Quasi-Unsupervised Color Constancy

Simone Bianco
University of Milano-Bicocca
simone.bianco@unimib.it

Claudio Cusano
University of Pavia
claudio.cusano@unipv.it

Abstract

We present here a method for computational color constancy in which a deep convolutional neural network is trained to detect achromatic pixels in color images after they have been converted to grayscale. The method does not require any information about the illuminant in the scene and relies on the weak assumption, fulfilled by almost all images available on the Web, that training images have been approximately balanced. Because of this requirement we define our method as quasi-unsupervised*. After training, unbalanced images can be processed thanks to the preliminary conversion to grayscale of the input to the neural network. The results of an extensive experimentation demonstrate that the proposed method is able to outperform the other unsupervised methods in the state of the art being, at the same time, flexible enough to be supervisedly fine-tuned to reach performance comparable with those of the best supervised methods.*

1. Introduction

Computational color constancy is a longstanding problem which consists in correcting images to make them appear as if they were taken under a neutral illuminant. Computational color constancy can benefit the solution to many computer vision problems such as visual recognition [14], surveillance [22], etc., where color is an important feature for distinguishing objects. Despite its apparent simplicity, this problem is very challenging for both human and computer vision systems [25, 20].

The last decade has witnessed a remarkable improvement in our capability of addressing many computer vision problems. The main factor behind this is the development of deep learning algorithms that make it possible to follow a very effective data-driven approach [35]. It is not surprising therefore that several attempts have been made to exploit this machine learning paradigm for computational color constancy as well [7, 37, 28, 38]. In our opinion, however, these approaches only partially took advantage of the potential of deep learning.

The main difficulty in applying deep learning methods to color constancy consists in the lack of large data sets annotated with ground truth illuminants. In fact, data sets used for this purpose are usually obtained by taking pictures of scenes where a standard object of known chromatic properties (e.g. a *color target*) has been placed. This procedure is clearly impractical for the collection of the large data sets needed by supervised deep learning. Another issue with color constancy methods based on machine learning is that the learned models are often specialized for images acquired with the same devices used to collect the training set. Their application to images taken with other devices requires some form of adaptation or retraining [2].

We present here a method for computational color constancy that is based on a deep convolutional neural network[1]. The method leverages large datasets of publicly available images to train the network in a *quasi-unsupervised* setup. No ground truth about the color of the illuminant is needed. Instead, the method exploits the assumption that training images have been approximately balanced either manually or by unspecified automatic processing pipelines. Because of this assumption (that, as we will see, is easily fulfilled in practice) we define our method as "quasi-unsupervised", instead of just "unsupervised".

More in detail, the neural network is trained to detect achromatic pixels. To do so, only a grayscale version of the input image is taken into account. This way, the output is independent on the actual color of the illuminant and, therefore, the network can be later applied to both balanced and unbalanced images. The weighted average of the detected pixels is the estimate of the illuminant that is finally used to correct the input color image.

We verified the feasibility of the approach by training several neural networks on three large datasets commonly used for image recognition and retrieval. The evaluation on two annotated datasets of raw images demonstrates that very accurate results can be obtained even if no images from those datasets were used for training.

The novel design of the proposed method determines re-

[1] Source code and trained models available at https://claudio-unipv.github.io/quasi-unsupervised-cc/

markable advantages over competing methods in the state of the art: (i) the method exploits a complex neural network architecture, without requiring a large training set of annotated images; (ii) the trained model can be applied to unbalanced images acquired with any camera without the need of any kind of adaptation. Despite the complexity of the setup, the accuracy in estimating the illuminant compares favorably with respect to those reported in the literature. In particular, the proposed method is able to outperform the other unsupervised methods in the state of the art. Moreover, it optionally supports a supervised fine tuning on specific datasets that makes it possible to reach performance comparable with those of the best supervised methods.

2. Related work

The computational color constancy methods in the state of the art are usually divided into two categories: statistics-based and learning-based. Methods in the former category make assumptions about the statistical properties of natural scenes, and estimate the color of the illuminant as the deviation from such assumptions [46]. Methods in the latter category estimate the color of the illuminant using a model learned from training data. The great majority of recent methods are learning-based, because this approach allows to reach a generally higher accuracy with respect to statistics-based methods. Many of these methods employ models trained on handcrafted features extracted from the input image, e.g [18, 11, 21, 41, 16], while the most recent works learn features by using deep convolutional neural networks, e.g. [7, 37, 8, 44, 28].

The main difficulty in applying the above deep learning methods consists in the lack of large scale data sets annotated with ground truth illuminants. As a reference, the largest data set available for color constancy is three orders of magnitude smaller than those available for other computer vision tasks such as visual class recognition [42]. Furthermore, such methods tend to degrade their performance when used in a cross-dataset setup, requiring a fine-tuning phase to adapt to the new dataset.

These reasons motivate the research of new algorithms that do not need a dataset with annotated illuminant ground truths, that we call unsupervised methods, and that produce results comparable to learning-based methods in the state of the art. To this end, we propose an alternative categorization of color constancy algorithms into three different classes: *parametric*, including methods that rely on a very small set of parameters to be tuned, e.g. [4, 19, 46]; *supervised*, including methods that need a proper training phase, e.g. [23, 12, 6]; *unsupervised*, including methods that do not need an annotated dataset and can be readily applied to new datasets without any form of adaptation, e.g. [34, 9].

In the following we review these latter works, which are the most relevant to the aim of this paper. We refer the readers to the surveys [26, 25] for additional background.

Interestingly, the first color constancy algorithms that have been proposed in the state of the art are unsupervised algorithms. For example the White Point (or MaxRGB) [34] algorithm assumes that the maximum values obtained independently from each of the three color channels represents the color of the illumination. Gray World [9] is based on the assumption that the average color in the image is gray and that the illuminant color can be estimated as the shift from gray of the averages in the image color channels. Recently Buzzelli et al. [10] proposed a deep learning method that is not trained with illuminant annotations, but with the objective of improving performance on an auxiliary task such as object recognition. The method therefore learns to predict illuminant color in absence of any illumination ground truth data, but it requires label information for the auxiliary task. Banic and Loncaric [3] proposed an heuristic named green stability assumption, that can be used to fine-tune the values of the parameters of the statistics-based methods by using only raw images without known ground-truth illumination. In [2] an unsupervised learning-based method is proposed that learns its parameter values after approximating the unknown ground-truth illumination of the training images. Therefore [2] and [3] do not need the illuminant ground-truth information to be available, but require a raw training dataset. The same data is also needed by Qian et al. [40] that proposed a statistical color constancy method that relies on a novel gray pixel detection followed by mean shift clustering.

3. Method

Computational color constancy is usually tackled in two steps: first the color of the illuminant is estimated, then the estimate is used to correct the input image. In this work we propose an illuminant estimation method based on a convolutional neural network trained on large datasets of pictures. The method is “quasi-unsupervised” since its training process does not rely on the knowledge of the actual color of the illuminant in the scene. The method, instead, is based on the assumption that the training images have been approximately balanced by their owners before publication. As a consequence of that, we expect that in most cases the color of the illuminant appears close to gray. We define the method as “quasi-unsupervised” (instead of just “unsupervised”) because the assumed preliminary color correction entails some form of weak supervision, even though such a correction has not been performed explicitly to achieve color constancy.

After the training, in order to make it possible to apply the resulting model to unbalanced raw images, two main issues need to be addressed: (i) these images will be of different kind with respect to those used for training, and (ii) an actual ground truth will be available for the evaluation,

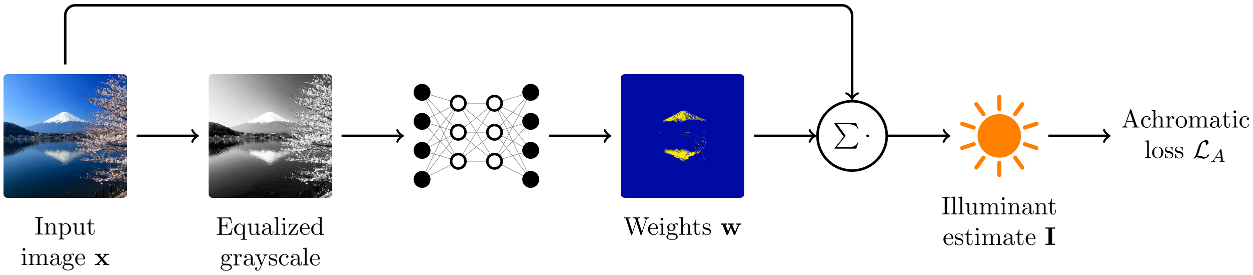

Figure 1. Schematic view of the proposed method. The input image $\mathbf{x}$ is first converted to grayscale and then fed to a convolutional neural network. Then, the color of the illuminant is estimated as the sum of the input RGB pixels, weighted by the output $\mathbf{w}$ of the network. During training the estimate is evaluated with an achromatic loss function.

but not for training. We solve the first issue by converting the images to grayscale before passing them to the network, so that they are almost independent on the color of the scene illuminant. The lack of a ground truth is addressed by training the network to solve a problem that can be considered a proxy of illuminant estimation: the detection of achromatic pixels. An overview of the method is given in Figure 1, details are explained in the following sections.

Once an estimate of the color of the illuminant has been computed, it can be used to correct the input image. To do so we apply the von Kries model [47], that consists in scaling the color components of the pixels by the corresponding components of the estimate.

3.1. Illuminant estimation

Many illuminant estimation methods are based on the simple fact that the color of the illuminant directly contributes to the color of the image pixels. These methods expect that by averaging all or some of the pixels the corresponding reflectances in the scene cancel out, leaving as a result the color of the light source. Examples of this strategy are the *Gray World* algorithm (that takes the average over the whole image), the *White Patch* algorithm (that averages a group of the brightest pixels), and the *White Point* algorithm (that takes just the pixel with the highest level of brightness).

Here we propose to train a convolutional neural network to select which pixels should be used to estimate the color of the illuminant. More precisely, the estimate will be a weighted sum of the input pixels where the weights are the output of the network. For a $H \times W$ input RGB image $\mathbf{x}$ ($\mathbf{x}_{ij} \in \mathbb{R}^3$) the network yields a weight map $\mathbf{w}$ ($w_{ij} \in [0,1]$) which is used to compute the scene illuminant $\mathbf{I}$ as:

$$\mathbf{I} = \frac{\sum_{i=1}^{H} \sum_{j=1}^{W} \mathbf{x}_{ij} w_{ij}}{Z}, \tag{1}$$

where Z is a factor that normalizes the vector to make it have unit Euclidean norm.

When fed with public images, taken from the web or from datasets used by the computer vision community, in accordance with the balancing assumption we want the network to produce an achromatic estimate. The divergence of the estimated color from the gray axis can then be used as a loss function to train the network. We call this loss the *achromatic loss* $\mathcal{L}_A(\mathbf{I})$, and we define it in terms of the cosine of the angle between the network estimate $\mathbf{I} = (I_r, I_g, I_b)^T$ and the gray axis:

$$\mathcal{L}_A(\mathbf{I}) = 1 - \frac{I_r + I_g + I_b}{\epsilon + \sqrt{3(I_r^2 + I_g^2 + I_b^2)}}, \tag{2}$$

where $\epsilon = 10^{-4}$ is a small quantity used to stabilize the ratio. It can be easily shown that the loss is non-negative, and that it is close to zero only when $I_r \simeq I_g \simeq I_b$.

The key idea is that by minimizing $\mathcal{L}_A(\mathbf{I})$ the network will learn to assign high weights to the pixels that are likely to be achromatic. Of course this task would be trivial if the network was able to see the input RGB image. To force the network to learn a more elaborate strategy we feed it, instead, with a gray-level version of the image. In fact, it has already been shown that is possible to train a network to predict colors from gray-level images (i.e. to perform an automatic colorization [49]). Here, through the achromatic loss, we implicitly train it to identify gray pixels.

The advantage of this approach is that what the network learns can be used to estimate the illuminant even when the balancing assumption does not hold. In particular it is possible to apply it to the unbalanced images that form the datasets commonly used to evaluate color constancy algorithms. For these images we expect that the network selects pixels that appear with the same color of the illuminant because they are part of matte gray objects in the scene, of highlights reflecting the light source, etc. In other words, by making use of information that is substantially independent on the color of the illuminant, we expect that the network will learn to make accurate estimates for both balanced and unbalanced images.

However, to make this possible care has to be taken to make the network work equally well for both the public balanced images used for training and the unbalanced images used for the evaluation. In fact, the two kind of images usually have a very different dynamic range, may have clipped values or not, etc. Moreover, public images are likely to be in the sRGB color space, while test images are in the raw format of the acquisition device. To make their graylevel versions comparable we compute them as the average of the RGB color channels (that, with respect to more sophisticated conversions, makes as less assumptions as possible), and we apply a histogram equalization. Equalization adaptively distorts the gray levels, reducing the differences between images taken with different devices, or processed by different pipelines.

Moreover, before their conversion to grayscale training images, that are assumed to be in the sRGB color space, are preliminary processed with a gamma removal to make their pixels' values linear with respect to energy:

$$c' = \begin{cases} c/12.92 & \text{if } c \leq 0.04045, \\ ((c+0.055)/1.055)^{2.4} & \text{otherwise,} \end{cases} \tag{3}$$

where c and c' represent one of the three color channels before and after the transformation [29].

A difficulty in minimizing $\mathcal{L}_A(\mathbf{I})$ is that, due to the normalization in (1), the estimate $\mathbf{I}$ is invariant under scalings of $\mathbf{w}$. A consequence of this invariance is that the neural network is not encouraged to use the whole $[0, 1]$ range for $\mathbf{w}$, since it can assign tiny weights to the pixels without changing the final estimate. This negatively affects the stability of the optimization algorithm. To push the network to use larger weights we introduced an additional noise term $\mathbf{n} \in \mathbb{R}^3$ in Equation (1):

$$\mathbf{I} = \frac{\mathbf{n} + \sum_{i=1}^{H} \sum_{j=1}^{W} \mathbf{x}_{ij} w_{ij}}{Z}, \tag{4}$$

where the three components of $\mathbf{n}$ are normally distributed with zero mean and σ^2 variance. The larger the variance, the larger the average weight assigned by the network to make the contribution of noise negligible. The noise term also acts as regularizer and is used only during training.

3.2. Extensions and variations

The proposed method is quite flexible, and can easily accommodate several variations. In particular, the grayscale image can be replaced by, or combined with other information, provided that it is independent on the color of the illuminant. We experimented with information derived from the spatial gradient computed on each color channel. Since the magnitude of the gradient is strongly correlated with the color of the illuminant, we consider only the direction. More precisely, for each color channel we compute the horizontal and the vertical spatial derivatives by applying the Sobel operators [45]. Then, the two derivatives are normalized to form a unit-length vector. This procedure yields a six-channel image (two derivatives times three channels) that can be used as input of the neural network.

3.3. Supervised fine tuning

Even though the main focus of this work is the quasi-unsupervised setting, it is possible to adapt the method to supervised learning as well. To do so it is sufficient to replace the achromatic loss in (2) with a *chromatic loss* $\mathcal{L}_C$, defined in terms of the cosine of the angle between the estimated illuminant $\mathbf{I}$ and the target illuminant $\hat{\mathbf{I}} = (\hat{I}_r, \hat{I}_g, \hat{I}_b)$:

$$\mathcal{L}_C(\mathbf{I}, \hat{\mathbf{I}}) = 1 - \frac{I_r \cdot \hat{I}_r + I_g \cdot \hat{I}_g + I_b \cdot \hat{I}_b}{\epsilon + \sqrt{(I_r^2 + I_g^2 + I_b^2)(\hat{I}_r^2 + \hat{I}_g^2 + \hat{I}_b^2)}}. \tag{5}$$

Supervised training of a deep learning model requires a large dataset annotated with a suitable ground truth. Since this is hard to achieve, we propose to follow a *fine tuning* procedure [36]. In this case, the parameters of the network are initialized by the quasi-unsupervised learning on a large dataset. Then the training process continues with supervised learning over a smaller annotated dataset with a small learning rate.

3.4. Neural network architecture

The architecture of the neural network used in this work is an adaptation of that proposed by Isola *et al.* [30] for image-to-image translation. We chose to take inspiration from that architecture because it demonstrated to be suitable for image colorization, a task that is somewhat related to the detection of achromatic pixels.

The network takes as input a 256×256 graylevel image (possibly augmented with gradient information) and produces as output a 256×256 weight map. The layers form a U-shaped encoder/decoder with skip connections. There are eight convolutions with kernel size 4×4 and stride 2 that are paired to eight deconvolutions (transposed convolutions) with the same kernel size and stride. All these operations (with exception of the first and the last) are followed by a batch normalization and other non-linearities (leaky ReLUs with slope 0.2 for convolutions and conventional ReLUs for deconvolutions). For the last deconvolution the ReLU is replaced by a sigmoid that yields the weights assigned to the input pixels. During training, the first three deconvolutional blocks include dropout with probability 0.5. The entire net include about 54 millions of learnable parameters. Preliminary tests shown that simplifying the architecture slightly reduces the final accuracy.

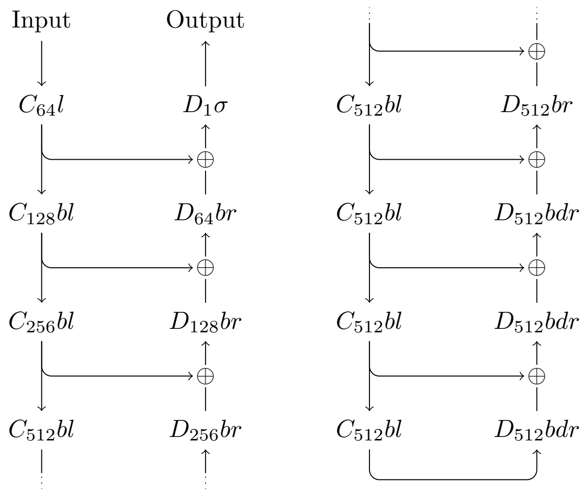

Figure 2. Structure of the neural network. In the diagram C_k denotes a convolution with k output channels. Similarly, D_k represents a deconvolution (transposed convolution). All convolutions and deconvolutions have a kernel size of 4×4 and stride 2. The other operations are denoted by: $l \rightarrow$ leaky ReLU, $r \rightarrow$ ReLU, $b \rightarrow$ batch normalization, $d \rightarrow$ dropout, $\sigma \rightarrow$ sigmoid, $\oplus \rightarrow$ concatenation along channels. Operations in the same group are executed from left to right.

4. Experimentation

The neural network has been trained by running 300 000 iterations of the Adam optimization algorithm [33]. The objective function was the achromatic loss, as defined in Equation (4), with the standard deviation of the noise term set to 100. Each iteration analyzed a mini-batch of 16 images; the learning rate was 10^{-4} and the coefficient for weight decay was 10^{-5}. All the parameters have been empirically set on the basis of a few preliminary experiments.

For deep learning applications the quality of training data is of paramount importance. In this work we decided to adopt three large datasets widely used to train image recognition and retrieval systems. The *Ilsvrc12* is the dataset that has been made publicly available for the Internet Large Scale Visual Recognition Challenge [42] which probably represents the most popular benchmark for image recognition. The dataset consists of approximately 1.2 millions samples taken from of 1000 different categories among those collected for the ImageNET initiative [17].

The second dataset is *Places365* [50] which includes about 1.8 millions images representing 365 different categories of scenes. Images were obtained by querying several search engines with terms taken from WordNet, and then manually annotated. The main purpose of the dataset is to serve as benchmark for scene recognition systems.

The last dataset we considered is the *Flickr100k* dataset [39]. It consists of 100 071 images collected from the Flickr photo sharing service by searching for the 146 most popular tags. The dataset has been collected to evaluate image retrieval algorithms.

We chose three diverse datasets with the aim of assessing how much the nature of the training images influences the quality of the learned model. Ilsvrc12 and Places365 consist of images taken from search engines, while Flickr100k includes images from a single source. Ilsvrc12 includes many "object centric" images with little or no background, while Places365 focuses on whole scenes. Images from Flickr100k seem, on average, of higher quality than those from the other two datasets.

Figure 3 reports some examples of images from the three datasets processed by the trained networks (images have been taken from the validation sets of Ilsvrc12 and Places365 and from the training set of Flickr100k). The strategy followed in selecting the pixels for the estimation of the illuminant can be inferred by looking at the weights. The network often selects light sources such as lamps, the sky or the sun. Windows are often selected in indoor scenes with the light coming from outside. The network seems also quite good in identifying highlights and surfaces diffusing the light directly from the source. It is quite common the case in which dark areas are selected: this is due to their limited impact on the sum in Equation (1).

Figure 3 also shows that not all the images are well balanced. Some of them present a strong non-neutral color cast which is very evident in the case of sunsets and nighttime images, and for some indoor images. However, the illuminant estimate provided by the method seems coherent with the content of the images. Even though we cannot quantitatively evaluate the estimates due to the lack of a ground truth, we can observe that the images balanced according to the estimates look pleasantly natural. This suggests that the network learned how to balance the outliers by modeling a large number of "almost balanced" images.

4.1. Evaluation

The aim of the proposed method is to achieve a high accuracy in estimating the color of the illuminant in unbalanced pictures. To assess this we processed two different datasets of raw images commonly used to evaluate color constancy algorithms. Both datasets contain high-resolution photographs, representing scenes including a color calibration target (the Macbeth ColorChecker). For each image a ground truth illuminant has been computed by analyzing the gray patches in the color target.

The first test dataset is the Color Checker (CC), in the variant reprocessed by Shi and Funt [23, 43]. It consists of 568 images acquired with a Canon 1D, and a Canon 5D cameras. The second dataset has been collected by a research group in the National University of Singapore (NUS) [15] and includes 1853 images acquired with 9 dif-

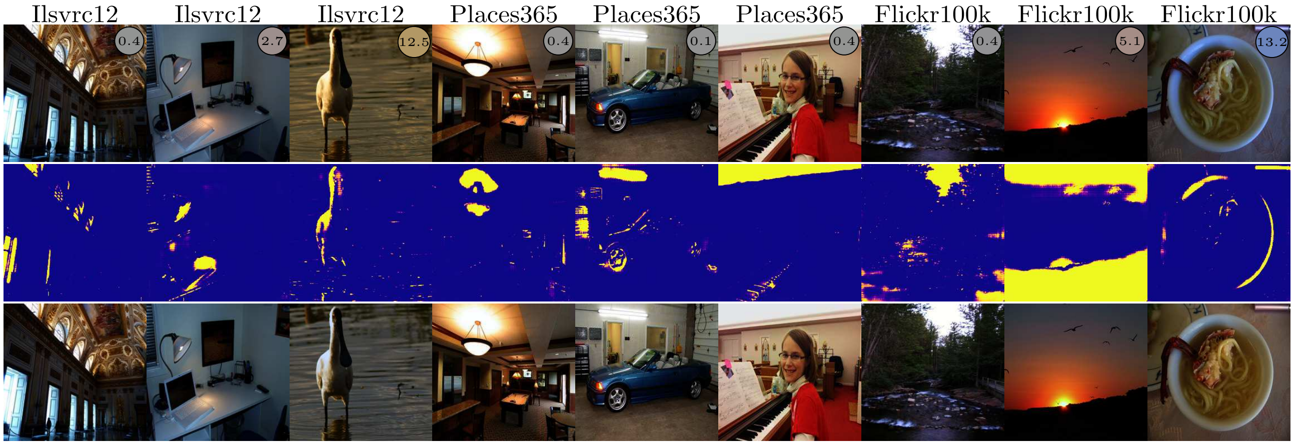

Figure 3. Examples of images from the training sets, processed by the trained networks. The top row shows the input images, each one with superimposed a circle representing the estimated color of the illuminant. Inside the circle is reported the angular difference (in degrees) between the estimate and the gray axis. The second row reports the weights assigned by the network to the pixels (blue → 0, yellow → 1). The third row reports the images balanced with the estimated illuminant.

ferent cameras. As suggested by Hordley and Finlayson the error metric we considered is the angle between the estimated and the ground truth illuminants [27].

The results we obtained by applying the trained models to the two test datasets are summarized in Table 1. The mean and median angular errors are quite uniform over the three training sets with differences of about 0.2 degrees or less. This is very important since it demonstrates that the type of photographs used for training is not of primary importance. It also suggests that our method is relying on assumptions (i.e. that training images have been balanced) that are easily met in practice.

For each combination of training and test datasets we evaluated three variants differing in the kind of data processed by the network. The first processes equalized grayscale images, the second analyzes gradient directions, and the third is based on a combination of the two. In all the cases using gradient directions, alone or in combination, allowed to obtain better results than using just the grayscale image. For the CC dataset the lowest median angular error has been obtained by the model trained on Ilsvrc12 using both grayscale and directions. For NUS the best combination in terms of median angular error was to use the model trained on Flickr100k using just the gradient directions. For the rest of the experiments we consider as reference version the one trained on Ilsvrc12 with grayscale and directions.

Figure 4 shows the result of processing some images from the test sets. It can be noticed how, even in the case of unbalanced images, the network selects meaningful regions such as those representing the light sources or highlights. Differently than with training images, this time the selected pixels are not achromatic. They appear, instead, approximately with the color of the ground truth illuminant. As a result, the images balanced according to the estimates appear as if they were taken under a neutral illuminant.

Training set	Test set	Input	Mean	Median	Max
Ilsvrc12	CC	Grayscale	4.04	2.67	27.88
	CC	Directions	3.67	2.53	**17.62**
	CC	Both	3.46	**2.23**	21.17
Places365	CC	Grayscale	4.01	2.60	27.72
	CC	Directions	**3.43**	2.38	18.31
	CC	Both	3.60	2.45	21.47
Flickr100k	CC	Grayscale	4.09	2.67	27.09
	CC	Directions	3.70	2.48	20.86
	CC	Both	3.59	2.25	20.04
Ilsvrc12	NUS	Grayscale	3.14	2.24	22.39
	NUS	Directions	2.97	2.15	15.89
	NUS	Both	3.00	2.27	19.16
Places365	NUS	Grayscale	3.24	2.32	22.66
	NUS	Directions	**2.91**	2.24	16.05
	NUS	Both	3.07	2.20	17.12
Flickr100k	NUS	Grayscale	3.27	2.38	21.28
	NUS	Directions	2.95	**2.12**	16.40
	NUS	Both	2.98	2.16	**15.86**

Table 1. Statistics of angular errors (in degrees) obtained by variations of the proposed method on the CC and NUS datasets. Training has been performed on three datasets with different inputs: equalized grayscale, gradient directions, and their combination.

4.2. Fine tuning

When an annotated training set is available it is possible to improve the performance of the neural network by fine tuning its parameters. This is done by continuing the train-

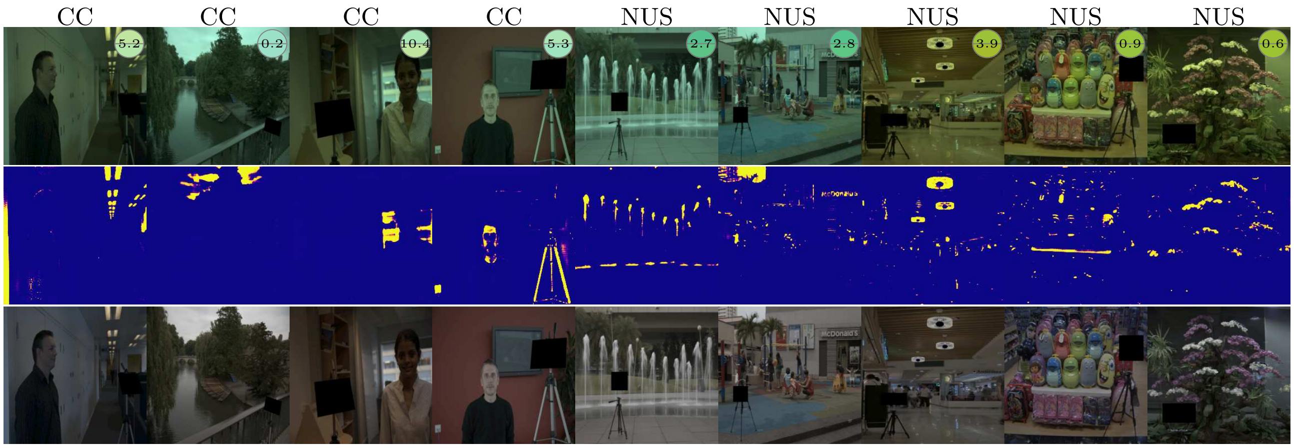

Figure 4. Examples of test images processed by the network trained on ILSVRC12 (grayscale version). From top to bottom the rows show the input image, the weights assigned to the pixels, and the images balanced according to the illuminant estimates. The colors of the circle over the input images show the illuminant estimate (upper half) and the ground truth (lower half). The number in the circle is the angular error expressed in degrees. For visualization purposes, input and balanced images have been gamma corrected.

Dataset	Mean	Median	Max
CC	2.91 (-0.55)	1.98 (-0.25)	19.9 (-1.2)
NUS	1.97 (-1.03)	1.41 (-0.86)	20.5 (+1.6)

Table 2. Statistics of angular errors (in degrees) obtained by the network trained on Ilsvrc12 and fine tuned on the two test datasets. The values in brackets report the difference with respect to those obtained in the quasi-unsupervised setting.

ing in a supervised way using a small learning rate. Here we performed 250 000 additional iterations, with a learning rate of 10^{-7}, and without the noise term in Equation (4). We repeated the experiment for both the Color Checker and the NUS datasets. In both cases we evaluated the final performance with a three-fold cross validation.

Table 2 reports the results we obtained by fine-tuning the neural network trained on Ilsvrc12 processing the combination of grayscale image and gradient directions (for the sake of brevity we omit the performance obtained by the other variants). For both test datasets the mean and the median angular error decreased. In the case of the NUS dataset the improvement was particularly noticeable, with more than one degree of difference in the mean error.

4.3. Comparison with the state of the art

Table 3 reports the statistics of the angular errors for several methods in the state of the art. The values have been taken from the literature or obtained by executing publicly available implementations. The methods are divided in unsupervised, parametric and supervised ones. The three categories are further split into: "in dataset", meaning that the method is trained/tuned with cross validation on the same color constancy dataset on which is tested; "cross dataset", meaning that the method is trained/tuned on one color constancy dataset and tested on a different one; "no dataset", meaning that the method is not trained/tuned on any color constancy dataset.

From the results reported in Table 3 and Figure 5 it is possible to notice that the proposed method is able to outperform all the purely unsupervised algorithms (i.e. Unsupervised no-db) in the state of the art by a large margin with a reduction of the median angular error by 37.9% and 9.6% on the CC and NUS datasets respectively, showing at the same time a more stable performance across all the different cameras. In the cross-dataset setting, the proposed method is able to outperform all the supervised methods. Concerning parametric methods, our method outperforms all of them on all the error statistics considered except for the median error on NUS. Interestingly, parametric methods have shown to perform better than supervised methods in this setting. In the completely supervised setting, the fine-tuned version of the proposed method is able to outperform all the parametric methods and to compete with the supervised ones obtaining the best mean error in the state of the art on NUS, and the second best median.

5. Conclusions

We presented here a method for computational color constancy which exploits a deep convolutional neural network and leverages large unannotated datasets thanks to a quasi-unsupervised learning procedure. We trained several variants of the method differing in the kind of information processed and in the training dataset. The experimental results showed that the proposed method is able to outperform

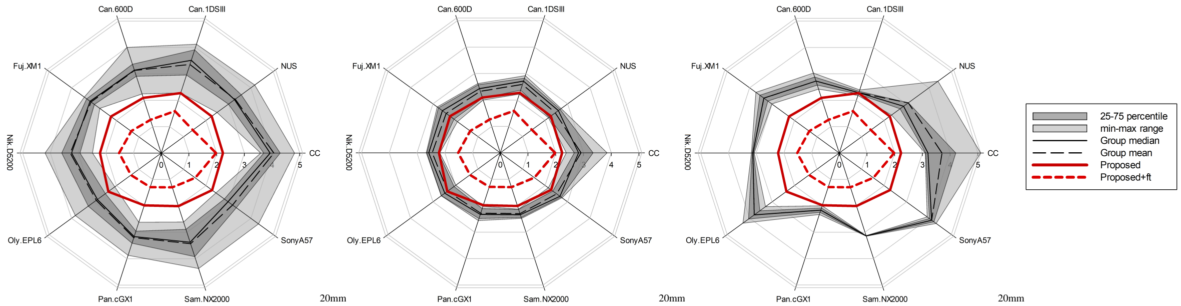

Figure 5. Visual summary of the median errors of the proposed method, with and without fine tuning, on CC, NUS and individual NUS cameras. The method is compared with three groups of algorithms. From left to right: unsupervised (no-db), parameteric (cross-db) and supervised (cross-db). For each group are drawn the best and worst median error, the interquartile range, the median ad the mean.

		Color Checker			NUS			NUS median, camera-by-camera								
	Method	Mean	Med.	Max	Mean	Med.	Max	C1	C600	Fuj.	N52	Oly.	Pan.	Sam.	Son.	N40
Unsupervised (in-db)	SoG [19] with GSA [3]	4.05	2.54	21.87	3.31	2.58	**21.01**	2.37	2.57	2.38	2.54	2.51	2.51	2.51	2.54	2.46
	gGW [4] with GSA [3]	4.05	2.58	21.15	3.45	2.68	22.48	2.45	2.61	2.61	2.59	2.71	2.77	2.63	2.55	2.66
	GE1[46] with GSA [3]	**4.03**	**3.08**	18.89	3.18	2.48	24.16	2.26	2.72	2.33	2.70	2.72	2.62	2.33	2.36	**2.36**
	GE2[46] with GSA [3]	4.13	3.34	**17.78**	3.41	2.52	31.21	2.27	2.59	2.39	2.66	2.71	2.61	2.63	2.45	2.37
	Banic and Loncaric [2]				**2.96**	**1.70**		**1.79**	**1.66**	**1.72**	**1.70**	**1.71**	**1.63**	**1.61**	**1.60**	
Unsupervised (no-db)	WP [34]	5.97	3.74	45.00	3.57	2.49	26.87	2.28	2.24	2.74	2.48	**2.07**	2.62	2.61	2.44	3.67
	GW [9]	4.76	3.59	24.92	4.17	3.17	22.34	3.84	3.13	3.29	3.39	2.63	3.07	2.98	2.94	3.50
	Buzzelli et al. (gl. norm)[10]	4.84	4.12	20.80	4.88	4.17	**18.70**	4.12	4.00	3.43	4.19	3.83	3.87	4.37	4.34	5.37
	Buzzelli et al. (ch. norm)[10]	5.48	4.81	**19.88**	4.32	3.37	22.36	3.18	3.15	3.06	3.08	3.06	3.26	3.77	3.02	4.76
	Proposed	**3.46**	**2.23**	21.17	**3.00**	**2.25**	19.16	**2.27**	**2.09**	**2.24**	**2.21**	2.36	**1.98**	**2.01**	**2.27**	**2.97**
Parametric (in-db)	SoG [19]	3.85	2.43	**20.89**	3.42	2.45	26.27	2.28	2.24	2.69	2.36	2.17	2.42	2.53	2.39	3.71
	gGW [4]	4.12	2.52	22.51	3.37	2.49	23.73	2.43	2.35	2.50	2.29	2.50	2.39	2.61	2.58	3.69
	GE1 [46]	4.06	2.67	23.05	3.18	2.18	**21.81**	2.37	2.00	**2.06**	2.23	1.98	2.02	2.13	2.34	**2.70**
	GE2 [46]	4.18	2.68	24.05	3.19	2.18	24.29	2.29	**1.86**	2.13	2.16	1.94	**2.01**	2.07	**2.31**	2.72
	BP[32]	3.98	2.61			2.48		2.45	2.48	2.67	2.30	2.18	2.15	2.49	2.62	3.13
	Cheng et al.[15]	**3.52**	**2.14**	28.35	**3.02**	**2.12**	23.28	**2.01**	1.89	2.15	**2.08**	**1.87**	2.02	**2.03**	2.33	2.72
	Grey Pixel (edge) [48]	4.60	3.10		3.15	2.20										
Parametric (cross-db)	SoG [19]	6.08	3.85	37.24	3.44	2.59	**18.40**	2.73	2.43	2.70	2.58	2.47	2.41	2.41	2.68	2.94
	gGW [4]	4.66	2.84	31.59	3.53	2.71	19.87	2.71	2.53	2.81	2.67	2.62	2.56	2.48	2.75	3.14
	GE1 [46]	4.06	2.67	**23.05**	3.18	2.18	21.81	2.37	**2.00**	2.06	2.23	**1.98**	**2.02**	**2.13**	2.34	**2.70**
	GE2 [46]	4.26	2.82	23.45	3.53	2.62	23.00	2.92	2.62	2.58	2.62	2.63	2.33	2.34	2.66	3.20
	Qian et al. [40]	**3.65**	**2.38**	26.12	**3.16**	**2.15**	21.93	**2.22**	2.07	**1.91**	**2.18**	2.06	2.11	2.14	**2.14**	3.05
Supervised (in-db)	Bayesian [23]	4.70	3.44			2.81		2.80	2.35	3.20	3.10	2.81	2.41	3.00	2.36	3.53
	Spatio-Spectral (ML) [13]	3.55	2.93			2.54		2.80	2.32	2.70	2.43	2.24	2.28	2.51	2.70	2.99
	Spatio-Spectral (GP) [13]	3.47	2.90			2.39		2.67	2.03	2.45	2.26	2.21	2.22	2.29	2.58	2.89
	Natural Image Statistics [24]	4.09	3.13			2.69		3.04	2.46	2.95	2.40	2.17	2.28	2.77	2.88	3.51
	Exemplar-based [31]	2.89	2.27													
	Chakrabarti (Empirical)[12]	2.89	1.89													
	Chakrabarti (End-to-end)[12]	2.56	1.67													
	Cheng et al. [16]	2.42	1.65			1.58		1.57	1.62	1.58	1.65	1.41	1.61	1.78	1.48	
	Color Dog [1]		1.49			1.76		1.72	1.85	1.81	1.94	1.46	1.69	1.89	1.77	
	Bianco et al. [8]	2.36	1.44	16.98		1.77		1.71	1.85	1.75	1.88	1.65	1.59	1.88	1.63	2.00
	FFCC [6]	1.78	**0.96**	**16.25**	1.99	**1.34**	**19.80**	**1.34**	1.33	1.35	**1.45**	**1.16**	**1.28**	1.47	**1.35**	**1.36**
	Oh and Kim [38]	2.16	1.47		2.41	2.15		2.18	1.75	2.75	2.00	2.22	1.53	1.65	3.11	2.68
	CCC (dist+ext) [5]	1.95	1.22		2.38	1.48										
	FC^4(AlexNet) [28]	**1.77**	1.11		2.12	1.53										
	DS-Net (HypNet+SelNet) [44]	1.90	1.12		2.24	1.46										
	Proposed + Fine Tuning	2.91	1.98	19.9	**1.97**	1.41	20.50	1.59	**1.26**	**1.34**	1.52	1.35	1.29	**1.30**	1.52	1.84
Supervised (cross-db)	Bayesian [23]	4.75	3.11		3.65	3.08										
	Exemplar-based [31]	6.50	5.10													
	Chakrabarti (Empirical) [12]	**3.87**	3.25		3.49	2.87		2.39	3.03	3.72	**3.07**	4.30	**2.00**	3.15	**3.92**	
	Chakrabarti (End-to-end) [12]	3.89	**3.10**		3.52	2.71		**2.18**	**2.42**	**3.01**	3.17	**3.29**	2.33	**3.13**	4.32	
	Cheng et al. [16]	5.52	4.52		4.86	4.40										
	FFCC [6]	3.91	3.15		**3.19**	**2.33**										

Table 3. Performance comparison with the state of the art in terms of angular error on the CC and NUS datasets.

the other unsupervised methods in the state of the art, being at the same time flexible enough to be supervisedly fine-tuned on a specific dataset reaching performance comparable with those of the top supervised methods. In this work we focused on the quasi-unsupervised setting. In the future we plan to explore more thoroughly the supervised fine tuning step, possibly by experimenting with more complex techniques taken from the literature on transfer learning and on domain adaptation.

Acknowledgment

We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPU used for this research.

References

[1] Nikola Banic and Sven Loncaric. Color dog-guiding the global illumination estimation to better accuracy. In *International Conference on Computer Vision Theory and Applications*, pages 129–135, 2015. 8

[2] Nikola Banić and Sven Lončarić. Unsupervised learning for color constancy. *arXiv preprint arXiv:1712.00436*, 2017. 1, 2, 8

[3] Nikola Banić and Sven Lončarić. Green stability assumption: Unsupervised learning for statistics-based illumination estimation. *arXiv preprint arXiv:1802.00776*, 2018. 2, 8

[4] Kobus Barnard, Vlad Cardei, and Brian Funt. A comparison of computational color constancy algorithms. ii: Experiments with image data. *IEEE Transactions on Image Processing*, 11(9):985–994, 2002. 2, 8

[5] Jonathan T Barron. Convolutional color constancy. In *IEEE International Conference on Computer Vision*, pages 379–387, 2015. 8

[6] Jonathan T Barron and Yun-Ta Tsai. Fast fourier color constancy. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 2, 8

[7] Simone Bianco, Claudio Cusano, and Raimondo Schettini. Color constancy using cnns. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 81–89, 2015. 1, 2

[8] Simone Bianco, Claudio Cusano, and Raimondo Schettini. Single and multiple illuminant estimation using convolutional neural networks. *IEEE Transactions on Image Processing*, 26(9):4347–4362, 2017. 2, 8

[9] Gershon Buchsbaum. A spatial processor model for object colour perception. *Journal of the Franklin institute*, 310(1):1–26, 1980. 2, 8

[10] Marco Buzzelli, Joost van de Weijer, and Raimondo Schettini. Learning illuminant estimation from object recognition. *arXiv preprint arXiv:1805.09264*, 2018. 2, 8

[11] Vlad C Cardei, Brian Funt, and Kobus Barnard. Estimating the scene illumination chromaticity by using a neural network. *Journal of the Optical Society of America A*, 19(12):2374–2386, 2002. 2

[12] Ayan Chakrabarti. Color constancy by learning to predict chromaticity from luminance. In *Advances in Neural Information Processing Systems*, pages 163–171, 2015. 2, 8

[13] Ayan Chakrabarti, Keigo Hirakawa, and Todd Zickler. Color constancy with spatio-spectral statistics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(8):1509–1519, 2012. 8

[14] Yu-Hsiu Chen, Ting-Hsuan Chao, Sheng-Yi Bai, Yen-Liang Lin, Wen-Chin Chen, and Winston H Hsu. Filter-invariant image classification on social media photos. In *ACM international conference on Multimedia*, pages 855–858, 2015. 1

[15] Dongliang Cheng, Dilip K Prasad, and Michael S Brown. Illuminant estimation for color constancy: why spatial-domain methods work and the role of the color distribution. *Journal of the Optical Society of America A*, 31(5):1049–1058, 2014. 5, 8

[16] Dongliang Cheng, Brian Price, Scott Cohen, and Michael S Brown. Effective learning-based illuminant estimation using simple features. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1000–1008, 2015. 2, 8

[17] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. 5

[18] Graham D Finlayson, Steven D Hordley, and Paul M Hubel. Color by correlation: A simple, unifying framework for color constancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(11):1209–1221, 2001. 2

[19] Graham D Finlayson and Elisabetta Trezzi. Shades of gray and colour constancy. In *Color and Imaging Conference*, volume 2004, pages 37–41. Society for Imaging Science and Technology, 2004. 2, 8

[20] David H Foster. Color constancy. *Vision research*, 51(7):674–700, 2011. 1

[21] Brian Funt and Weihua Xiong. Estimating illumination chromaticity via support vector regression. In *Color and Imaging Conference*, volume 2004, pages 47–52. Society for Imaging Science and Technology, 2004. 2

[22] Hiren Galiyawala, Kenil Shah, Vandit Gajjar, and Mehul S Raval. Person retrieval in surveillance video using height, color and gender. *arXiv preprint arXiv:1810.05080*, 2018. 1

[23] Peter Vincent Gehler, Carsten Rother, Andrew Blake, Tom Minka, and Toby Sharp. Bayesian color constancy revisited. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2008. 2, 5, 8

[24] Arjan Gijsenij and Theo Gevers. Color constancy using natural image statistics and scene semantics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(4):687–698, 2011. 8

[25] Arjan Gijsenij, Theo Gevers, Joost Van De Weijer, et al. Computational color constancy: Survey and experiments. *IEEE Transactions on Image Processing*, 20(9):2475–2489, 2011. 1, 2

[26] Steven D Hordley. Scene illuminant estimation: past, present, and future. *Color Research & Application*, 31(4):303–314, 2006. 2

[27] Steven D. Hordley and Graham D. Finlayson. Reevaluation of color constancy algorithm performance. *Journal of the Optical Society of America A*, 23(5):1008–1020, 2006. 6

[28] Yuanming Hu, Baoyuan Wang, and Stephen Lin. Fc4: Fully convolutional color constancy with confidence-weighted pooling. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4085–4094, 2017. 1, 2, 8

[29] Multimedia systems and equipment — colour measurements and management — part 2-1: Colour management — default RGB color space — sRGB'. Standard, International Electrotechnical Commission, IEC, 1999. 4

[30] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 5967–5976, 2017. 4

[31] Hamid Reza Vaezi Joze and Mark S Drew. Exemplar-based color constancy and multiple illumination. *IEEE*

Transactions on Pattern Analysis and Machine Intelligence, 36(5):860–873, 2014. 8

[32] Hamid Reza Vaezi Joze, Mark S Drew, Graham D Finlayson, and Perla Aurora Troncoso Rey. The role of bright pixels in illumination estimation. In *Color and Imaging Conference*, volume 2012, pages 41–46. Society for Imaging Science and Technology, 2012. 8

[33] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representation*, 2015. 5

[34] Edwin H Land and John J McCann. Lightness and retinex theory. *Journal of the Optical Society of America A*, 61(1):1–11, 1971. 2, 8

[35] Y LeCun, Y Bengio, and G Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015. 1

[36] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(12):2935–2947, 2017. 4

[37] Zhongyu Lou, Theo Gevers, Ninghang Hu, Marcel P Lucassen, et al. Color constancy by deep learning. In *BMVC*, pages 76–1, 2015. 1, 2

[38] Seoung Wug Oh and Seon Joo Kim. Approaching the computational color constancy as a classification problem through deep learning. *Pattern Recognition*, 61:405–416, 2017. 1, 8

[39] James Philbin, Ondrej Chum, Michael Isard, Josef Sivic, and Andrew Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007. 5

[40] Yanlin Qian, Ke Chen, Jarno Nikkanen, Joni-Kristian Kämäräinen, and Jiri Matas. Revisiting gray pixel for statistical illumination estimation. In *International Conference of Computer Vision Theory and Applications*, 2019. 2, 8

[41] Charles Rosenberg, Martial Hebert, and Sebastian Thrun. Color constancy using kl-divergence. In *IEEE International Conference on Computer Vision*, volume 1, pages 239–246. IEEE, 2001. 2

[42] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015. 2, 5

[43] Lilong Shi and Brian Funt. Shi's re-processed version of the gehler color constancy dataset of 568 images. `http://www.cs.sfu.ca/~colour/data`, Accessed: 12/11/2018. 5

[44] Wu Shi, Chen Change Loy, and Xiaoou Tang. Deep specialized network for illuminant estimation. In *European Conference on Computer Vision*, pages 371–387. Springer, 2016. 2, 8

[45] Irwin Sobel and Gary Feldman. A 3 × 3 isotropic gradient operator for image processing. Talk at Stanford Artificial Intelligence Project (SAIL), 1968. 4

[46] Joost Van De Weijer, Theo Gevers, and Arjan Gijsenij. Edge-based color constancy. *IEEE Transactions on image processing*, 16(9):2207–2214, 2007. 2, 8

[47] J von Kries. Chromatic adaptation, festschrift der albercht-ludwig-universität, 1902. 3

[48] Kai-Fu Yang, Shao-Bing Gao, and Yong-Jie Li. Efficient illuminant estimation for color constancy using grey pixels. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2254–2263, 2015. 8

[49] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *European Conference on Computer Vision*, pages 649–666, 2016. 3

[50] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017. 5