DeepLPF: Deep Local Parametric Filters for Image Enhancement

Sean Moran[1], Pierre Marza[†,1,2], Steven McDonagh[1], Sarah Parisot[1,3], Gregory Slabaugh[1]
Huawei Noah's Ark Lab[1] INSA Lyon[2] Mila Montréal[3]

CVPR SEATTLE WASHINGTON JUNE 16-18 2020

Introduction

Problem

- Automated image enhancement is an ill-posed problem requiring smooth local adjustments as well as global transformations.
- State of the art methods mainly rely on global (lack fine-grained details) or pixel level enhancements (noisy, difficult to interpret).

Inspiration

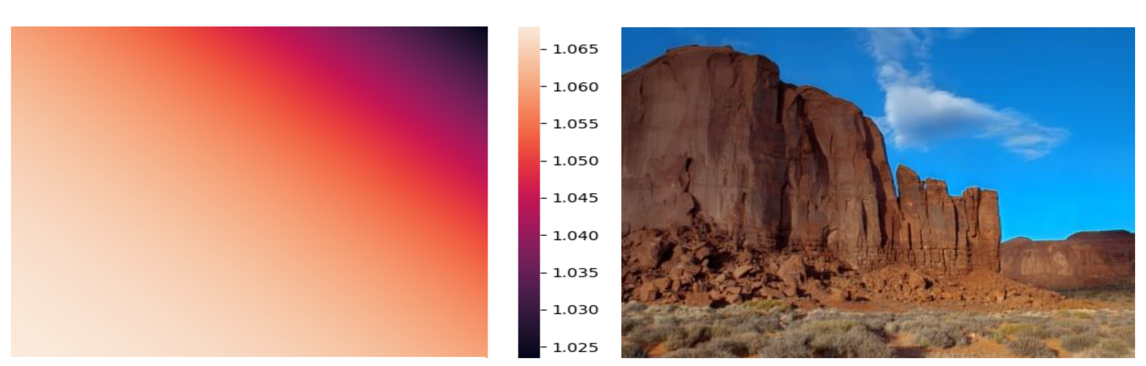
- Professional artists typically use a combination of global and local enhancement tools to manually enhance images.
- Highly popular local, parametrised enhancement tools allow for smooth local adjustments:

Brush tool Graduated filter Radial filter

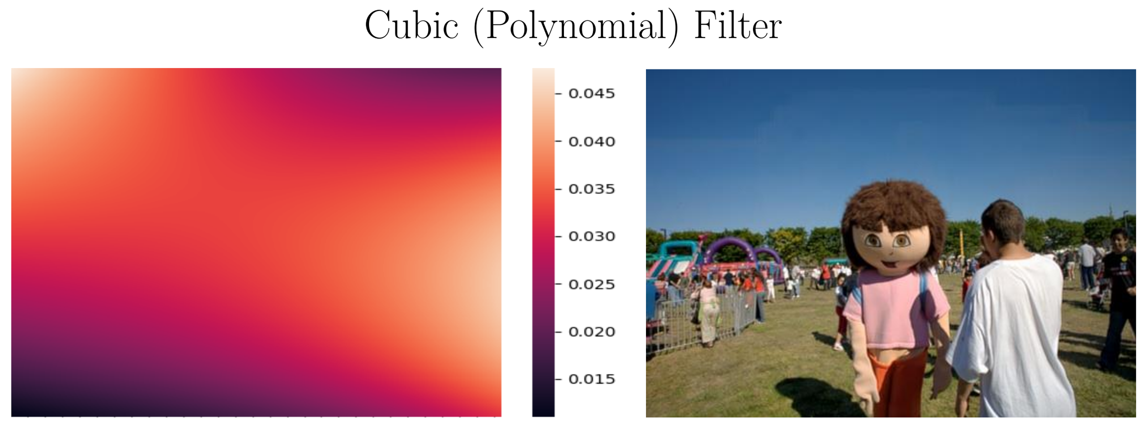
Proposed approach and Contributions

- Introduction of learnable parametric **Elliptical, Graduated, Polynomial** image filters to reproduce artist **local** image retouching practices.
- A novel architecture enabling regression of spatially localized image filter parameters for the target application of input image enhancement.
- Filter parameterization provides interpretable and intrinsically regularised filters and facilitates human feedback.
- A plug-and-play neural block with a filter fusion mechanism enabling the learning of multiple filters simultaneously.
- State of the art performance on three competitive benchmarks.

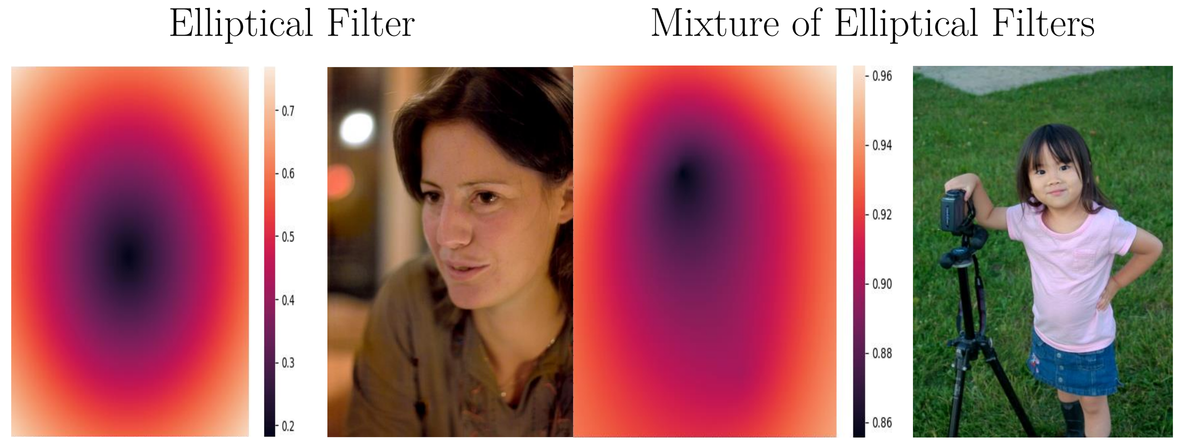
Graduated Filter



Cubic (Polynomial) Filter

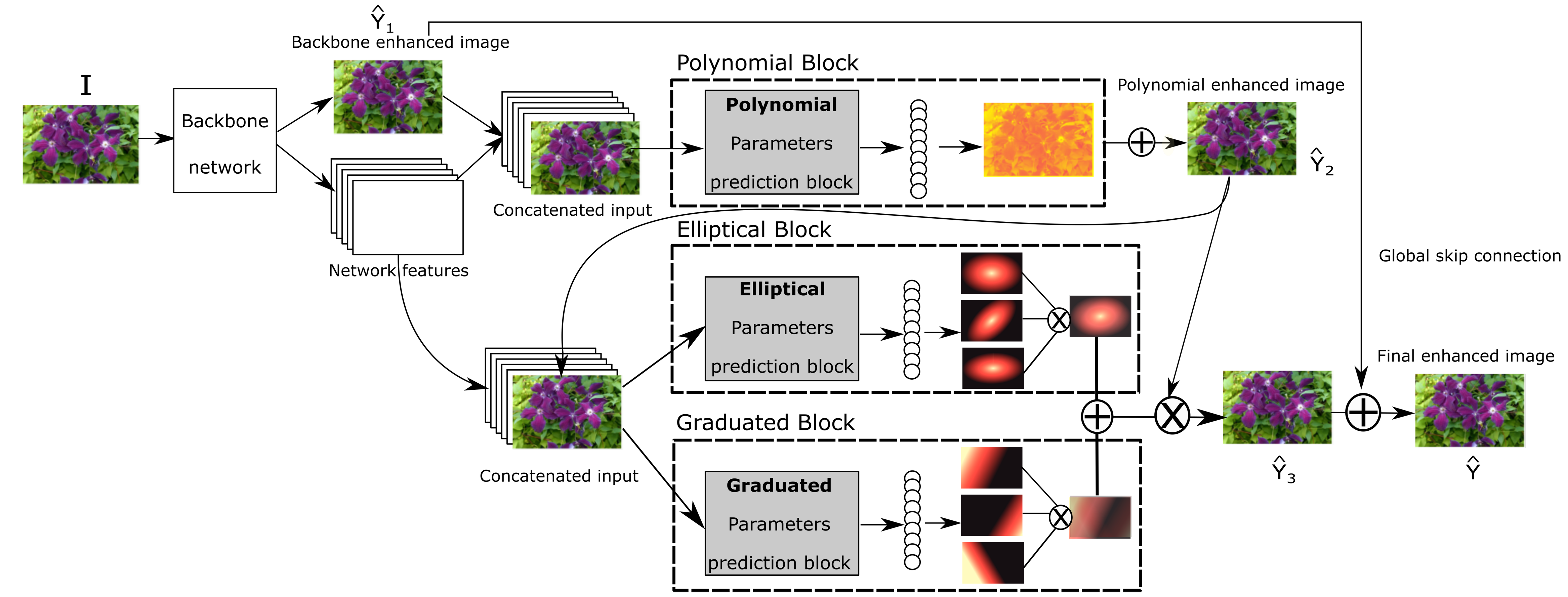


Elliptical Filter Mixture of Elliptical Filters



† work carried out during an internship at Huawei Noah's Ark Lab

Method Overview



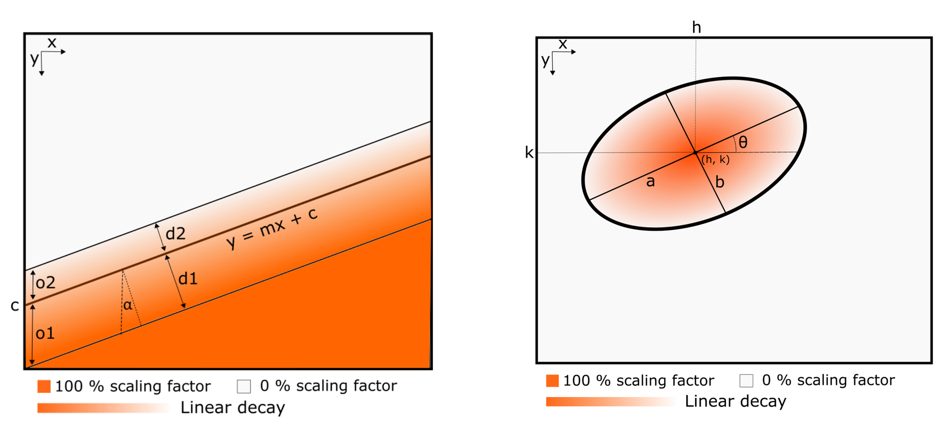
Problem decomposition

- Estimate a **H×W×C dimensional feature map** using a standard CNN backbone (e.g. ResNet, UNet).
- The first **single stream path** estimates the parameters of a **polynomial filter** that is subsequently applied to the pixels of the backbone enhanced image.
- The **two stream path** applies more constrained, local enhancements in the form of **elliptical and graduated filters**. The adjustment maps of these filters are estimated using **two parallel regression blocks** and then fused, resulting in a scaling map that is element-wise multiplied to the input.

Parametric Filters Instantiation

Graduated and Elliptical filters

- Instantiating the software Graduated and Radial filter tools respectively.
- Geometrically-regularized heatmaps: associating each pixel in the image with 3 scaling factors (one per RGB channel) to be applied.
- k instances of these filter types can be predicted and then combined.



(a) (b)

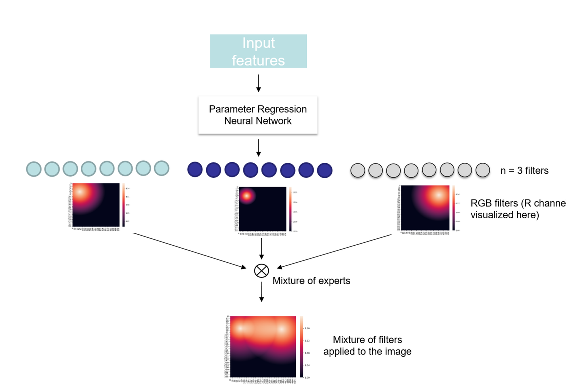
Polynomial filter

- General instantiation of the brush tool.
- A single filter estimated for a given input.
- Polynomial function mapping the intensity i and location (x, y) of each pixel within the image to a new intensity value i'.

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

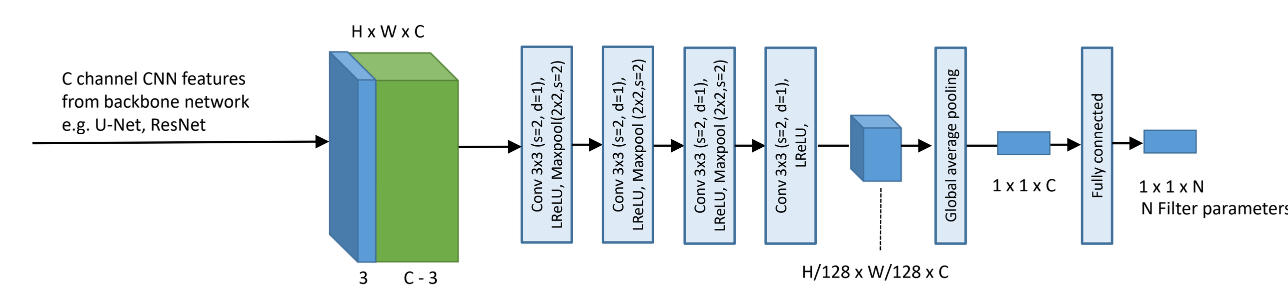
Fusion of graduated and elliptical filters

- Multiple instances of the same filter type : Element-wise multiplication resulting in a final heatmap
- Fusion of graduated and elliptical heatmaps : Simple addition



Filter Parameter Prediction Network

Each of the three filter blocks contains a filter parameter prediction network.



- Input : set of features inferred by a backbone network (e.g. U-Net) concatenated with the image to be enhanced.
- Regresses the parameter values of the corresponding filter type.
- k (hyperparameter) instances of a same filter type can be obtained by estimating $k * nb_{parameters}$ outputs.

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

Loss Function

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:

$$\mathcal{L} = \sum_{i=1}^{N} \{\omega_{\text{lab}} \underbrace{||Lab(\hat{Y}_i) - Lab(Y_i)||_1}_{\text{Local enhancement}} + \omega_{\text{ms-ssim}} \underbrace{(1 - \text{MS-SSIM}(L(\hat{Y}_i), L(Y_i)))}_{\text{Global enhancement}}\}$$

$Lab(\cdot)$: CIELab Lab channels of the input image
$L(\cdot)$: L channel of the image in CIELab colour space

Quantitative results

- **PSNR, SSIM, LPIPS state-of-the-art performance** for classical image retouching and low-light enhancement tasks
- **Efficient** architecture (few neural weights)

Model	PSNR↑	SSIM↑	LPIPS↓	# Weights
MIT-Adobe-5K-DPE dataset [4]				
DeepLPF	**23.93**	**0.903**	**0.582**	**1.8 M**
DPE [4]	23.80	0.900	0.587	3.3 M
MIT-Adobe-5K-UPE dataset [7]				
DeepLPF	**24.48**	0.887	**0.103**	**800K**
DPE [4]	22.15	0.850	–	3.3 M
DeepUPE [7]	23.04	**0.893**	0.158	1.0 M
SID dataset [3]				
DeepLPF	**26.82**	**0.702**	**0.564**	**2.0 M**
U-Net [3]	26.61	0.680	0.586	7.8 M

Qualitative results

References

[1] Mathieu Aubry et al. Fast local laplacian filters: Theory and applications. *ACM Trans. Graph.*, 2014.
[2] Vladimir Bychkovsky et al. Learning photographic global tonal adjustment with a database of input/output image pairs. In *CVPR*, 2011.
[3] Chen Chen et al. Learning to see in the dark. In *CVPR*, 2018.
[4] Yu-Sheng Chen et al. Deep photo enhancer: Unpaired learning for image enhancement from photographs with GANs. In *CVPR*, 2018.
[5] Yuanyuan Gao et al. Naturalness preserved non-uniform illumination estimation for image enhancement based on retinex. *IEEE Trans. on Multimedia*, 2017.
[6] Andrey Ignatov et al. Dslr-quality photos on mobile devices with deep convolutional networks. In *ICCV*, 2017.
[7] Ruixing Wang et al. Underexposed photo enhancement using deep illumination estimation. In *CVPR*, 2019.
[8] Shuhang Wang et al. Contrast-dependent saturation adjustment for outdoor image enhancement. *J. Opt. Soc. Am. A*, 2017.

Links

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

Paper: Code: