Journal of Electronic Imaging

JElectronicImaging.org

Artistic photo filter removal using convolutional neural networks

Simone Bianco
Claudio Cusano
Flavio Piccoli
Raimondo Schettini

SPIE. IS&T imaging.org

Simone Bianco, Claudio Cusano, Flavio Piccoli, Raimondo Schettini, "Artistic photo filter removal using convolutional neural networks," *J. Electron. Imaging* **27**(1), 011004 (2017), doi: 10.1117/1.JEI.27.1.011004.

Journal of Electronic Imaging 27(1), 011004 (Jan/Feb 2018)

Artistic photo filter removal using convolutional neural networks

Simone Bianco,[a,*] Claudio Cusano,[b] Flavio Piccoli,[a] and Raimondo Schettini[a]
[a]University of Milano-Bicocca, Department of Informatics, Systems and Communication, Milano, Italy
[b]University of Pavia, Department of Electrical, Computer and Biomedical Engineering, Pavia, Italy

Abstract. We present a method for the automatic restoration of images subjected to the application of photographic filters, such as those made popular by photo-sharing services. The method uses a convolutional neural network (CNN) for the prediction of the coefficients of local polynomial transformations that are applied to the input image. The experiments we conducted on a subset of the Places-205 dataset show that the quality of the restoration performed by our method is clearly superior to that of traditional color balancing and restoration procedures, and to that of recent CNN architectures for image-to-image translation. *© 2017 SPIE and IS&T* [DOI: 10.1117/1.JEI.27.1.011004]

Keywords: photographic filters; convolutional neural networks; image restoration.
Paper 170637SS received Aug. 1, 2017; accepted for publication Nov. 22, 2017; published online Dec. 23, 2017.

1 Introduction

Photo-sharing services allow their users to archive, organize, and share their collections of pictures. Beside these core functionalities, they also offer many other features, such as editing, tagging, searching by content, creating albums, etc. One popular feature is the option to apply photographic filters to change the mood of the pictures in a completely automatic way. Several preset filters are available, corresponding to various combinations of visual effects, including shifts in the color distribution, adjustments in brightness and contrast, introduction of blur or noise, etc. These filters may enhance the expressiveness of pictures, but in many cases, they also make it impossible to recover the original images. Moreover, the disruption of the low-level image content makes automatic understanding problematic. In fact, with their experiments on the ImageNet large scale visual recognition competition classification data, Chen et al.[1] have shown how the application of photographic filters decreases by a large margin of the accuracy of state-of-the-art image recognition systems.

In this work, we propose a method that uses a convolutional neural network (CNN) for the automatic removal of photographic filters. Besides the already mentioned work by Chen et al.,[1] the scientific literature about this topic is quite limited. To the best of our knowledge, we can only refer to the work by Bianco et al.,[2] who experimented with several CNN architectures trained to distinguish among a set of photographic filters. They have shown how a retrained version of AlexNet was able to achieve about 99% of accuracy in deciding which filter (if any) was applied to the input image. For the same task, they also have shown how the simple fine-tuning of an already trained network performs quite badly. Besides photographic filters, many methods have been proposed for image restoration and enhancement. For the sake of brevity, here, we focus only on those that include CNNs in their processing pipelines.

Computational color constancy is a task that is somewhat related to the removal of a photographic filter. In computational color constancy, the aim is to correct the image by making it look as if the scene was taken under a canonical illuminant, which usually is achromatic. The desired effect is therefore to remove dominant colors caused by non-neutral illuminants. Recently, CNNs have been used with success in computational color constancy by Lou et al.,[3] Bianco et al.,[4,5] Shi et al.,[6] and Oh and Kim.[7] All these works outperformed the traditional color constancy methods on several reference datasets.

Colorization is another task resembling the recovery of images processed by photographic filters. Colorization methods take as input a gray-level image and assign plausible colors to the pixels according to its content. Since some photographic filters remove color, the recovery of the original image requires some form of colorization. Zhang et al.[8] approached the problem of colorization by posing it as a classification task and by using class-rebalancing during the training of the CNN to increase the diversity of colors in the result. The output of their network was able to fool human observers in 32% of the cases. Larsson et al.[9] trained a network to predict per-pixel color histograms and used this intermediate output to automatically generate a color image. Iizuka et al.[10] designed a colorization network, where local information dependent on small image patches are merged with global priors computed using the entire image. They evaluated their model on a large set of images showing that it can produce very credible results. They compared favorably against the state of the art and also performed a user study that corroborates their results.

Neural networks have been used for a variety of other image processing tasks. Liu et al.[11] combined a convolutional network with several recurrent neural networks that act as learnable infinite impulse response filters for denoisinig, inpainting, and edge-preserving smoothing. Gao and Grauman[12] leveraged deep models to solve image restoration

*Address all correspondence to: Simone Bianco, E-mail: bianco@disco.unimib.it

tasks without overfitting. They devised a symmetric encoder–decoder network and proposed an on-demand learning algorithm that turns a fixated model into one that performs well on various tasks, including image inpainting, pixel interpolation, image deblurring, and image denoising.

Among neural networks, generative adversarial networks (GANs) are defined in terms of a complex loss function represented by an auxiliary network, which is trained together with the main model in a game-theoretical framework.[13] This approach allowed users to obtain astounding results in a variety of image processing tasks, often formulated as a form of image-to-image translation. For instance, Isola et al.[14] proposed a GAN model, where the adversarial network analyzes small patches of the output image. Their model has been evaluated with success in several tasks including translation between daytime and nocturnal photos, sketches of objects and their pictures, maps and aerial photographs, etc. Their method requires a training set with paired input and output images. Another method, cycle GAN,[15] relaxed this requirement and allowed learning of image-to-image translation problems without paired input/output images. Ignatov et al.[16] proposed a photo enhancement solution to effectively transform cameras from common smartphones into high quality DSLR cameras. Their end-to-end deep learning approach uses a composite perceptual error function that combines content, color, and texture losses.

The method we propose here is able to automatically recover the images as they were before the application of photographic filters. The recovery is performed by applying a parametric local transformation whose parameters are adaptively estimated by a CNN that analyzes the input image. We will show how this solution allows restoration of the original properties of the image even when it has been subject to very disruptive filters preventing, at same time, any alteration of the original content.

To demonstrate the effectiveness of our approach, we conducted several experiments on a subset of 20,000 images taken from the Places-205 dataset[17] to which we applied 22 different photographic filters. The method was able to convincingly remove a variety of editing effects without any prior knowledge about which ones were applied to the input image. The results will show that, not only our method obtains a better recovery than other methods in the state-of-the-art but also that it allows significant improvement of the performance of image recognition methods for filtered images.

The paper is organized as follows: Sec. 2 reports all of the information about the photographic filters and the data used in the experimentation, and Sec. 3 describes our method for removing the filters. Section 4 reports the results obtained and discusses their implications. Finally, Sec. 5 concludes the paper by summarizing our findings and by suggesting future directions of research.

2 Photographic Filters

Many photo sharing services, such as Instagram, give their users the option to apply photographic filters to their own pictures. These filters consist of a pipeline of photo editing operations that are applied in a completely automatic way. Editing operations that are often used in photographic filters include: global transformations of the color distribution by using "color levels" and "color curves" or by changing the pixels' hue, saturation, and lightness; global adjustment of the image brightness and contrast; introduction of blur or noise; introduction of a "vignette" effect (i.e., darkening of the border of the image); spatially varying modification of color with the use of a gradient; conversion to black and white; and introduction of a "flare" effect in the central part of the image.

In this work, we considered 22 photographic filters defined by photo editing enthusiasts to match those made available by the Instagram® photo-sharing service. In particular, we considered the filters named 1977, Amaro, Apollo, Brannan, Earlybird, Gotham, Hefe, Hudson, Inkwell, Lofi, Lord Kelvin, Mayfair, Nashville, Poprocket, Rise, Sierra, Sutro, Toaster, Valencia, Walden, Willow, and Xpro-II. As a special case of filter, we also included the original images without any further processing. Table 1 summarizes the filters in terms of their editing operations, whereas Fig. 1 reports, for each one, a brief description taken from the Instagram® website.

During the experimentation, we used the Places-205 dataset.[17] Places-205 has been designed to represent places and scenes found in the real world. It includes over one million images labeled with 205 different categories. Each category is represented by at least 5000 images. For our experiments, we randomly sampled 20,000 images. After that, we processed them with the 22 filters to form a dataset of 460,000 filtered images (including the original ones). The images were randomly divided into training, validation, and test sets with ratios 75%, 5%, and 20% having care to place all the filtered variants of the same image in the same set.

3 Proposed Method

We propose here a method for the automatic removal of photographic filters. The method takes as input a color image that has been possibly processed with a photographic filter and produces as output a color image representing the same content but with the style modified to reproduce the appearance of a "natural," unfiltered image. Note that no knowledge is required about which filter (if any) needs to be removed.

Not all the editing operations involved in the computations of the filters are invertible. In fact, some of them cause a loss of information, making unfiltering an ill-posed problem. For instance, to an image processed by a filter that includes a conversion to gray-level (such as Gotham or Inkwell) corresponds to many plausible unfiltered images. However, as we will show in Sec. 4, our method is often able to guess a reasonable recovery of the missing information by inferring it from the semantic content of the input image (for instance, by recognizing the sky in a gray-level image and by coloring it in blue).

Many image-to-image deep learning models have been recently proposed.[14–16] Their results are often remarkable, but they come at the price of a high computational cost. In fact, all the information in the input image that is required to generate the output has to be preserved through all the layers, either by using large intermediate representations[18] or by using skip connections.[19] Applied to the problem at hand, this fact implies that all the fine details that are not affected by the photographic filters need to be preserved from input to output. To address this issue, we diverged from the popular image-to-image approach: instead of directly estimating the pixel values of the original unfiltered

Table 1 Summary of the basic image processing operations used in the 23 photographic filters considered.

Filter name	Color levels	Color curves	Brightness/ contrast	Blur/ noise	Hue/sat/ lightness	Vignette	Color layer	Gradient	Black and white	Flare
Original	·	·	·	·	·	·	·	·	·	·
1977	·	✓	·	·	·	·	·	·	·	·
Amaro	·	✓	·	·	·	✓	·	·	·	·
Apollo	·	·	·	·	·	✓	✓	·	·	·
Brannan	·	✓	✓	·	✓	·	·	·	·	·
Earlybird	·	✓	✓	·	✓	✓	✓	·	·	·
Gotham	·	✓	·	✓	·	·	·	·	✓	·
Hefe	·	·	✓	·	✓	✓	·	·	·	·
Hudson	·	✓	·	·	·	·	✓	·	·	·
Inkwell	·	✓	✓	·	·	·	·	·	✓	·
Lofi	·	✓	✓	·	·	✓	·	✓	·	·
Lord Kelvin	·	✓	·	·	·	·	·	·	·	·
Mayfair	✓	✓	·	✓	✓	·	·	·	·	·
Nashville	✓	✓	✓	·	·	·	·	·	·	·
Poprocket	·	·	·	·	·	·	·	✓	·	·
Rise	·	·	·	✓	✓	✓	✓	✓	·	·
Sierra	·	✓	·	·	·	✓	·	✓	·	·
Sutro	·	✓	✓	·	✓	✓	✓	·	·	·
Toaster	·	✓	·	·	·	✓	✓	✓	·	·
Valencia	✓	✓	·	·	·	·	·	·	·	·
Walden	✓	✓	·	·	·	·	·	✓	·	·
Willow	·	·	·	✓	·	✓	·	✓	✓	✓
X-pro II	·	✓	·	·	·	·	·	·	·	·

image, our model estimates the parameters of a set of local transformations that, when applied to the input image, approximate the desired output. Note that a single global transformation would not be suitable, since many filters (e.g., Amaro) are spatially varying. However, a small number of local transformations may be enough to reverse the photographic filters considered keeping manageable, at the same time, the complexity of the model and the number of parameters that needs to be learned. We chose to work with polynomial transformations, since they have been demonstrated to be very effective for color processing.[20–22] More precisely, we decided to use a grid of $T \times T$ polynomials that are then bilinearly upsampled to produce a per-pixel transformation. This can be done since the set of polynomials is closed under linear combinations.

In addition to the relatively low computational requirements, our approach also has the advantage of keeping the training procedure simple. In fact, one of the problems of image-to-image methods is that they require a complex loss function (often in the form of an adversarial network) to avoid losing the details of the image.[14,16] Our model, instead, can be trained simply by minimizing the mean squared error (MSE) between the desired and the actual output since details are preserved by the local transformations.

Given an input image of size 256×256, the method works as follows: first, for each pixel x, the (x_R, x_G, x_B) components are projected into the monomial basis $(1, x_R, x_G, x_B, x_R x_G, x_R x_B, x_G x_B, x_R^2, x_G^2, x_B^2, \ldots, x_R^D, x_G^D, x_B^D)$ for a set degree D. The result of this operation, that we call "polynomial expansion," is an image of $\binom{D+3}{D}$ channels. After that, the method applies a sequence of five convolutional blocks each one including a 3×3 convolution with 200 output channels, preceded by batch normalization and followed by the rectified linear unit (ReLU) activation function. Convolutions are applied with a stride of two in both spatial dimensions to progressively reduce the size of the image. The resulting $7 \times 7 \times 200$ image is flattened into a vector that is processed by a sequence of two linear layers (followed by a ReLU), producing a vector of 2000 components. A further linear transformation produces the coefficients of $3 \times T \times T$ polynomials of degree D (T^2 polynomials for each of the three color channels, each one defined by $\binom{D+3}{D}$ coefficients). Bilinear upsampling is then applied to smoothly interpolate the $T \times T$ transformations over the 256×256 input in such a way that each pixel has its own triplet of polynomials (for the R, G, and B channels). Finally, the polynomial expansion of the input image is

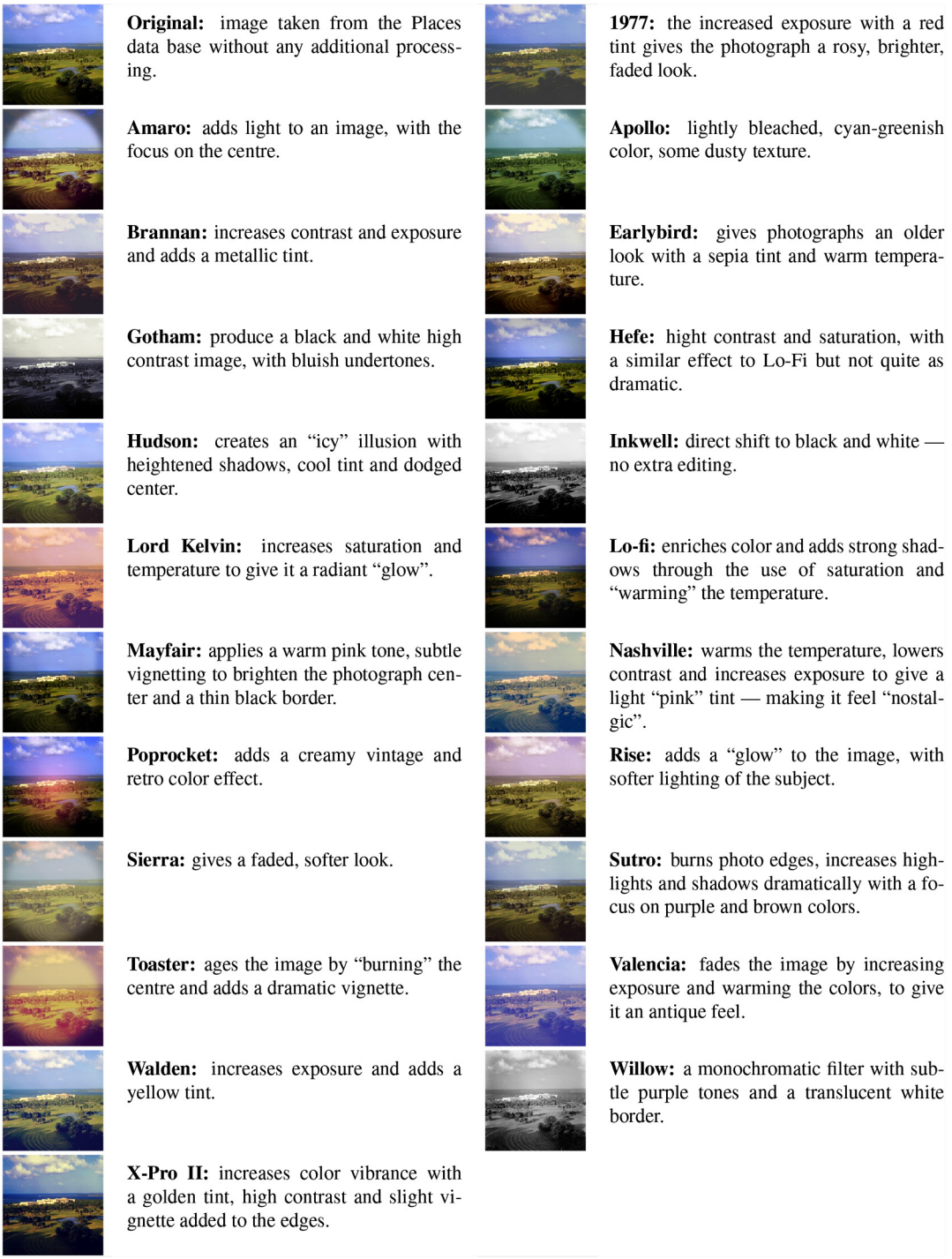


Fig. 1 Examples of the photographic filters considered in this work. The text has been taken from the Instagram® website.

multiplied by the interpolated coefficients yielding to the unfiltered image. Note that, due to the fully connected linear layers, the coefficients of each local transformation depend on the whole image. This facilitates the modeling of global corrections, when appropriate. Note also that the bilinear upsampling of the polynomials makes it easy to preserve the details in the input image.

We experimented with polynomial transformations up to the third degree, implying the use of the following polynomial expansions:

$$
\begin{aligned}
E_1(x) &= (1, x_R, x_G, x_B), \quad (D = 1),\\
E_2(x) &= (1, x_R, x_G, x_B, x_R x_G, x_R x_B, x_G x_B, x_R^2, x_G^2, x_B^2),\\
&\quad (D = 2),\\
E_3(x) &= (1, x_R, x_G, x_B, x_R x_G, x_R x_B, x_G x_B, x_R^2, x_G^2, x_B^2,\\
&\quad x_R x_G x_B, x_R^2 x_G, x_R^2 x_B, x_R x_G^2, x_R x_B^2, x_G^2 x_B, x_G x_B^2,\\
&\quad x_R^3, x_G^3, x_B^3), \quad (D = 3).
\end{aligned} \tag{1}
$$

The bilinear interpolation produces, for each pixel of the input image, a set of coefficients $\{k^c_{ijk}\}$ with $i+j+k \leq D, c \in \{R,G,B\}$. The components (y_R, y_G, y_B) of a pixel in the output image are finally computed as in the following expression:

$$y_c = \sum_{i+j+k \leq D} k^c_{ijk} x^i_R x^j_G x^k_B, c \in \{R,G,B\}, \quad (2)$$

which can be casted as an inner product between the set of coefficients (suitably encoded as a vector) and the polynomial expansion in Eq. (1).

The whole method is summarized in Fig. 2 and in Table 2. We used stochastic gradient descent with minibatches to train the CNN, by minimizing the MSE [Eq. (3)] between the output pixels y and the corresponding ground truth $\hat{y}$:

$$\text{MSE} = \frac{1}{3 \times 256^2} \sum_{i=1}^{256} \sum_{j=1}^{256} \sum_{c \in \{R,G,B\}} [y_c(i,j) - \hat{y}_c(i,j)]^2. \quad (3)$$

4 Experimental Results

To assess the effectiveness of the proposed method, we run various experiments, where test images are processed and the result is compared against the original image, before the application of photographic filters. We evaluated three aspects:

- faithfulness of the result of the unfiltering process with respect to the ground truth original image, measured with objective error metrics;
- "naturalness" of the result, measured by a neural network trained to identify filtered images;

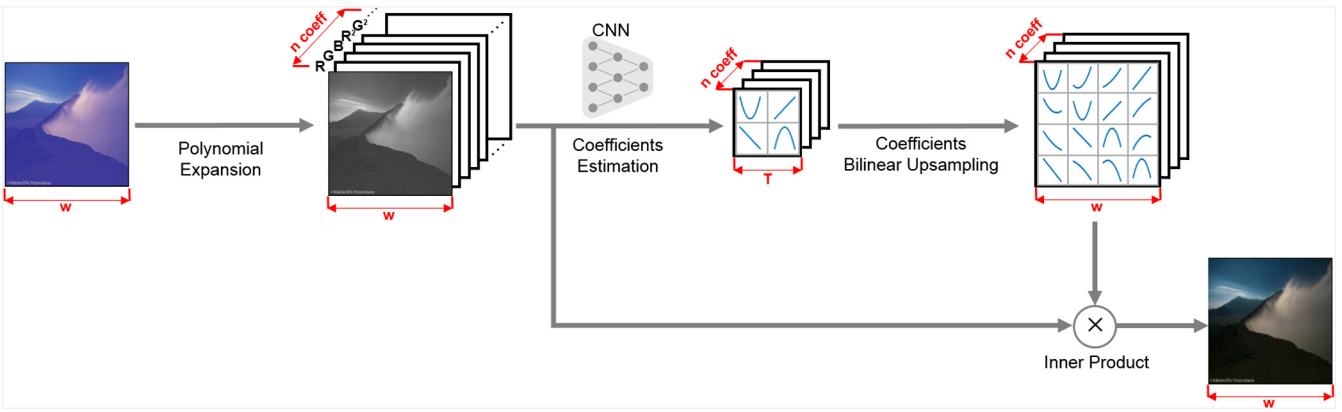


Fig. 2 Pipeline of the unfiltering process. The input image is unfiltered through a set of local polynomial color transformations whose coefficients are estimated by the CNN.

Table 2 Structure of the CNN. D denotes the degree of the polynomial transformations while T^2 is their number of local transformations. All convolutional layers have filters of dimension 3×3 and stride 2.

Stage	Operation	Output size
Preprocessing	Input	$256 \times 256 \times 3$
	Polynomial expansion	$256 \times 256 \times \binom{D+3}{D}$
Conv. network	Batch norm. + conv. + ReLU	$127 \times 127 \times 200$
	Batch norm. + conv. + ReLU	$63 \times 63 \times 200$
	Batch norm. + conv. + ReLU	$31 \times 31 \times 200$
	Batch norm. + conv. + ReLU	$15 \times 15 \times 200$
	Batch norm. + conv. + ReLU	$7 \times 7 \times 200$
	Linear + ReLU	2000
	Linear + ReLU	2000
	Linear	$T \times T \times 3 \times \binom{D+3}{D}$
Postprocessing	Bilinear upsampling	$256 \times 256 \times 3 \times \binom{D+3}{D}$
	Polynomial transformation	$256 \times 256 \times 3$

- improvement in recognizability of the unfiltered content, measured by a network trained to classify the image content.

4.1 *Error Metrics*

Four different objective metrics are used to evaluate the quality of the recovered images. The first two are the simplest and most widely used full-reference quality metrics: the former is the MSE. Given an image I and its recovered version K, MSE is defined as follows:

$$\mathrm{MSE} = \frac{1}{m\,n}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[I(i,j)-K(i,j)]^2. \tag{4}$$

This metric is used since it is the one used as loss function in the training of our CNN. The latter is the peak signal-to-noise ratio (PSNR) and is related to MSE:

$$\mathrm{PSNR} = 20\cdot\log_{10}\left(\frac{\mathrm{MAX}_I}{\sqrt{\mathrm{MSE}}}\right), \tag{5}$$

where MAX_I is the maximum possible pixel value of the image.

The third error metric considered is the structural similarity (SSIM) index.[23] It exploits the assumption that human visual perception is highly adapted for extracting structural information from a scene and is an alternative complementary framework for quality assessment based on the degradation of structural information. In particular, it is used here to assess if the filter removal process degrades the structural information present in the original image. The general form of SSIM combines information from the luminance $l(I,K)$, the contrast $c(I,K)$, and structure $s(I,K)$ as follows:

$$\mathrm{SSIM}(I,K) = l(I,K)^{\alpha}\cdot c(I,K)^{\beta}\cdot s(I,K)^{\gamma}, \tag{6}$$

where $\alpha>0$, $\beta>0$, and $\gamma>0$ are parameters used to adjust the relative importance of the three components (we used the values $\alpha=\beta=\gamma=1$).

The fourth error metric considered is the spatial extension of CIELAB (S-CIELAB) that has been specifically designed for measuring reproduction errors of color images.[24]

Polynomial degree D	Number of transformations $T\times T$					
	32×32	16×16	8×8	4×4	2×2	1×1
$D=1$	30.70	30.57	30.47	29.47	27.83	27.53
$D=2$	31.60	31.49	31.41	30.43	28.64	28.33
$D=3$	32.01	31.99	31.92	30.83	28.86	28.50

(a) PSNR (higher is better)

Polynomial degree D	Number of transformations $T\times T$					
	32×32	16×16	8×8	4×4	2×2	1×1
$D=1$	0.9343	0.9332	0.9327	0.9288	0.9141	0.9123
$D=2$	0.9403	0.9400	0.9394	0.9361	0.9203	0.9210
$D=3$	0.9435	0.9435	0.9431	0.9395	0.9252	0.9257

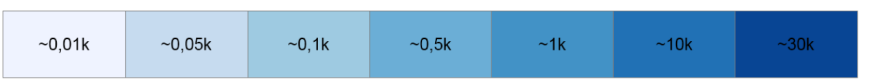
(b) SSIM (higher is better)

Polynomial degree D	Number of transformations $T\times T$					
	32×32	16×16	8×8	4×4	2×2	1×1
$D=1$	78.94	81.15	82.03	100.19	155.86	178.72
$D=2$	66.13	67.47	68.58	83.67	135.56	148.31
$D=3$	61.61	61.34	62.37	78.08	129.43	141.85

(c) MSE (lower is better)

Polynomial degree D	Number of transformations $T\times T$					
	32×32	16×16	8×8	4×4	2×2	1×1
$D=1$	5.14	5.33	5.22	5.74	5.93	6.14
$D=2$	4.71	4.82	4.80	4.98	5.72	5.79
$D=3$	4.72	4.75	4.71	5.11	5.64	5.66

(d) S-CIELAB (lower is better)



(e)

Fig. 3 Comparison of the performance of the proposed method varying the degree of the polynomial and the number of transformations used. Results are assessed using the four error metrics considered: (a) PSNR, (b) SSIM, (c) MSE, and (d) S-CIELAB. The background is color-coded with the legend reported in panel (e), to represent the total number of parameters used by the transformations.

Fig. 4 Example of removal of a spatially varying filter (Amaro) with a different number of polynomial transformations. As the number of transformation decreases, the quality of the restoration gets worse.

4.2 *Performance Varying the Degree and Number of Polynomials*

In this section, we show the performance we obtained in terms of the four error metrics considered in the previous section. Our method depends on two parameters: the degree D of the polynomial transformations and their number T^2. Figure 3 reports the results obtained by changing the degree of the polynomial used for the recovery, as well as the

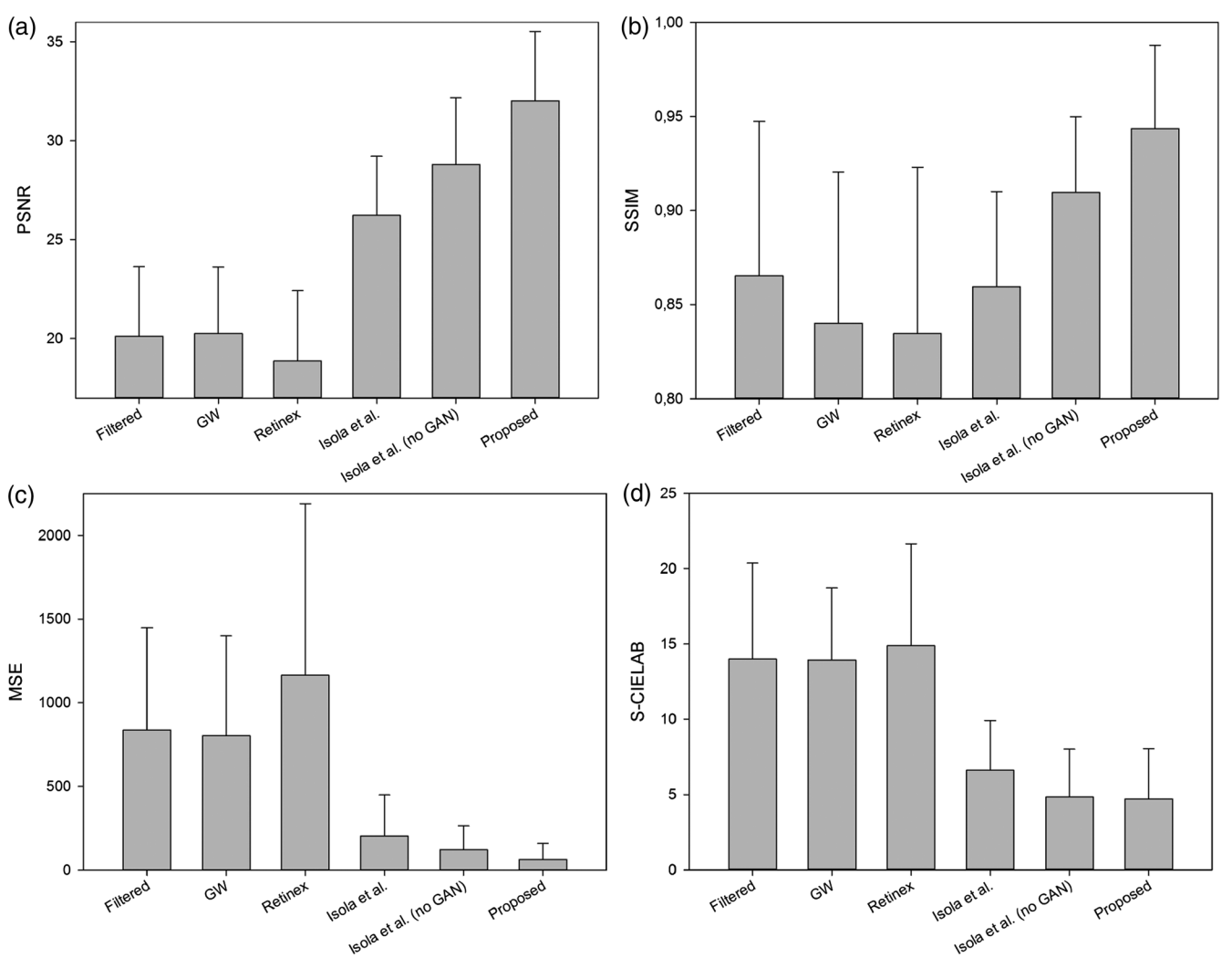


Fig. 5 Comparison of the proposed method with the state-of-the-art for the four error metrics considered: (a) PSNR, (b) SSIM, (c) MSE, and (d) S-CIELAB. For the former two metrics (reported in the top row) the higher the better, for the latter two (reported in the bottom row), the lower the better.

Fig. 6 Examples of recovery by our proposed method, GW, retinex, and by two variants of the method by Isola et al.[14]

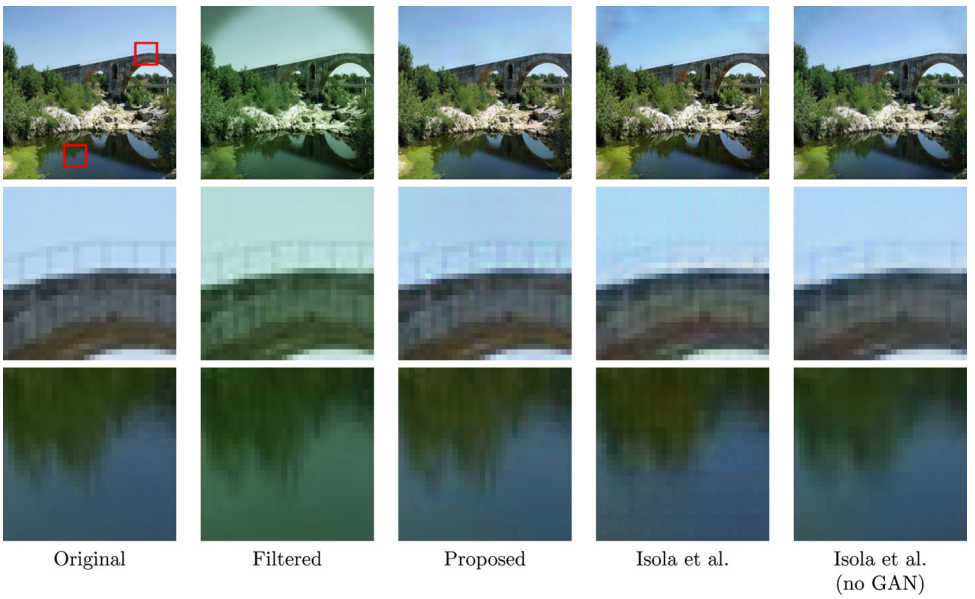


Fig. 7 Qualitative comparison of the details produced by the proposed method and by the method by Isola et al.,[14] with and without the GAN term in the loss function.

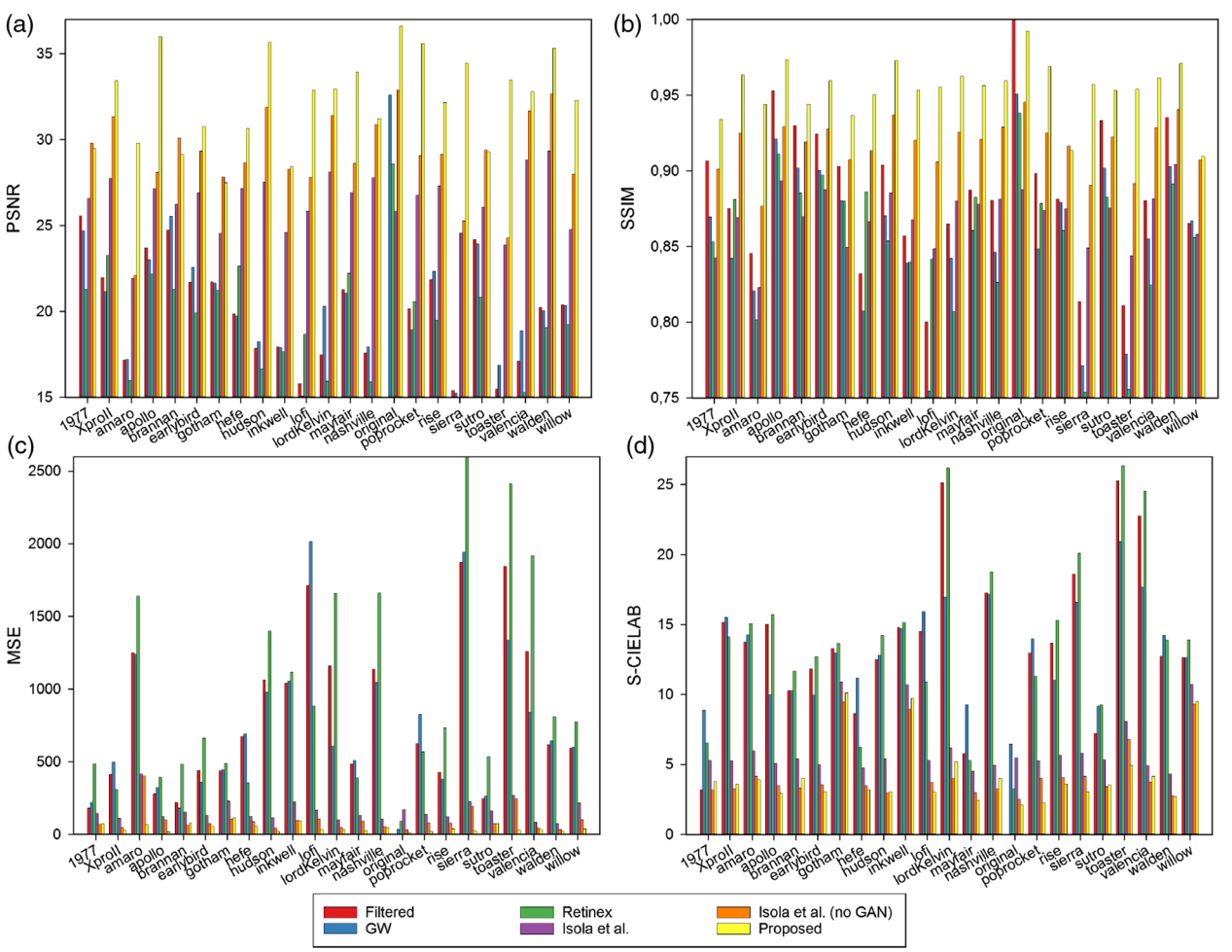


Fig. 8 Detailed comparison of the proposed method with the state-of-the-art in removing each of the 23 filters considered. Results are reported using the four error metrics considered: (a and b) for those in the top row, the higher the better and (b) for those in the bottom row, the lower the better.

Table 3 Confusion matrix summarizing the output of the filter classification network proposed by Bianco et al.,[2] obtained on the test images transformed by photographic filters. For each filter, the most common prediction is reported in bold.

	Original	1977	Amaro	Apollo	Brannan	Earlybird	Gotham	Hefe	Hudson	Inkwell	Lofi	Lord-K.	Mayfair	Nashville	Poprocket	Rise	Sierra	Sutro	Toaster	Valencia	Walden	Willow	X-pro II
Orig	**91.5**	0.6	0.1	0.2	0.3	0.3	0.0	1.9	1.3	0.1	0.2	—	1.2	0.1	0.0	0.9	—	0.4	—	0.0	0.2	0.1	0.3
1977	0.1	**99.2**	—	—	0.2	—	—	—	0.2	—	—	—	—	0.1	—	—	—	0.1	—	0.1	0.2	—	—
Amar	0.0	—	**99.9**	0.0	—	—	—	—	—	—	—	—	—	—	—	—	0.0	—	—	—	—	—	—
Apol	0.5	—	0.0	**99.2**	—	—	—	0.0	0.0	—	0.0	—	—	—	—	0.2	—	—	—	—	—	—	—
Bran	0.1	0.3	—	—	**99.0**	—	0.0	—	0.1	—	—	0.0	0.0	0.0	—	—	—	0.1	—	0.1	0.1	—	0.1
Earl	0.3	0.1	—	0.1	0.1	**99.0**	—	0.1	—	—	0.1	—	—	—	—	0.1	0.1	—	—	0.0	0.1	—	—
Goth	—	—	—	—	—	—	**100.0**	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Hefe	0.8	—	—	0.1	—	—	—	**98.6**	—	0.1	0.2	—	0.1	—	0.0	0.1	—	0.0	—	0.0	—	—	—
Huds	0.7	0.2	0.0	—	0.2	—	—	—	**98.4**	—	—	—	—	0.1	0.1	0.0	0.0	0.0	—	0.1	0.2	—	0.1
Inkw	0.0	—	—	—	—	—	—	0.0	—	**97.9**	—	—	—	—	—	—	—	—	—	—	—	2.1	—
Lofi	0.1	—	—	0.0	—	—	—	0.0	—	—	**99.9**	—	—	—	0.0	—	—	—	—	0.0	—	—	—
Lord	—	—	—	—	—	—	—	—	—	—	—	**99.9**	—	0.1	—	—	0.0	—	0.0	0.0	—	—	—
Mayf	0.9	—	—	0.0	—	0.1	—	0.1	0.1	0.1	0.0	—	**98.5**	—	0.0	0.1	—	—	—	0.0	—	0.0	0.1
Nash	—	—	—	—	—	—	—	—	—	—	—	0.1	—	**99.8**	—	—	—	—	—	0.0	0.1	—	0.1
Popr	0.0	—	—	—	—	—	—	—	0.1	—	—	—	—	—	**99.9**	0.0	—	—	—	—	0.0	—	—
Rise	0.6	—	—	0.5	0.1	0.1	—	—	0.1	—	—	—	0.1	—	0.0	**98.3**	—	0.1	—	0.0	—	—	—
Sier	—	—	—	—	—	—	—	—	0.0	—	—	—	—	—	—	—	**100.0**	—	—	0.0	—	—	—
Sutr	0.1	0.1	—	—	0.1	—	—	0.0	0.0	—	0.1	—	—	0.0	—	—	—	**99.5**	—	0.1	0.1	—	—
Toas	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	**100.0**	—	—	—	—
Vale	0.0	—	—	—	—	—	—	—	—	—	—	—	—	0.1	—	—	—	—	—	**99.9**	—	—	—
Wald	—	0.1	—	—	0.1	—	—	—	0.1	—	—	0.1	—	0.5	—	—	—	0.1	—	0.0	**99.1**	—	0.1
Will	0.1	—	—	—	—	—	—	—	—	1.9	—	—	—	—	—	—	—	—	—	—	—	**98.0**	—
X-pr	0.0	—	—	0.0	0.1	—	—	—	—	—	—	—	—	0.1	—	—	—	—	—	0.0	0.1	—	**99.6**

Table 4 Confusion matrix summarizing the output of the filter classification network proposed by Bianco et al.,[2] obtained on the test images restored by the proposed method. For each filter, the most common prediction is reported in bold.

	Original	1977	Amaro	Apollo	Brannan	Earlybird	Gotham	Hefe	Hudson	Inkwell	Lofi	Lord-K.	Mayfair	Nashville	Poprocket	Rise	Sierra	Sutro	Toaster	Valencia	Walden	Willow	X-pro II
Orig	**96.0**	0.2	0.1	0.3	—	0.4	—	1.2	0.1	—	0.1	—	0.9	—	—	0.4	—	0.1	—	0.0	0.0	—	0.1
1977	**81.4**	2.4	0.0	0.2	0.3	0.1	—	0.1	10.7	—	0.5	—	0.2	—	0.0	1.7	0.0	2.2	—	0.0	0.1	—	0.1
Amar	**90.8**	0.6	0.2	0.2	0.2	0.4	—	2.2	2.2	—	0.3	—	1.1	—	0.0	0.6	0.0	0.7	—	—	0.2	—	0.1
Apol	**90.9**	0.7	0.1	1.8	0.2	0.6	—	1.3	1.5	—	0.1	—	0.9	0.1	—	1.2	—	0.3	—	0.0	0.1	—	0.2
Bran	**88.3**	1.0	0.1	0.2	1.0	0.2	0.0	1.2	2.4	—	0.6	—	0.7	—	—	1.3	—	2.8	—	0.0	0.0	—	0.3
Earl	**89.1**	0.8	—	0.4	1.0	1.4	0.0	1.9	0.9	—	0.2	—	0.9	0.0	0.0	1.3	—	1.6	—	0.0	0.0	—	0.2
Goth	**88.7**	1.3	0.0	0.7	0.1	0.4	0.1	1.1	0.7	0.0	0.1	—	0.7	—	—	5.4	—	0.8	—	—	—	0.0	—
Hefe	**88.3**	0.4	0.0	0.2	0.4	0.1	0.1	4.4	1.4	—	0.4	—	1.0	—	—	2.2	—	0.9	—	—	0.1	—	0.1
Huds	**91.0**	0.6	0.0	0.2	0.2	0.6	0.0	1.7	1.2	—	0.5	0.0	0.9	0.0	—	0.9	—	1.8	—	0.0	0.1	—	0.2
Inkw	**88.4**	0.7	—	0.5	0.1	0.9	0.1	0.6	0.0	0.1	0.2	—	0.8	—	—	6.4	—	1.0	—	—	—	0.1	—
Lofi	**91.1**	0.8	—	0.2	0.3	0.1	0.0	1.4	2.0	—	0.9	—	1.0	0.1	—	1.1	—	0.5	—	—	0.2	—	0.2
Lord	**91.0**	0.6	0.1	2.3	0.1	0.4	—	1.7	0.6	—	0.1	—	1.4	—	0.0	0.7	—	0.8	0.0	—	—	—	0.2
Mayf	**89.8**	0.7	0.0	0.2	0.2	0.2	0.1	0.9	1.2	—	0.4	—	3.6	0.0	0.1	1.4	—	0.7	—	0.1	0.2	0.0	0.2
Nash	**88.8**	1.8	—	0.3	0.7	0.8	—	1.8	0.3	—	0.2	—	0.9	—	—	0.8	0.1	2.6	—	—	0.5	—	0.5
Popr	**90.2**	0.8	0.1	0.4	0.4	0.5	0.0	1.8	1.8	—	0.3	—	1.1	—	0.3	1.0	—	0.7	—	0.0	0.2	—	0.3
Rise	**91.1**	0.8	0.1	0.7	0.2	0.2	—	1.3	1.5	—	0.2	—	1.1	0.0	0.0	2.2	—	0.5	—	—	0.1	—	0.1
Sier	**91.5**	1.0	0.1	0.3	0.6	0.4	—	1.1	2.6	—	0.3	0.0	0.8	0.0	0.0	0.7	0.1	0.2	—	—	0.2	—	0.2
Sutr	**83.0**	1.1	0.1	0.4	0.6	0.2	0.1	2.3	1.3	—	0.5	—	1.2	—	—	2.5	—	6.8	—	—	0.0	—	0.1
Toas	**91.1**	0.7	0.1	0.2	0.6	1.2	—	0.7	2.8	—	0.4	—	0.7	0.1	0.0	0.8	—	0.8	—	—	0.0	—	—
Vale	**88.6**	1.2	—	1.8	0.2	0.1	0.1	1.7	1.4	—	0.2	0.0	0.9	0.1	0.0	2.0	—	1.0	—	0.0	0.7	—	0.1
Wald	**92.2**	0.8	0.1	0.4	0.4	0.5	—	1.4	0.9	—	0.2	—	1.8	—	—	0.5	0.0	0.2	—	0.0	0.5	—	0.2
Will	**86.2**	0.9	—	0.2	0.1	2.8	0.0	1.5	0.2	0.0	0.1	—	0.3	—	—	4.5	—	3.0	—	—	—	0.0	—
X—pr	**89.8**	0.5	0.0	0.3	0.4	0.1	—	1.0	0.9	—	0.4	—	1.0	0.0	—	2.8	—	1.6	—	0.0	0.2	—	0.9

number of transformations considered. The number of transformations we considered are $32 \times 32 = 1024$, $16 \times 16 = 256$, $8 \times 8 = 46$, $4 \times 4 = 16$, $2 \times 2 = 4$, and $1 \times 1 = 1$ (i.e., a global transformation of the whole image). From the results, it can be noticed that the performance obtained with 8×8 transformations or more is almost equivalent and that they start to decrease when fewer transformations are used. This behavior is consistent across all four error metrics considered. A visual example is reported in Fig. 4, where the results obtained using the different number of third degree polynomial transformations considered are reported. Fixing the number of transformations, it is possible to notice how the results improve for all the four error metrics considered at the increase of the polynomial degree. The same conclusion can be drawn also by fixing the number of parameters required, i.e., looking at the results in Fig. 3 taking into account the color coded background. Therefore, for the next experiments, we considered only the configuration of the proposed method with 32×32 polynomial transformations of third degree.

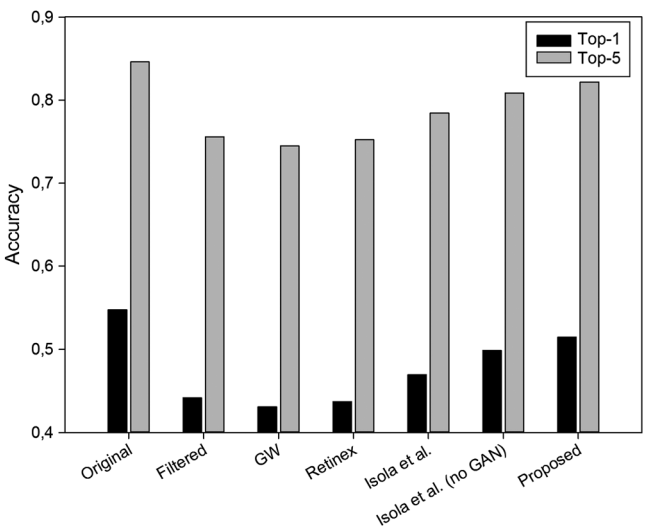


Fig. 9 Classification accuracy of a CNN trained to recognize the semantic labels in the Places-205 dataset.

4.3 *Comparison with the State-of-the-Art*

In this section, we compare the results of the proposed method with those of other algorithms from the state-of-the-art. The first one is the gray world (GW),[25] which is a global color correction algorithm. It is based on the assumption that the average reflectance in a natural scene is gray, and it balances the image channels in order to make their average value match. The second one is the retinex algorithm,[26] which is able to deal with nonuniform illumination by assuming that an abrupt change in chromaticity is caused by a change in reflectance properties. The third method is the image-to-image translation architecture recently proposed by Isola et al.[14] that we used in two variants: the former obtained with the default setup recommended by the authors and the latter obtained by disabling the GAN loss during training so that the method becomes a regression with a per-pixel L_1 loss. The comparison is reported in Fig. 5 in terms of all the four error metrics considered. We can see from the plots that our method significantly outperforms the compared methods.

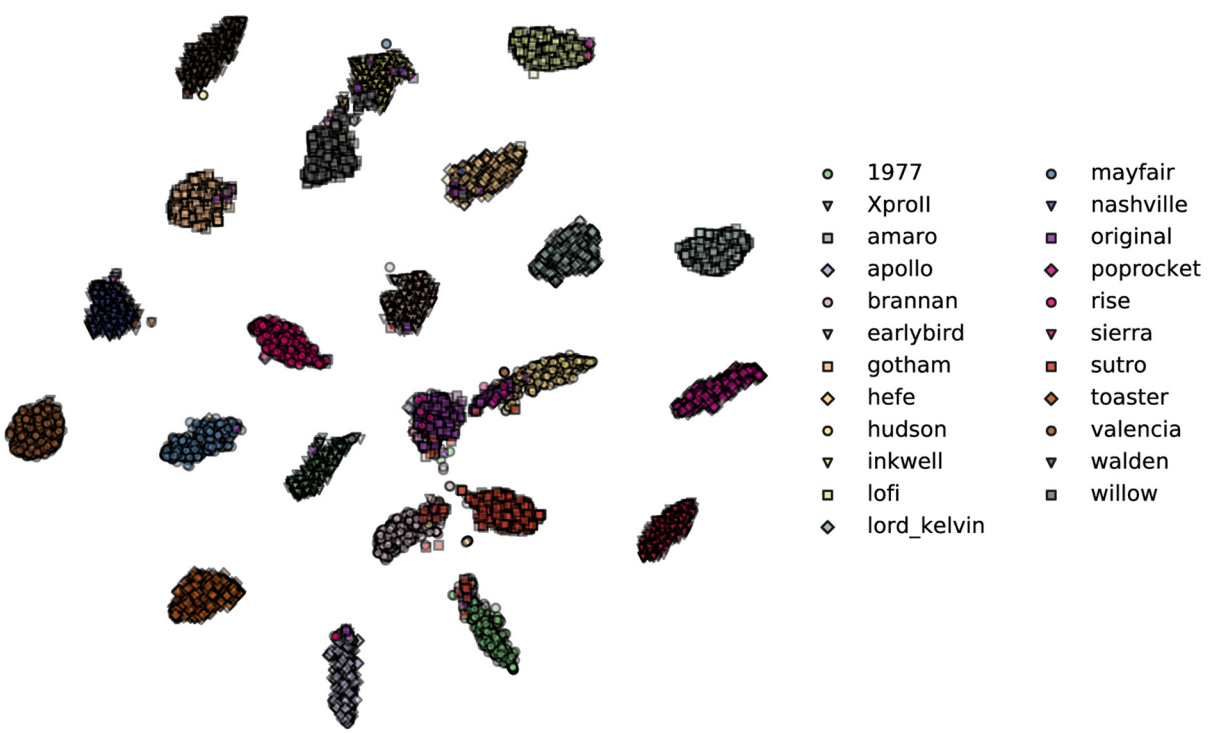


Fig. 10 Output of the t-SNE data visualization method applied to a set of 10,000 feature vectors computed by the second fully connected layer. The projected data points are displayed according to the corresponding photographic filter, even though this information was not available to t-SNE.

Some visual examples of images recovered with all the compared methods are reported in Fig. 6. Figure 7 shows a couple of enlargements of portions of the same image recovered with our method compared with the two variants of Isola et al.,[14] from which it is possible to notice how our method is able to preserve also the finest details.

As a further analysis, we report in Fig. 8 the breakdown of the results with respect to the different filters considered. From the results, it is possible to see that the performance of the different methods tested is consistent across the filters considered with very few ranking inversions. Focusing on the proposed method, we can notice that the worst results across the four error metric considered are obtained on those filters characterized by a heavy information loss, i.e., Gotham, Inkwell, and Willow, that are all black and white filters. This is particularly evident looking at the plot of the S-CIELAB metric, where it can be seen that the error on these three filters is almost the double of the others. Nevertheless, this error is lower than that of the filtered image, which means that the proposed method also performs a form of colorization.

4.4 *Filter Classification*

The filter removal procedure described here produces an output image that most of the times look more natural than its filtered version. To partially quantify this aspect, we measured how often this happens from the point-of-view of a convolutional network trained to recognize the application of photographic filters. More in detail, we considered the network that Bianco et al.[2] designed to recognize the same 22 photographic filters used in this work. The accuracy of that network was very high (about 98.9%), as it can be seen in the confusion matrix summarizing the results of classification for the test images (Table 3). All the 22 filters are correctly recognized in at least 97% of the times and original images are recognized as such in 91.5% of the cases.

We assessed the same network on the test images after removing the filters with the proposed method. The results, reported as confusion matrix in Table 4, are very different. The classification accuracy dropped to about 5.39%, which is slightly better than random guessing. In the great majority of cases, the network classified the recovered images as original, i.e., without any photographic filter applied. This happens, on average, 89.4% of the time. It seems that the recovery process is very good in canceling out those characteristic properties that make the filters recognizable for the neural network.

4.5 *Semantic Classification on Places-205 Dataset*

As a further experiment, we assessed the classification performance of a CNN trained to identify the semantic classes defined in the Places-205 dataset. The CNN architecture used is AlexNet,[27] and the trained model can be downloaded on the website of Places-205. The classification accuracy is measured on the original images (on which no filter was applied), on the filtered images, on the images corrected with the other methods included in the evaluation, and on those corrected with the proposed method. From Fig. 9, it is possible to see that, as expected, the best classification result is obtained on the original images. When the filters are applied, classification accuracy drops by almost 10% for the top-1 result. Correcting the images with GW and retinex does not improve this result, but it actually lowers the accuracy by almost a further 1%. The proposed method, instead, is able to reduce the gap with respect to the results obtained on the original images, increasing the accuracy by almost 7% with respect to the results obtained on the filtered images.

4.6 *Analysis of the Learned Features*

Finally, in order to better understand the behavior of the network, we conducted an analysis on the features computed by the second fully connected linear layer. More in detail, we randomly sampled 10,000 images from the test set and used the t-distributed stochastic neighbor embedding (t-SNE) method[28] to project the 2000-dimensional feature vectors produced by that layer onto a plane. The projection computed by t-SNE preserves the similarity among the feature vectors, and it is determined in a completely unsupervised way, without any knowledge about the photographic filter applied to the image. From the results depicted in Fig. 10, it is clear how the method identified 23 clusters, one for each photographic filter. This fact suggests that the main purpose of the second fully connected layer is to recognize which filter has been applied to the input image, information that allows effectively selecting the restoration strategy that is implemented in the following layer.

5 Conclusions

In this paper, we addressed the problem of restoring the appearance of images that have been processed by photographic filters. Our method corrects the images by applying a set of polynomial transformations, whose parameters are found by a CNN especially designed for this task. The source code of our method is available at the following link: https://github.com/dros1986/filter_removal.

To assess the effectiveness of the method, we processed a subset of the Places-205 dataset with 22 different photographic filters. The quality of the reconstructions we obtained, measured with several objective quality measures, clearly outperformed that of the other algorithms included in the evaluation. Moreover, we also demonstrated how, with respect to the filtered images, the images recovered by our method were significantly easier to recognize for image recognition systems in the state-of-the-art.

As a future work, we plan to experiment with a larger and more diverse set of photographic filters and with other restoration problems. At the present time, our method cannot deal with certain kinds of degradation (e.g., noise or blur) that were not particularly relevant for the filters we considered here. In order to address them, a possible enhancement would be the replacement or the integration of the polynomial transformations with other parametric image operators such as, for instance, linear convolutions.

References

1. Y.-H. Chen et al., "Filter-invariant image classification on social media photos," in *Proc. of the 23rd Annual ACM Conf. on Multimedia Conf.*, pp. 855–858, ACM (2015).
2. S. Bianco, C. Cusano, and R. Schettini, "Artistic photo filtering recognition using CNNs," in *Proc. of 6th Int. Workshop on Computational Color Imaging*, pp. 249–258 (2017).
3. Z. Lou et al., "Color constancy by deep learning," in *Proc. of the British Machine Vision Conf.*, pp. 1–12 (2015).

4. S. Bianco, C. Cusano, and R. Schettini, "Color constancy using CNNs," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition Workshops*, pp. 81–89 (2015).
5. S. Bianco, C. Cusano, and R. Schettini, "Single and multiple illuminant estimation using convolutional neural networks," *IEEE Trans. Image Process.* **26**(9), 4347–4362 (2017).
6. W. Shi, C. C. Loy, and X. Tang, "Deep specialized network for illuminant estimation," in *European Conf. on Computer Vision*, pp. 371–387 (2016).
7. S. W. Oh and S. J. Kim, "Approaching the computational color constancy as a classification problem through deep learning," *Pattern Recognit.* **61**, 405–416 (2017).
8. R. Zhang, P. Isola, and A. A. Efros, "Colorful image colorization," in *European Conf. on Computer Vision*, pp. 649–666 (2016).
9. G. Larsson, M. Maire, and G. Shakhnarovich, "Learning representations for automatic colorization," in *European Conf. on Computer Vision*, pp. 577–593 (2016).
10. S. Iizuka, E. Simo-Serra, and H. Ishikawa, "Let there be color!: joint end-to-end learning of global and local image priors for automatic image colorization with simultaneous classification," *ACM Trans. Graph.* **35**(4), 1–11 (2016).
11. S. Liu, J. Pan, and M.-H. Yang, "Learning recursive filters for low-level vision via a hybrid neural network," in *European Conf. on Computer Vision*, pp. 560–576 (2016).
12. R. Gao and K. Grauman, "On-demand learning for deep image restoration," in *Proc. IEEE Conf. Comput. Vision and Pattern Recognition* pp. 1086–1095 (2017).
13. I. Goodfellow et al., "Generative adversarial nets," in *Advances in Neural Information Processing Systems*, pp. 2672–2680 (2014).
14. P. Isola et al., "Image-to-image translation with conditional adversarial networks," in *the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)* (2017).
15. J.-Y. Zhu et al., "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *the IEEE Int. Conf. on Computer Vision (ICCV)* (2017).
16. A. Ignatov et al., "DSLR-quality photos on mobile devices with deep convolutional networks," in *the IEEE Int. Conf. on Computer Vision (ICCV)* (2017).
17. B. Zhou et al., "Learning deep features for scene recognition using places database," in *Advances in Neural Information Processing Systems*, pp. 487–495 (2014).
18. G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science* **313**(5786), 504–507 (2006).
19. O. Ronneberger, P. Fischer, and T. Brox, "U-net: convolutional networks for biomedical image segmentation," in *Int. Conf. on Medical Image Computing and Computer-Assisted Intervention*, pp. 234–241 (2015).
20. H. R. Kang, *Color Technology for Electronic Imaging Devices*, SPIE Press, Bellingham, Washington (1997).
21. S. Bianco et al., "Polynomial modeling and optimization for colorimetric characterization of scanners," *J. Electron. Imaging* **17**(4), 043002 (2008).
22. S. Bianco and R. Schettini, "Error-tolerant color rendering for digital cameras," *J. Math. Imaging Vision* **50**(3), 235–245 (2014).
23. Z. Wang et al., "Image quality assessment: from error visibility to structural similarity," *IEEE Trans. Image Process.* **13**(4), 600–612 (2004).
24. X. Zhang and B. A. Wandell, "A spatial extension of CIELAB for digital color-image reproduction," *J. Soc. Inf. Disp.* **5**(1), 61–63 (1997).
25. G. Buchsbaum, "A spatial processor model for object colour perception," *J. Franklin Inst.* **310**(1), 1–26 (1980).
26. E. H. Land and J. J. McCann, "Lightness and retinex theory," *J. Opt. Soc. Am. A* **61**(1), 1–11 (1971).
27. A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, pp. 1097–1105 (2012).
28. L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.* **9**, 2579–2605 (2008).

Simone Bianco obtained his PhD in computer science at DISCo (Dipartimento di Informatica, Sistemistica e Comunicazione) of the University of Milano-Bicocca, Italy, in 2010. He obtained his BSc and MSc degrees in mathematics from the University of Milano-Bicocca, Italy, respectively, in 2003 and 2006. He is currently an assistant professor and his research interests include computer vision, machine learning, optimization algorithms, and color imaging.

Claudio Cusano was graduated in computer science in 2002 at the University of Milano-Bicocca, where he also obtained his PhD in computer science in 2006. Since 2002, he has been a fellow at the Multimedia Information Technologies Institute of the Italian National Council of Research, where he started its research activity as a researcher with grant. In 2006, he became a postdoc researcher at the Imaging and Vision Laboratory of the University of Milano-Bicocca. Since 2012, he served as assistant professor in computer engineering at the University of Pavia, where he became an associate professor in 2015. His topics of interest are focused on automatic image analysis and recognition, and include scene classification, texture analysis, color processing, face recognition, and 3-D imaging.

Flavio Piccoli obtained his BSc and MSc degrees in computer science from the University of Milano-Bicocca, Italy, respectively, in 2010 and 2014. He worked on object detection during his internship in STMicroelectronics. Currently, he is a PhD candidate at the University of Milano-Bicocca. His research interests are in computer vision and machine learning with a focus on anomaly detection for industrial quality inspection.

Raimondo Schettini is a professor at the University of Milano Bicocca (Italy). He is a vice director of the Department of Informatics, Systems, and Communication, and head of the Imaging and Vision Lab. He has been associated with the Italian National Research Council since 1987, where he led the color imaging lab from 1990 to 2002. He has been a team leader in several research projects and published more than 300 refereed papers and six patents about color reproduction, and image processing, analysis, and classification. He is a fellow of the International Association of Pattern Recognition for his contributions to pattern recognition research and color image analysis.