








Learning Parametric Functions for Color Image Enhancement

Simone Bianco¹ , Claudio Cusano² , Flavio Piccoli¹  ,
and Raimondo Schettini¹ 

¹ University of Milano - Bicocca, viale Sarca 336, 20126 Milan, Italy
{simone.bianco,flavio.piccoli,schettini}@disco.unimib.it

² University of Pavia, via Ferrata 1, 27100 Pavia, Italy
claudio.cusano@unipv.it

Abstract. In this work we propose a novel CNN-based method for image enhancement that simulates an expert retoucher. The method is fast and accurate at the same time thanks to the decoupling between the inference of the parameters and the color transformation. Specifically, the parameters are inferred from a downsampled version of the raw input image and the transformation is applied to the full resolution input. Different variants of the proposed enhancement method can be generated by varying the parametric functions used as color transformations (i.e. polynomial, piecewise, cosine and radial), and by varying how they are applied (i.e. *channelwise* or *full color*). Experimental results show that several variants of the proposed method outperform the state of the art on the MIT-Adobe FiveK dataset.

Keywords: Automatic retouching · Image enhancement · Parametric enhancement

1 Introduction

The color enhancement of a raw image is a crucial step that significantly affects the quality perceived by the final observer [2]. This enhancement is either done automatically, through a sequence of on-board operations or, when high quality is needed, manually by professional retouchers in post-production. The manual enhancement of an image, in contrast to a pre-determined pipeline of operations, leads to more engaging outcomes because it may reflect the imagination and the creativity of a human being. Manual processing however, requires the retoucher to be skilled in using post-production software and it is a time consuming process. Even if there are several books and manuals describing techniques of photo retouching [10], the decision factors are highly subjective and therefore they cannot be easily replicated in procedural algorithms. For these reasons the emulation of a professional retoucher is a challenging task in computer vision that nowadays resides mostly in the field of machine learning, and in particular in the area of convolutional neural networks.

In its naive formulation, the color enhancement task can be seen as an *image-to-image transformation* that is learned from a set of raw images and their corresponding retouched versions. Isola et al. [9] propose a generative method that uses an adversarial training schema to learn a mapping between the raw inputs and the manifold of enhanced images. The preservation of the content is ensured by evaluating the raw and its corresponding enhanced image in a pairwise manner. Zhu et al. [12] relax the constraint of having paired samples by ensuring the preservation of the content through the use of cyclic redundancy instead of pair-matching. Those image-to-image transformation algorithms led to important results, but they are very time- and memory-consuming and thus are not easily applicable to high-resolution images. Furthermore, sometimes they produce annoying artifacts that heavily degrade the quality of the output image.

As an evolution of these approaches, Gharbi et al. [6] propose a new pipeline for image enhancement in which the parameters of a color transformation are inferred through a convolutional neural network processing a downsampled version of the input image. The use of a shape-conservative transform and of a fast inference logic, make this method (and all methods following this schema) more suitable for image retouching. Following this approach, Bianco et al. [3] use a polynomial color transformation of degree three, whose inputs are the raw pixels projected into a monomial basis containing the terms of the polynomial expansion. Transforms are inferred in a patchwise manner and then interpolated to have a color transform for each raw pixel. The size of the patch becomes a tunable parameter to move the system toward accuracy or toward speed. Hu et al. [8] use reinforcement learning to define a meaningful sequence of operations to retouch the raw image. In their method, the reward used to promote the enhancement is the expected return over all possible trajectories induced by the policy under evaluation.

In this work, we propose a method that follows the same decoupling between the inference of the parameters and the application of the shape-conservative color transform adopted in the last described methods. We compare four different color transformations (polynomial, piecewise, cosine and radial) applied either *channelwise* or *full color*.

The rest of the paper is organized as follows: Sect. 2 describes the pipeline of the proposed method, the architecture of the CNN used to infer the parameters (Subsect. 2.2) and the parametric functions used as color transforms (Subsect. 2.1). Section 3 assesses the accuracy of the proposed method and all its variants comparing them with the relative state of the art.

2 Method

The image enhancement method we propose here is based on a Convolutional Neural Network (CNN) which estimates the parameters of a global image transformation that is then applied to the input image. With respect to a straightforward neural regression, this approach presents two main advantages. The former is that the parametric transformation automatically preserves the content of the images preventing the introduction of artifacts. The latter is that the whole procedure is very fast since the parameters can be estimated from a downsampled

version of the image, and only the final transformation needs to be safely applied to the full resolution image. The complete pipeline of the proposed method is depicted in Fig. 1: the input image is downsampled and fed to a CNN that estimates the coefficients to be applied to the chosen basis function (among polynomial, piecewise, cosine and radial) to obtain the color transformation to be applied to the full resolution input image.

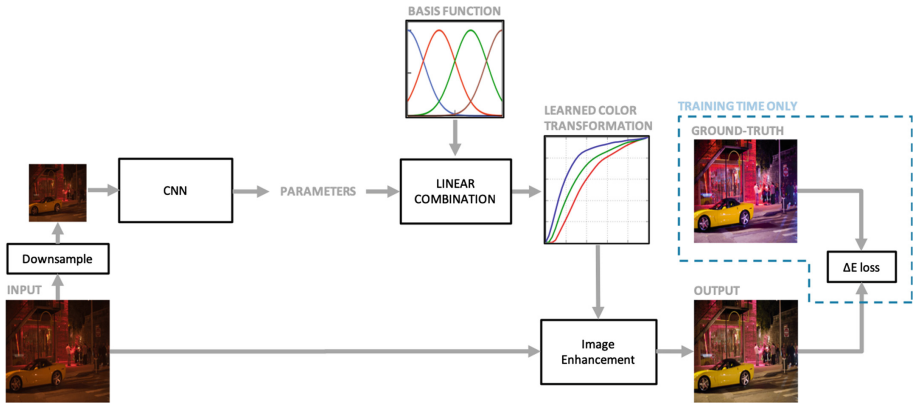


Fig. 1. Pipeline of the proposed method. The input image is downsampled and fed to a CNN that estimates the coefficients to be applied to the chosen basis function (among polynomial, piecewise, cosine and radial) to obtain the color transformation to be applied to the full resolution input image.

2.1 Architecture of the CNN

The structure of the neural network is that of a typical CNN. It includes a sequence of parametric linear operations (convolutions and linear products) interleaved by non-linear functions (ReLU and batch normalizations). Due to the limited amount of training data, we had to contain the number of learnable parameters. The final architecture is similar to that used by Bianco et al. [3] to address the problem of artistic photo filter removal. The neural network is relatively small in size, if compared to other popular ones [1].

Given the input image, downsampled to 256×256 pixels, a sequence of four convolutional blocks extracts a set of local features. Each convolutional block consists of convolution, ReLU activation and batch normalization. Each convolutional block increases the number of channels in the output feature map. All the convolutions have kernel size of 3×3 and stride of 2, except the first one that has a kernel of 5×5 pixels with stride 4 (the first convolution differs from the others also because it is not followed by batch normalization). The output of the convolutional blocks is a map of $8 \times 8 \times 64$ local features that are averaged into a single 64-dimensional vector by average pooling. Two linear layers, interleaved by a ReLU activation, compute the parameters of the enhancement transformation. Figure 2 shows a schematic view of the architecture of the network.

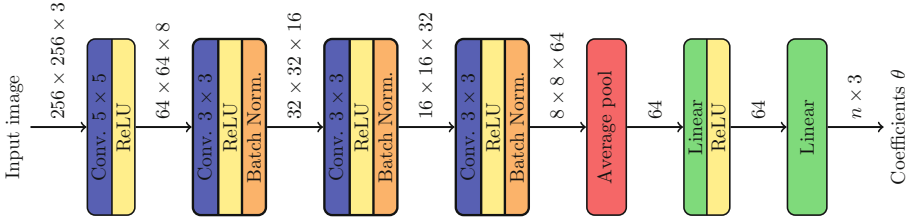


Fig. 2. Architecture of the convolutional neural network. The size of the output depends on the dimension n of the functional basis and on the kind of transformation ($n \times 3$ for channelwise, $n \times n \times n \times 3$ for full color transformations).

2.2 Image Transformations

We considered two main families of parametric functions: the *channelwise transformations* and the *full color transformations*. Channelwise transformations are defined by a triplet of functions, each one respectively applied to the Red, Green and Blue color channels. Full color transformations are also formed by triplets of functions, but each one considers all the color coordinates of the input pixel in the chosen color space. In both cases the functions are linear combinations of the elements $\phi_1, \phi_2, \dots, \phi_n$ of a n -dimensional basis. The coefficients θ of the linear combination are the output of the neural network. Full color transformations are potentially more powerful as they allow to model more complex operations in the color space. Channelwise transformations are necessarily simpler, since they are forced to modify each channel independently. However, channelwise transformations require a smaller number of parameters and are therefore easier to estimate.

We assume that the values of the color channels of the input pixels are in the $[0, 1]$ range.

In the case of channelwise transformations the color of the input pixel $\mathbf{x} = (x_1, x_2, x_3)$ is transformed into a new color $\mathbf{y} = (y_1, y_2, y_3)$ by applying the equation:

$$y_c = x_c + \sum_{i=1}^n \theta_{ic} \phi_i(x_c), \quad c \in \{1, 2, 3\}, \quad (1)$$

where $\boldsymbol{\theta} \in \mathbb{R}^{n \times 3}$ is the output of the CNN. Note that the term x_c outside the summation makes it so we have $\mathbf{x} \simeq \mathbf{y}$ when $\boldsymbol{\theta} \simeq \mathbf{0}$. This detail was inspired by residual networks [7] and allows for an easier initialization of the network parameters that speeds-up the training process.

Full color transformations take into account all the three color channels. Since modeling a generic $\mathbb{R}^3 \rightarrow \mathbb{R}^3$ transformation would require a prohibitively large basis, we restricted it to a combination of *separable* multidimensional functions in which each element of the three-dimensional basis is the product of three

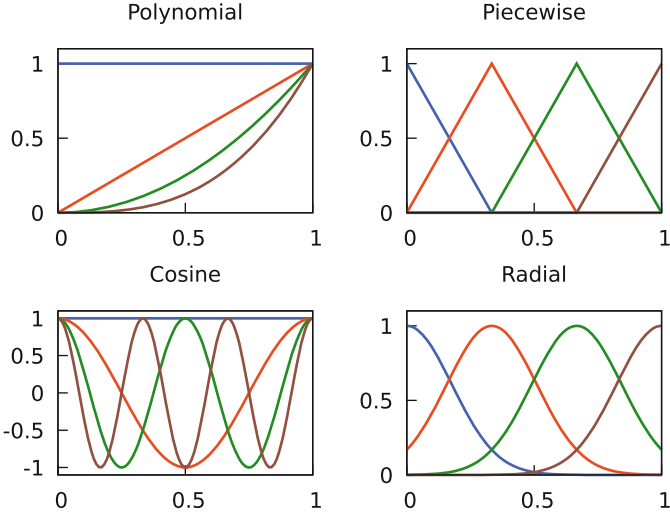


Fig. 3. The four function basis used in this work, in the case $n = 4$.

elements in a one-dimensional basis. A full color transformation can be therefore expressed as follows:

$$y_c = x_c + \sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^n \theta_{ijk} \phi_i(x_1) \phi_j(x_2) \phi_k(x_3), \quad c \in \{1, 2, 3\}, \quad (2)$$

where $\theta \in \mathbb{R}^{n \times n \times n \times 3}$ is the output of the CNN. Note that in this case the number of coefficients grows as the cube of the size n of the one-dimensional function basis.

We experimented with four basis: polynomial, piecewise linear, cosine and radial. See Fig. 3 for their visual representation. The polynomial basis in the variable x is formed by the integer powers of x :

$$\phi_i(x) = x^{i-1}, \quad i \in \{1, 2, \dots, n\}. \quad (3)$$

For the piecewise case, the basis includes triangular functions of unitary height, centered at equispaced nodes in the $[0, 1]$ range:

$$\phi_i(x) = \max\{0, 1 - |(n-1)x - i + 1|\}, \quad i \in \{1, 2, \dots, n\}. \quad (4)$$

In practice $\phi_i(x)$ is maximal at $(i-1)/(n-1)$ and decreases to zero at $(i-2)/(n-1)$ and $i/(n-1)$. The linear combination of these functions is a continuous piecewise linear function.

The cosine basis is formed by cosinusoidal functions of different angular frequencies:

$$\phi_i(x) = \cos(2\pi(i-1)x), \quad i \in \{1, 2, \dots, n\}. \quad (5)$$

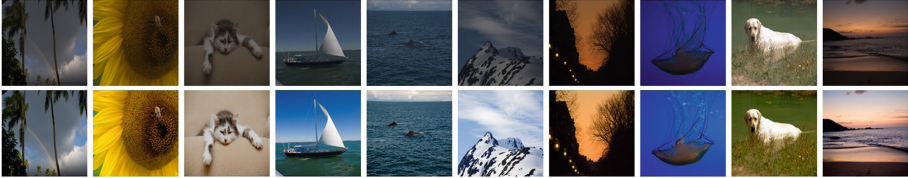


Fig. 4. Samples from the FiveK dataset. The top row shows the RAW input images while the bottom row reports their version retouched by Expert C.

The last basis is formed by radial Gaussian functions centered at equispaced nodes in the $[0, 1]$ range:

$$\phi_i(x) = \exp\left(-\frac{(x - (i-1)/(n-1))^2}{\sigma^2}\right), \quad \sigma = \frac{1}{n}, \quad i \in \{1, 2, \dots, n\}. \quad (6)$$

Note that the width of the Gaussians scales with the dimension of the basis.

3 Experimental Results

To assess the accuracy of the different variants of proposed method, we took a dataset of manually retouched photographs and we measured the difference between them and the automatically retouched output images. The dataset we considered is the FiveK [4] which consists of 5000 photographs collected by MIT and Adobe for the evaluation of image enhancement algorithms. Each image is given in the RAW, unprocessed format and in five variants manually retouched by five photo editing experts.

We used the procedure described by Hu *et al.* [8] to preprocess the images and to divide them into a training set of 4000 images, and a test set of 1000 images. We followed the common practice to use the third expert (Expert C) as reference, since he is considered the one with the most consistent retouching style. Some images from the dataset are depicted in Fig. 4 together with their versions retouched by Expert C.

As a performance measure we considered the average color difference between the images retouched by the proposed method and by Expert C. Color difference is measured by computing the ΔE_{76} and the ΔE_{94} color differences. ΔE_{76} is defined as the Euclidean distance in the CIELab color space [5]:

$$\Delta E_{76} = \sqrt{(L_1 - L_2)^2 + (a_1 - a_2)^2 + (b_1 - b_2)^2}, \quad (7)$$

where (L_1, a_1, b_1) and (L_2, a_2, b_2) are the coordinates of a pixel in the two images after their conversion from RGB to CIELab. We also considered the more recent ΔE_{94} :

$$\Delta E_{94} = \sqrt{\left(\frac{\Delta L}{K_L S_L}\right)^2 + \left(\frac{\Delta C}{K_C S_C}\right)^2 + \left(\frac{\Delta H}{K_H S_H}\right)^2}, \quad (8)$$

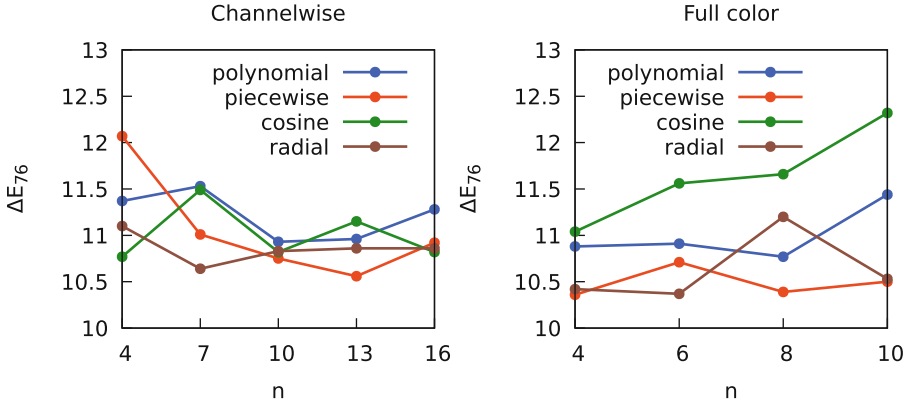


Fig. 5. Performance (average ΔE_{76}) obtained with the different function basis varying their dimension n .

where $K_L = K_C = K_H = 1$, $\Delta C = C_1 - C_2 = \sqrt{a_1^2 - b_1^2} - \sqrt{a_2^2 - b_2^2}$, $\Delta H = \sqrt{(a_1 - a_2)^2 + (b_1 - b_2)^2} - \Delta C^2$, $S_L = 1$, $S_C = 1 + K_1 C_1$, $S_H = 1 + K_2 C_1$ with $K_1 = 0.045$ and $K_2 = 0.015$. Finally, we also computed the difference in lightness:

$$\Delta L = |L_1 - L_2|. \quad (9)$$

3.1 Basis Function Comparison

As a first experiment, we compared the performance of the proposed method using the two kinds of transformations (channelwise and full color) and the four basis functions (polynomial, piecewise, cosine and radial). We trained the CNN in the configurations considered by using the Adam optimization algorithm [11] using the average ΔE_{76} between the groundtruth and the color enhanced image as loss function. The training procedure consisted of 40000 iterations with a learning rate of 10^{-4} , a weight decay of 0.1, and minibatches of 16 samples.

We repeated the experiments multiple times by changing the cardinality n of the basis function. Taking into account that that our training dataset is rather small, for channelwise transformations we limited our investigation to $n \in \{4, 7, 10, 13, 16\}$ while for full color transformations we had to use even

smaller values ($n \in \{4, 6, 8, 10\}$) limiting in this way also memory consumption. The results obtained on the test set are summarized in Fig. 5. The plots show different behaviors for the two kinds of transformations: for channelwise transformations increasing the size of the basis tends to improve the accuracy (with the possible exception of polynomials), while for full color transformations small basis seem to perform better.

Table 1 reports in details the performance obtained with the values of n that, on average, have been found to be the best for the two kinds of transformations: $n = 10$ for channelwise and $n = 4$ for full color transformations. For channelwise transformations, it seems that all the basis perform quite similarly. On the other hand, in the case of full color transformations polynomials and cosines are clearly less accurate than piecewise and radial functions. Full color transformations allowed to obtain the best results for all the three performance measures considered.

Table 1. Results obtained on the test set by the variations of the proposed method. For each metric, the lowest error is reported in bold.

Transformation type	Basis	ΔE_{76}	ΔE_{94}	ΔL
Channelwise ($n = 10$)	Polynomial	10.93	8.36	5.77
	Piecewise	10.75	8.24	5.63
	Cosine	10.82	8.14	5.58
	Radial	10.83	8.16	5.53
Full color ($n = 4$)	Polynomial	10.88	8.33	5.79
	Piecewise	10.36	8.02	5.48
	Cosine	11.04	8.45	5.48
	Radial	10.42	7.99	5.47

Figure 6 visually compares the results of applying channelwise transformations to a test image. The functions obtained with the piecewise and the radial basis look very smooth. The cosine basis produces a less regular behavior, and the polynomial basis seems unable to keep under control all the color curves.

Figure 7 compares, on the same test image, the behavior of full color transformations. We cannot easily visualize these transformations as simple plots. However, we can notice how a better outcome is obtained for low values of n . For large values of n some basis (e.g. cosine) even fail to keep the pixels within the $[0, 1]$ range. This can be noticed in the brightest region of the image, where it causes the introduction of unreasonable colors (we could have clipped these values, but we preferred to make it visible to better show this behavior).

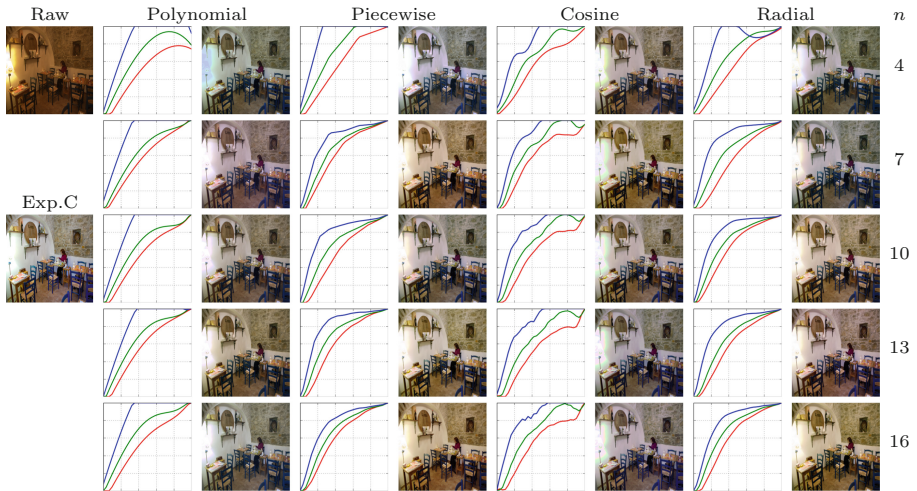


Fig. 6. Comparison of channelwise transformations applied to a test image. On the left of each processed image are shown the transformations applied to the three color channels.



Fig. 7. Comparison of full color transformations applied to a test image.

3.2 Comparison with the State of the Art

As a second experiment we compared the results obtained by the best versions of the proposed method with other recent methods from the state of the art. We included in the comparison methods based on the use of Convolutional

Neural Networks that made publicly available their source code. The methods we considered are:

- Pix2pix [9], that has been proposed by Isola *et al.* for image-to-image translation; even though not specifically proposed for image enhancement, its flexibility make this method suitable for this task as well.
- CycleGAN [12], that is similar to Pix2pix but targeted to the more challenging learning from unpaired examples.
- Exposure [8] is also trained from unpaired examples. It uses reinforcement learning, and has been originally evaluated on the FiveK dataset.
- HDRNet [6], that Gharbi *et al.* designed to make it possible to process high resolution images. Among the others, the authors evaluated it on the FiveK dataset.
- Unfiltering [3], is a method originally proposed for image restoration. However, it was possible to retrain it for image enhancement.

Table 2 reports the results obtained by these methods and compares them with those obtained by the proposed approach. The table includes an example for each type of color transformation. In both cases the piecewise basis is considered with $n = 10$ for the channelwise case and $n = 4$ for the full color transformation. Both variants outperformed all the competing methods in terms of average ΔE_{76} and ΔE_{94} . Among the other methods Pix2pix was quite close (less than half unit of ΔE) and slightly outperformed the proposed one in terms of ΔL . These results confirm the flexibility of the proposed architecture. HDRNet and Unfiltering also obtained good results. Exposure and cycleGAN, instead, are significantly less accurate than the other methods. This confirms the difficulties in learning from unpaired examples.

Table 2. Accuracy in reproducing the test images retouched by expert C by methods in the state of the art.

Method	ΔE_{76}	ΔE_{94}	ΔL
Exposure [8]	16.98	13.42	9.54
CycleGAN [12]	16.23	12.30	9.20
Unfiltering [3]	13.17	10.42	6.76
HDRNet [6]	12.14	9.63	6.27
Pix2pix [9]	11.13	8.53	5.55
Channelwise (piecewise $n = 10$)	10.75	8.24	5.63
Full color (piecewise $n = 4$)	10.36	8.02	5.48

Figure 8 shows some test images processed by the methods included in the comparison.

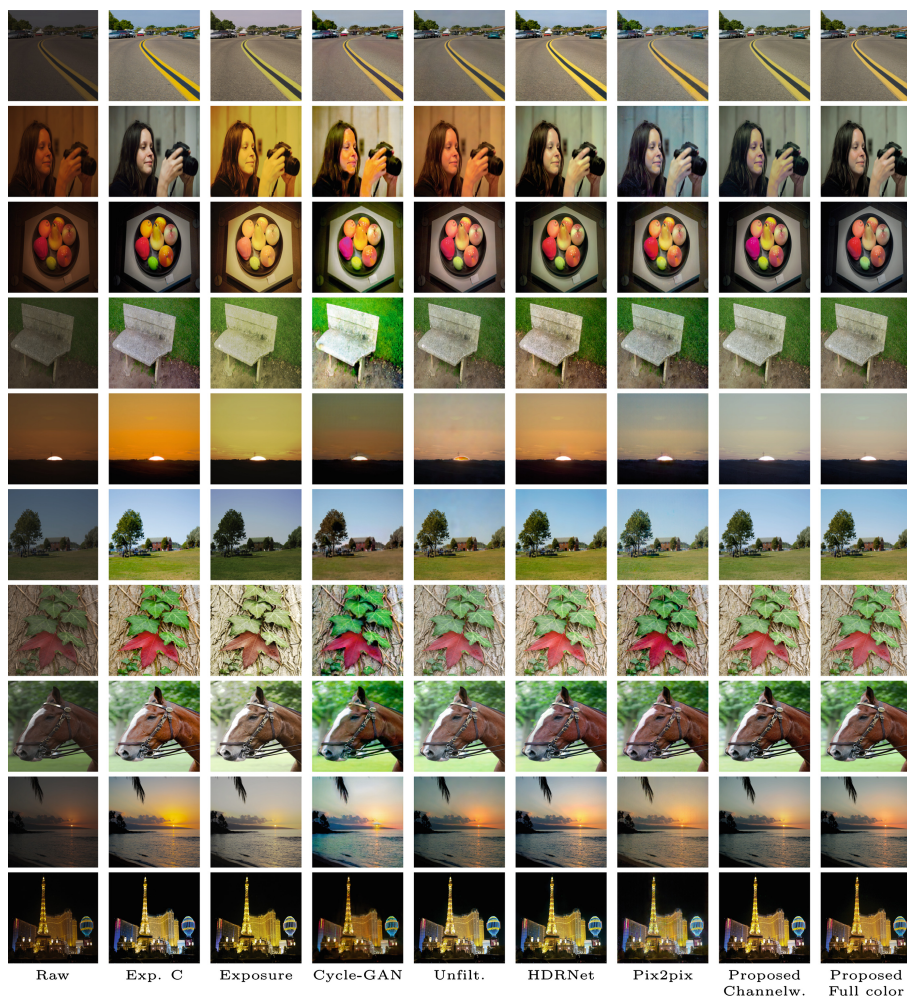


Fig. 8. Results of applying image enhancement methods in the state of the art to some test images. For the proposed method, both variants refer to the piecewise basis ($n = 10$ for the channelwise version, $n = 4$ for the full color transformation).

4 Conclusions

In this paper we presented a method for image enhancement of raw images that simulates the ability of an expert retoucher. This method uses a convolutional neural network to infer the parameters of a color transformation on a downsampled version of the input image. In this way, the inference stage becomes faster

in test time and less data-hungry in training time while remaining accurate. The main contributions of this paper include:

- a fast end-to-end trainable shape-conservative retouching algorithm able to emulate an expert retoucher;
- the comparison among eight different variants of the method obtained combining four parametric color transformations (i.e. polynomial, piecewise, cosine and radial) and how they are applied (i.e. *channelwise* or *full color*).

The relationship between the cardinality of the basis function and the performance deserve a further investigation. This investigation would need a much larger dataset that we hope to collect in the future. Notwithstanding this, the preliminary results obtained show that the proposed method is able to reproduce with great accuracy a huge variety of retouching styles, outperforming the algorithms that represent the state of the art on the MIT-Adobe FiveK dataset.

References

1. Bianco, S., Cadene, R., Celona, L., Napoletano, P.: Benchmark analysis of representative deep neural network architectures. *IEEE Access* **6**, 64270–64277 (2018)
2. Bianco, S., Ciocca, G., Marini, F., Schettini, R.: Image quality assessment by pre-processing and full reference model combination. In: *Image Quality and System Performance VI*, vol. 7242, p. 72420O. International Society for Optics and Photonics (2009)
3. Bianco, S., Cusano, C., Piccoli, F., Schettini, R.: Artistic photo filter removal using convolutional neural networks. *J. Electron. Imaging* **27**(1), 011004 (2017)
4. Bychkovsky, V., Paris, S., Chan, E., Durand, F.: Learning photographic global tonal adjustment with a database of input/output image pairs. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 97–104 (2011)
5. Fairchild, M.D.: *Color Appearance Models*. Wiley, Chichester (2013)
6. Gharbi, M., Chen, J., Barron, J.T., Hasinoff, S.W., Durand, F.: Deep bilateral learning for real-time image enhancement. *ACM T. Graphic. (TOG)* **36**(4), 118 (2017)
7. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778 (2016)
8. Hu, Y., He, H., Xu, C., Wang, B., Lin, S.: Exposure: a white-box photo post-processing framework. *ACM T. Graphic. (TOG)* **37**(2), 26 (2018)
9. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5967–5976 (2017)
10. Kee, E., Farid, H.: A perceptual metric for photo retouching. *Proc. Nat. Acad. Sci.* **108**(50), 19907–19912 (2011)
11. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. In: *International Conference on Learning Representation* (2015)
12. Zhu, J.Y., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. In: *IEEE International Conference on Computer Vision (ICCV)* (2017)