b) filters. See main text for further detail.

$$a(x, y) = s_g \min \left\{ \frac{1}{2} \left( 1 + \frac{\ell(x, y)}{d_2} \right), 1 \right\} \quad (1)$$

$$b(x, y) = s_g \max \left\{ \frac{1}{2} \left( 1 + \frac{\ell(x, y)}{d_1} \right), 0 \right\} \quad (2)$$

$$s(x, y) = \begin{cases} \hat{g}_{inv} a(x, y) + (1 - \hat{g}_{inv}) b(x, y) & , l(x, y) \geq 0 \\ (1 - \hat{g}_{inv}) a(x, y) + \hat{g}_{inv} b(x, y) & , l(x, y) < 0 \end{cases} \quad (3)$$

where  $\ell(x, y) = y - (mx + c)$  is a function of the location of a point  $(x, y)$  relative to the central line,  $d_1 = o_1 \cos \alpha$ ,  $d_2 = o_2 \cos \alpha$ ,  $\alpha = \tan^{-1}(m)$ , and  $\hat{g}_{inv}$  is a binary indicator variable. Parameter  $\hat{g}_{inv}$  permits *inversion* with respect to the top and bottom lines. Inversion determines the location of the 100% scaling area relative to the central line. To enable learnable inversion, we predict a binary indicator parameter  $\hat{g}_{inv} = \frac{1}{2}(\text{sgn}(g_{inv}) + 1)$ , where  $\text{sgn}$  denotes the sign function,  $g_{inv}$  is the real-valued predicted parameter and  $\hat{g}_{inv}$  is the binarised  $\hat{g}_{inv} \in \{0, 1\}$  version. The gradient of this sign function is everywhere zero and undefined at zero, therefore we use the straight-through estimator [2] on the backward pass to learn this binary variable.

### 3.2.3 Elliptical Filter

A further filter we utilise defines an ellipse parametrised by the center  $(h, k)$ , semi-major axis  $(a)$ , semi-minor axis  $(b)$  and rotation angle  $(\theta)$ . The learned scaling factor  $s_e$  is maximal at the center of the ellipse (100% point) and decreases linearly until reaching the boundary. Pixels are not adjusted outside the ellipse (0% area).

Mathematically, a channel-wise heatmap, defined by the elliptical filter, is denoted as:

$$s(x, y) = s_e \min \left( 0, 1 - \frac{[(x - h) \cos \theta + (y - k) \sin \theta]^2}{a^2} + \frac{[(x - h) \sin \theta - (y - k) \cos \theta]^2}{b^2} \right). \quad (4)$$

An example elliptical filter parametrisation and heatmap is illustrated in Fig. 4b. Elliptical filters are often used to enhance, accentuate objects or specific regions of interest within photographs *e.g.* human faces.

### 3.2.4 Polynomial Filter

Our third considered filter type constitutes a polynomial filter capable of providing fine-grained, regularised adjustments over the entire image. The polynomial filter emulates a brush tool, providing a broad class of geometric shapes while incorporating spatial smoothness. We consider order- $p$  polynomial filters of the forms  $i \cdot (x + y + \gamma)^p$  and  $(x + y + i + \gamma)^p$ , where  $i$  is the image channel intensity at pixel location  $(x, y)$ , and  $\gamma$  is an independent scalar. We empirically find a cubic polynomial ( $p = 3$ ) to offer both expressive image adjustments yet only a limited set of parameters. We explore two variants of the cubic filter, *cubic-10* and *cubic-20*.

Our smaller cubic filter (*cubic-10*) constitutes a set of 10 parameters to predict;  $\{A, B, C, \dots, J\}$ , defining a cubic function  $f$  that maps intensity  $i$  to a new adjusted intensity  $i'$ :

$$\begin{aligned} i'(x, y) &= f(x, y, i) \\ &= i * (Ax^3 + Bx^2y + Cx^2i + Dx^2 + Exy^2 \\ &\quad + Dxy^2 + Exy + Fx + Gy^3 + Hy^2 \\ &\quad + Iy + J) \end{aligned} \quad (5)$$

Considering twice as many learnable parameters, the *cubic-20* filter explores higher order intensity terms, and consists of a set of 20 parameters  $\{A, B, C, \dots, T\}$ :

$$\begin{aligned} i'(x, y) &= f(x, y, i) \\ &= Ax^3 + Bx^2y + Cx^2i + Dx^2 + Exy^2 \\ &\quad + Fxyi + Gxy + Hxi^2 + Ixi + Jx \\ &\quad + Ky^3 + Ly^2i + My^2 + Nyi^2 + Oyi \\ &\quad + Py + Qi^3 + Ri^2 + Si + T \end{aligned} \quad (6)$$

Our cubic filters consider *both* spatial and intensity information while constraining the complexity of the learnt mapping to a regularized form. This permits the learning of precise, pixel level enhancement while ensuring that transformations are locally smooth. We estimate an independent

cubic function for each colour channel yielding a total of 30 parameters for the *cubic-10* and 60 parameters for the *cubic-20* filter respectively.

### 3.3. Fusing Multiple Filters of the Same Type

Our graduated and elliptical prediction blocks can each output parameter values for  $n$  instances of their respective filter type. Further exploration towards choosing  $n$  appropriately is detailed in Section 4.2. In the case  $n > 1$ , multiple instances are combined into an adjustment map through element-wise multiplication,

$$s_g(x, y) = \prod_{i=1}^n s_{gi}(x, y), \quad s_e(x, y) = \prod_{i=1}^n s_{ei}(x, y) \quad (7)$$

where  $s_{gi}(x, y)$  and  $s_{ei}(x, y)$  are the adjustment map of the  $i^{th}$  instance of the corresponding graduated and elliptical filters respectively. On the contrary, since a single per-channel cubic filter allows intrinsically high flexibility in terms of expressive power, we opt not to fuse multiple cubic filters together.

### 3.4. DeepLPF Loss Function

The DeepLPF training loss leverages the CIELab colour space to compute the  $L_1$  loss on the Lab channels and the MS-SSIM loss on the L channel (Equation 8). By splitting chrominance and luminance information into separate loss terms, our model is able to account for separate focus on both local (MS-SSIM) and global ( $L_1$ ) image enhancements during training [22, 31]. Given a set of  $N$  image pairs  $\{(Y_i, \hat{Y}_i)\}_{i=1}^N$ , where  $Y_i$  is the reference image and  $\hat{Y}_i$  is the predicted image, we define the DeepLPF training loss function as:

$$\mathcal{L} = \sum_{i=1}^N \{ \omega_{lab} \|Lab(\hat{Y}_i) - Lab(Y_i)\|_1 + \omega_{ms-ssim} MS-SSIM(L(\hat{Y}_i), L(Y_i)) \} \quad (8)$$

where  $Lab(\cdot)$  is a function that returns the CIELab Lab channels corresponding to the RGB channels of the input image and  $L(\cdot)$  returns the L channel of the image in CIELab colour space. MS-SSIM is the multi-scale structural similarity [26], and  $\omega_{lab}, \omega_{ms-ssim}$  are hyperparameters weighting the relative influence of the terms in the loss function.

## 4. Experiments

### 4.1. Experimental Setup

**Datasets:** We evaluate DeepLPF on three challenging benchmarks, derived from two public datasets. Firstly (i) **MIT-Adobe-5K-DPE** [8]: 5000 images captured using various DSLR cameras. Each captured image is subsequently

(independently) retouched by five human artists. In order to make our results reproducible and directly comparable to the state-of-the-art, we consider only the (supervised) subset used by DeepPhotoEnhancer (DPE) [8] and additionally follow their dataset pre-processing procedure. The image retouching of Artist C is used to define image enhancement ground truth. The data subset is split into 2250, 500 training and testing image pairs respectively. We randomly sample 500 images from the training set, providing an additional validation set for hyperparameter optimisation. The images are resized to have a long-edge of 500 pixels. (ii) **MIT-Adobe-5K-UPE** [25]: our second benchmark consists of the same image content as **MIT-Adobe-5K-DPE** however image pre-processing here differs and instead follows the protocol of DeepUPE [25]. We therefore refrain from image resizing and dataset samples vary in pixel resolution; 6-25M. We additionally follow the train / test split provided by [25] and our second benchmark therefore consists of 4500 training image pairs, from which we randomly sample 500 to form a validation set. Testing images (500) are identical to those samples selected by DeepUPE and ground truth again consists of the Artist C manually retouched images. (iii) **See-in-the-dark (SID)** [4]: the dataset consists of 5094 image pairs, captured by a Fuji camera. For each pair the input is a short-exposure image in raw format and the ground truth is a long-exposure RGB image. Images are 24M pixel in size and content consists of both indoor and outdoor environments capturing diverse scenes and common objects of varying size.

**Evaluation Metrics:** We evaluate quantitatively using PSNR, SSIM and the perceptual LPIPS metric [30].

**Implementation details:** Our experiments all employ a U-Net backbone [21]. The base U-Net architecture, used for MIT-Adobe-5K-DPE experimentation, is detailed in the supplementary material. The U-Net architectures used for the other benchmarks are similar yet have a reduced number of convolutional filters (MIT-Adobe-5K-UPE) or include pixel shuffling functionality [17] to account for RAW input (SID). All experiments use the Adam Optimizer with a learning rate of  $10^{-4}$ . Our architecture makes use of three graduated (elliptical) filters per channel and we search for loss function (Eq. 8) hyperparameters empirically resulting in:  $\omega_{lab} = \{1, 1, 1\}$  and  $\omega_{ms-ssim} = \{10^{-3}, 10, 10^{-3}\}$  for all MIT-Adobe-5K-DPE, MIT-Adobe-5K-UPE and SID experiments, respectively.

### 4.2. Experimental Results

**Ablation study:** We firstly conduct experiments to understand the different contributions and credit assignment for our method filter components, using the MIT-Adobe-5K-DPE data. Table 2 (upper) shows ablative results for our considered image filter components. For each configuration, we report PSNR metrics and verify experimentally the



Figure 5: Model output images and examples of Elliptical, Graduated and Polynomial (cubic-10) image filters learnt by one instance of DeepLPF. Lighter filter heat map colours correspond to larger image adjustment values. Cubic-10 filter is shown without the intensity ( $i$ ) term. Best viewed in colour.

Table 2: Model filter type ablation study (upper) and comparisons with state-of-the-art methods (lower) using the **MIT-Adobe-5K-DPE** benchmark. PSNR and SSIM results, reported by competing works, are replicated from [8].

Architecture	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$	# Weights
U-Net	21.57	0.843	0.601	1.3 M
U-Net+Elliptical	22.56	0.879	—	1.5 M
U-Net+Graduated	22.64	0.888	—	1.5 M
U-Net+Elliptical+Graduated	22.88	0.886	—	1.6 M
U-Net+Cubic-10	22.69	0.871	—	1.5 M
U-Net+Cubic-20	23.44	0.886	—	1.5 M
U-Net+Cubic-20+Elliptical+Graduated	<b>23.93</b>	<b>0.903</b>	<b>0.582</b>	<b>1.8 M</b>
DPED [13]	21.76	0.871	—	—
8RESBLK [32, 18]	23.42	0.875	—	—
FCN [7]	20.66	0.849	—	—
CRN [6]	22.38	0.877	—	—
DPE [8]	23.80	0.900	0.587	3.3 M

importance of integrating each component and their superior combined performance.

Individual filters are shown to bring boosts in performance *c.f.* the U-Net backbone alone. The polynomial (cubic) filter has a greater impact than elliptical and graduated filters. We hypothesise that this can be attributed to the spatially large yet non-linear effects enabled by this filter type. Combining all filter blocks incorporates the benefits of each and demonstrates strongest performance. We highlight that only  $\sim \frac{1}{4}$  of model parameters (*i.e.* around 452k) in the full architecture are attributed to filter blocks. The remaining capacity is dedicated to the U-Net backbone, illustrating that the majority of performance gain is a result of designing models capable of frugally emulating manual image enhancement tools.

We further investigate the influence of graduated and elliptical filter *quantity* on model performance in Figure 6. We find a general upward trend in PSNR, SSIM metrics as the number of filters per channel are increased. This upward trend can be attributed to the increased modelling capacity brought about by the additional filters. We select 3 filters per channel in our experiments which provides a tradeoff between image quality and parameter count.

**Quantitative Comparison:** We compare DeepLPF image

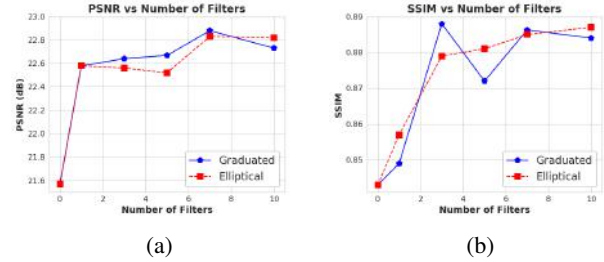


Figure 6: Experimental study evaluating the effect of the number of graduated and elliptical filters used (**MIT-Adobe-5K-DPE** dataset).

quality on MIT-Adobe-5K-DPE with contemporary methods (Table 2, lower). Our full architecture is able to outperform recent methods; 8RESBLK [32, 18] and the supervised state-of-the-art (Deep Photo Enhancer (DPE) [8]) on each of the three considered metrics, whilst our method parameter count approaches half that of their model capacity. Our implicitly regularised model formulation of filters allows similar quality enhancement capabilities yet is represented in a simpler form.

Similarly DeepLPF outperforms DeepUPE [25] on our second benchmark; MIT-Adobe-5K-UPE (*i.e.* using their dataset splits and pre-processing protocols) when considering PSNR and LPIPS and provides competitive SSIM performance, and improves upon on all other compared works across the PSNR, SSIM metrics (see Table 3).

Finally we examine performance on the Fuji portion of the challenging SID dataset. Results are presented in Table 4 where it can be observed that DeepLPF is able to improve upon the U-Net method presented in [4] across all three considered metrics. Our method again proves more frugal, with model capacity lower by a factor of nearly four.

**Qualitative Comparison:** Sample visual results comparing DeepLPF (trained independently using each of the three datasets) are shown in comparison to DPE [8], DeepUPE [25] and SID (U-Net) [4] models in Figure 7 rows, respectively. The park scene (first row) can be seen to visually improve with regard to reproduction of colour faithful-



Figure 7: Qualitative comparisons between DeepLPF and various state-of-the-art methods. See text for further details.

Table 3: Quantitative comparison with state-of-the art methods on the **MIT-Adobe-5K-UPE** benchmark. PSNR and SSIM reported by competing works, are replicated from [8].

Architecture	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$	# Weights
DeepLPF	<b>24.48</b>	0.887	<b>0.103</b>	<b>800K</b>
U-Net [21]	22.24	0.850	—	1.3 M
HDRNet [9]	21.96	0.866	—	—
DPE [8]	22.15	0.850	—	3.3 M
White-Box [10]	18.57	0.701	—	—
Distort-and-Recover [19]	20.97	0.841	—	—
DeepUPE [25]	23.04	<b>0.893</b>	0.158	1.0 M

Table 4: Quantitative performance comparisons for the image enhancement task defined by the RAW to RGB image pairs in the **SID** dataset (Fuji camera) [4].

Architecture	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$	# Weights
DeepLPF	<b>26.82</b>	<b>0.702</b>	<b>0.564</b>	<b>2.0 M</b>
U-Net [4]	26.61	0.680	0.586	7.8 M

ness in comparison to the DPE result. In the second row DeepUPE has overexposed the scene, whereas DeepLPF maintains both accurate exposure and colour content. The final row compares results on the challenging low-light dataset. The SID model output suffers from a purple colour

cast whereas the DeepLPF output provides improved colour constancy in comparison to the ground truth. Finally, Figure 5 provides a selection of parametric filters, represented by heatmaps, learned by our model. We provide additional qualitative results in the supplementary material.

## 5. Conclusion

In this paper, we have explored automated parameterisation of filters for spatially localised image enhancement. Inspired by professional image editing tools and software, our method estimates a sequence of image edits using graduated, elliptical and polynomial filters whose parameters can be regressed directly from convolutional features provided by a backbone network *e.g.* U-Net. Our localised filters produce interpretable image adjustments with visually pleasing results and filters constitute plugable and reusable network blocks capable of improving image visual quality.

In future work we can further explore automatic estimation of the optimal *sequence* of filter application; *e.g.* the Gumbel softmax trick [15] may prove useful to select operations from a potentially large bank of image editing tools. We think that combining our presented local filters with additional local or global filter types and segmentation masks, refining enhancements to semantically related pixels, also provide interesting future directions towards interpretable and frugal automated image enhancement.



## References

- [1] Mathieu Aubry, Sylvain Paris, Samuel W. Hasinoff, Jan Kautz, and Frédo Durand. Fast local laplacian filters: Theory and applications. *ACM Trans. Graph.*, 33(5):167:1–167:14, Sept. 2014. [2](#)
- [2] Yoshua Bengio. Estimating or propagating gradients through stochastic neurons. *CoRR*, abs/1305.2982, 2013. [5](#)
- [3] Vladimir Bychkovsky, Sylvain Paris, Eric Chan, and Frédo Durand. Learning photographic global tonal adjustment with a database of input/output image pairs. In *CVPR*, pages 97–104. IEEE, 2011. [1](#), [2](#)
- [4] Chen Chen, Qifeng Chen, Jia Xu, and Vladlen Koltun. Learning to see in the dark. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3291–3300, 2018. [2](#), [6](#), [7](#), [8](#)
- [5] Jiawen Chen, Andrew Adams, Neal Wadhwa, and Samuel W. Hasinoff. Bilateral guided upsampling. *ACM Trans. Graph.*, 35(6):203:1–203:8, Nov. 2016. [4](#)
- [6] Qifeng Chen and Vladlen Koltun. Photographic image synthesis with cascaded refinement networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1511–1520, 2017. [7](#)
- [7] Qifeng Chen, Jia Xu, and Vladlen Koltun. Fast image processing with fully-convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2497–2506, 2017. [2](#), [7](#)
- [8] Yu-Sheng Chen, Yu-Ching Wang, Man-Hsin Kao, and Yung-Yu Chuang. Deep photo enhancer: Unpaired learning for image enhancement from photographs with GANs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6306–6314, 2018. [2](#), [4](#), [6](#), [7](#), [8](#)
- [9] Michaël Gharbi, Jiawen Chen, Jonathan T Barron, Samuel W Hasinoff, and Frédo Durand. Deep bilateral learning for real-time image enhancement. *ACM Transactions on Graphics (TOG)*, 36(4):118, 2017. [2](#), [4](#), [8](#)
- [10] Yuanming Hu, Hao He, Chenxi Xu, Baoyuan Wang, and Stephen Lin. Exposure: A white-box photo post-processing framework. *ACM Transactions on Graphics (TOG)*, 37(2):26, 2018. [2](#), [8](#)
- [11] Sung Ju Hwang, Ashish Kapoor, and Sing Bing Kang. Context-based automatic local image enhancement. In *European Conference on Computer Vision*, pages 569–582. Springer, 2012. [2](#)
- [12] Sung Ju Hwang, Ashish Kapoor, and Sing Bing Kang. Context-based automatic local image enhancement. In Andrew Fitzgibbon, Svetlana Lazebnik, Pietro Perona, Yoichi Sato, and Cordelia Schmid, editors, *Computer Vision – ECCV 2012*, pages 569–582, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg. [2](#)
- [13] Andrey Ignatov, Nikolay Kobyshev, Radu Timofte, Kenneth Vanhoey, and Luc 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. [2](#), [3](#), [7](#)
- [14] Andrey Ignatov, Nikolay Kobyshev, Radu Timofte, Kenneth Vanhoey, and Luc Van Gool. Wespe: weakly supervised photo enhancer for digital cameras. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 691–700, 2018. [2](#)
- [15] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. In *ICLR*, 2017. [8](#)
- [16] Kyeongbo Kong, Junggi Lee, Woo-Jin Song, Minsung Kang, Kyung Joon Kwon, and Seong Gyun Kim. Multitask bilateral learning for real-time image enhancement. *Journal of the Society for Information Display*, 2019. [2](#)
- [17] Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew P. Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, and Wenzhe 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. [6](#)
- [18] Ming-Yu Liu, Thomas Breuel, and Jan Kautz. Unsupervised image-to-image translation networks. In *Advances in Neural Information Processing Systems*, pages 700–708, 2017. [2](#), [7](#)
- [19] Jongchan Park, Joon-Young Lee, Donggeun Yoo, and In 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. [2](#), [8](#)
- [20] Stephen M Pizer, R Eugene Johnston, James P Ericksen, Bonnie C Yankaskas, and Keith E Muller. Contrast-limited adaptive histogram equalization: speed and effectiveness. In *Proceedings of the First Conference on Visualization in Biomedical Computing*, pages 337–345. IEEE, 1990. [2](#)
- [21] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, volume 9351 of LNCS, pages 234–241. Springer, 2015. (available on arXiv:1505.04597 [cs.CV]). [6](#), [8](#)
- [22] 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. [6](#)
- [23] Yichang Shih, Sylvain Paris, Frédo Durand, and William T. Freeman. Data-driven hallucination of different times of day from a single outdoor photo. *ACM Trans. Graph.*, 32(6):200:1–200:11, Nov. 2013. [4](#)
- [24] J Alex Stark. Adaptive image contrast enhancement using generalizations of histogram equalization. *IEEE Transactions on image processing*, 9(5):889–896, 2000. [2](#)
- [25] Ruixing Wang, Qing Zhang, Chi-Wing Fu, Xiaoyong Shen, Wei-Shi Zheng, and Jiaya Jia. Underexposed photo enhancement using deep illumination estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6849–6857, 2019. [2](#), [6](#), [7](#), [8](#)
- [26] Z. Wang, E.P. Simoncelli, and A.C. Bovik. Multiscale Structural Similarity for Image Quality Assessment. In *The Thirty-Seventh Asilomar Conference on Signals, Systems and Computers*, 2003. [6](#)
- [27] Jianzhou Yan, Stephen Lin, Sing Bing Kang, and Xiaoou Tang. A learning-to-rank approach for image color enhancement. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2987–2994, 2014. [2](#)

- [28] Zhicheng Yan, Hao Zhang, Baoyuan Wang, Sylvain Paris, and Yizhou Yu. Automatic photo adjustment using deep neural networks. *ACM Transactions on Graphics (TOG)*, 35(2):11, 2016. [2](#)
- [29] Lu Yuan and Jian Sun. Automatic exposure correction of consumer photographs. In *European Conference on Computer Vision*, pages 771–785. Springer, 2012. [2](#)
- [30] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018. [6](#)
- [31] Hang Zhao, Orazio Gallo, Iuri Frosio, and Jan Kautz. Loss Functions for Neural Networks for Image Processing. *IEEE Transactions on Computational Imaging*, 3(1), 2017. [6](#)
- [32] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei 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. [2](#), [7](#)