# Single and Multiple Illuminant Estimation Using Convolutional Neural Networks

Simone Bianco, *Member, IEEE*, Claudio Cusano, *Member, IEEE*, and Raimondo Schettini, *Member, IEEE*

***Abstract*— In this paper, we present a three-stage method for the estimation of the color of the illuminant in RAW images. The first stage uses a convolutional neural network that has been specially designed to produce multiple local estimates of the illuminant. The second stage, given the local estimates, determines the number of illuminants in the scene. Finally, local illuminant estimates are refined by non-linear local aggregation, resulting in a global estimate in case of single illuminant. An extensive comparison with both local and global illuminant estimation methods in the state of the art, on standard data sets with single and multiple illuminants, proves the effectiveness of our method.**

***Index Terms*— Color constancy, illuminant estimation, convolutional neural networks.**

## I. Introduction

THE observed color of the objects in the scene depends on the surface spectral reflectance of the object, on the illumination, and on their relative positions. Many computer vision problems in both still images and videos can make use of color constancy processing as a pre-processing step to make sure that the recorded color of the objects in the scene does not change under different illumination conditions.

In general there are two methodologies to obtain reliable color description from image data: computational color constancy and color invariance [1]. Computational color constancy is a two-stage operation: the former is specialized on estimating the color of the scene illuminant from the image data, the latter corrects the image on the basis of this estimate to generate a new image of the scene as if it was taken under a reference illuminant. Color invariance methods instead represent images by features which remain unchanged with respect to imaging conditions.

In this work we focus on illuminant estimation. Our method is based on supervised learning and includes a Convolutional Neural Network (CNN) specially designed for the local estimation of the illuminant color. Recently, deep neural networks have gained the attention of numerous researchers outperforming state-of-the-art approaches on various computer vision tasks [2], [3]. One of CNNs advantages is that it can take raw images as input and incorporate feature design into the training process. With a deep structure, CNN can learn complicated mappings while requiring minimal domain knowledge. The main limitation of deep learning approaches is the large amount of training data they need. This is a serious drawback for those problems where the ground truth is not easily available, as is the case of illuminant estimation that requires training images having a known color target in the scene. Since training images cannot be obtained from the web or other common image sources, the datasets that can be used to train and evaluate illuminant estimation methods are orders of magnitude smaller than those commonly used for deep learning. To deal with this problem we propose an hybrid approach where a CNN provides a spatially varying estimate of the illuminant that are then refined by a local regressor based on non-linear Support Vector Regression (SVR) with an adaptive support. Since the CNN performs a local analysis, it can be trained on a large number of patches extracted from a relatively small training set of images. The training of the final regressor does not pose any problem, since it requires a limited number of training samples. This approach has also the advantage of allowing the estimation of multiple illuminants for the same picture. To estimate the number of illuminants in the scene we designed a multiple illuminant detector exploiting a Kernel Density Estimator (KDE). The size of the support of the local regressor is computed by assigning the local estimates to the density peak(s), by back-projecting them on the original image, and by computing the Distance Transform (DT). In case of a single illuminant, the regressor support is thus the whole image and its application produces a single global estimate. Therefore our method can be considered general purpose one that is able to deal with single and multiple illuminants in a comprehensive way.

Preliminary findings reported in this paper appeared in [4], where we presented the basic architecture of the CNN and evaluated its performance in the single illuminant scenario. This paper extends the previous one in several ways:

- since one of the assumptions that is often violated in color constancy is the presence of a uniform illumination in the scene, we have extended the applicability of the proposed algorithm to the case of non-uniform illumination. The method is adaptive, being able to distinguish and process in different ways images of scenes taken under a uniform and those acquired under non-uniform illumination.

Manuscript received July 31, 2016; revised March 13, 2017; accepted May 28, 2017. Date of publication June 7, 2017; date of current version July 6, 2017. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Weisheng Dong. *(Corresponding author: Simone Bianco.)*

S. Bianco and R. Schettini are with the Department of Informatics, Systems and Communication, University of Milano-Bicocca, 20126 Milan, Italy (e-mail: simone.bianco@disco.unimib.it; raimondo.schettini@disco.unimib.it).

C. Cusano is with the Department of Electrical, Computer and Biomedical Engineering, University of Pavia, 27100 Pavia, Italy (e-mail: claudio.cusano@unipv.it).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2017.2713044

- In the case of uniform illumination, the multiple local estimates must be aggregated in a single global estimate. To do so we designed a new local regression method that replaces the per-channel median operator used in [4] with a non-linear mapping based on a RBF kernel over local statistics of the CNN estimates. The mapping is optimized by applying a regression procedure that minimizes the median angular error on the training set.
- In case of non-uniform illumination, the local estimates are refined using the same local regression method applied for uniform illumination, changing its support size on the basis of the spatial extent of the illuminants in the scene.
- Preliminary results reported in [4] included only images having a single color target in the scene, thus allowing only the comparisons with global illuminant estimation methods. We present a much more detailed experimental evaluation using both a multiple illuminant synthetic dataset and a dataset of RAW images containing at least two known color targets for benchmarking.

We show experimentally that the proposed method advances the state-of-the-art on standard datasets of RAW images for both the cases of single and multiple illuminants.

The rest of the paper is organized as follows: Section II formalizes the problem of illuminant estimation and reviews the main approaches in the state of the art. Section III illustrates in detail the proposed method. Section IV describes the data and the algorithms used in the experimentation, while Section V discusses the results obtained. Section VI reviews the architecture of the CNN on which our method is based, and gives insights on the learned model from a computational color constancy point of view. Finally, Section VII summarizes the findings of our experimentation and proposes new directions of research in this field.

## II. Problem Formulation and Related Works

The image values for a Lambertian surface located at the pixel with coordinates $(x, y)$ can be seen as a function $\boldsymbol{\rho}(x, y)$, mainly dependent on three physical factors: the illuminant spectral power distribution $I(x, y, \lambda)$, the surface spectral reflectance $S(x, y, \lambda)$ and the sensor spectral sensitivities $\mathbf{C}(\lambda)$. Using this notation $\boldsymbol{\rho}(x, y)$ can be expressed as

$$\boldsymbol{\rho}(x, y) = \int I(x, y, \lambda) S(x, y, \lambda) \mathbf{C}(\lambda) \mathrm{d}\lambda, \tag{1}$$

where $\lambda$ is the wavelength, $\boldsymbol{\rho}$ and $\mathbf{C}(\lambda)$ are three-component vectors and the integration is performed over the visible spectrum. The goal of color constancy is to estimate the color $\mathbf{I}(x, y)$ of the scene illuminant, i.e. the projection of $I(x, y, \lambda)$ on the sensor spectral sensitivities $\mathbf{C}(\lambda)$:

$$\mathbf{I}(x, y) = \int I(x, y, \lambda) \mathbf{C}(\lambda) \mathrm{d}\lambda. \tag{2}$$

Usually the illuminant color is estimated up to a scale factor as it is more important to estimate the chromaticity of the scene illuminant than its overall intensity [5]. Thus, the error metric usually considered, as suggested by Hordley and Finlayson [5], is the angle between the RGB triplet of estimated illuminant ($\hat{\mathbf{I}}(x, y)$) and the RGB triplet of the measured ground truth illuminant ($\mathbf{I}(x, y)$):

$$e_{\mathrm{ANG}}(x, y) = \arccos\left( \frac{\mathbf{I}(x, y)^t \hat{\mathbf{I}}(x, y)}{\|\mathbf{I}(x, y)\| \|\hat{\mathbf{I}}(x, y)\|} \right). \tag{3}$$

Since the only information available are the sensor responses $\boldsymbol{\rho}$ across the image, color constancy is an under-determined problem [6] and thus further assumptions and/or knowledge are needed to solve it. Several computational color constancy algorithms have been proposed, each based on different assumptions. The most common assumption is that the color of the light source is uniform across the scene, i.e. $\mathbf{I}(x, y) = \mathbf{I}$ The next two sections review single and multiple illuminant estimation algorithms in the state of the art.

### A. Single Illuminant Estimation

Methods for single illuminant estimation can be divided into two main classes: statistic approaches, and learning-based approaches. Statistic approaches estimate the scene illumination only on the base of the content in a single image making assumptions about the nature of color images exploiting statistical or physical properties; learning-based approaches require training data in order to build a statistical image model, before the estimation of the illumination.

*1) Statistic-Based Algorithms:* van de Weijer *et al.* [7] have unified a variety of algorithms. These algorithms estimate the illuminant color $\mathbf{I}$ by implementing instantiations of the following equation:

$$\mathbf{I}(n, p, \sigma) = \frac{1}{k} \left( \iint \left| \nabla^n \boldsymbol{\rho}_\sigma(x, y) \right|^p \mathrm{d}x \ \mathrm{d}y \right)^{\frac{1}{p}}, \tag{4}$$

where $n$ is the order of the derivative, $p$ is the Minkowski norm, $\boldsymbol{\rho}_\sigma(x, y) = \boldsymbol{\rho}(x, y) \otimes G_\sigma(x, y)$ is the convolution of the image with a Gaussian filter $G_\sigma(x, y)$ with scale parameter $\sigma$, and $k$ is a constant to be chosen such that the illuminant color $\mathbf{I}$ has unit length (using the $2-$norm). The integration is performed over all pixel coordinates. Different $(n, p, \sigma)$ combinations correspond to different illuminant estimation algorithms. Are examples within this framework the Gray World algorithm [8] ($n = 0,\ p = 1,\ \sigma = 0$), the White Patch algorithm [9] ($n = 0,\ p = \infty,\ \sigma = 0$), and The Gray Edge algorithm [7] ($n = 1,\ p = 0,\ \sigma = 0$).

Examples of statistic-based methods not following eq. (4) are Gamut Mapping [10] and Color by Correlation [11].

*2) Learning-Based Algorithms:* The learning-based illuminant estimation algorithms, that estimate the scene illuminant using a model that is learned on training data, can be subdivided into two main categories: probabilistic methods and fusion/selection based methods.

One of the first learning-based algorithms is [12], where a Neural Network was trained on binarized chromaticity histograms. In [13] a neural network is used to better understand the color constancy in the human visual system, concluding that the human visual system achieves color constancy by concurrently calculating the difference between the test object and the background color as well as determining the background color.

Bayesian approaches [14] model the variability of reflectance and of illuminant as random variables, and then estimate illuminant from the posterior distribution conditioned on image intensity data.

Given a set illuminant estimation algorithms, in [15] an image classifier is trained to classify the images as indoor and outdoor, and different experimental frameworks are proposed to exploit this information in order to select the best performing algorithm on each class. In [16] it has been shown how intrinsic, low level properties of the images can be used to drive the selection of the best algorithm (or the best combination of algorithms) for a given image. In [17] the Weibull parametrization has been used to train a maximum likelihood classifier based on mixture of Gaussians to select the best performing illuminant estimation method for a certain image.

In [18] a statistical model for the spatial distribution of colors in white balanced images is developed, and then used to infer illumination parameters as those being most likely under their model. High level visual information has been used to select the best illuminant out of a set of possible illuminants [19]. In [20] and [21] the use of automatically detected objects having intrinsic color is investigated. In particular, they showed how illuminant estimation can be performed exploiting the color statistics extracted from the faces automatically detected in the image. When no faces are detected in the image, any other algorithm in the state-of-the-art can be used. In [22] and [23] the surfaces in the image are exploited and the illuminant estimation problem is addresses by unsupervised learning of an appropriate model for each training surface in training images. The model for each surface is defined using both texture features and color features.

In [24] it was showed how simple moment based algorithms can, with the addition of a simple correction step deliver much improved illuminant estimation performance. The approach employs first, second and higher moments of color and color derivatives and linearly corrects them to give an illuminant estimate. In [25] four simple image features are used for training an ensemble of decision trees. Each of these trees is computed from samples in the training data that are biased to a local region in chromaticity space of the ground truth illuminations. The final estimate is made by finding consensus among the different features trees estimations. In [26] illuminant color is predicted from luminance-to-chromaticity based on a conditional likelihood function for the true chromaticity of a pixel, given its luminance. Two approaches have been proposed to learn this function. The first was based purely on empirical pixel statistics, while the second was based on maximizing accuracy of the final illuminant estimate. In [27] the illuminant estimation problem is reformulated as a 2D spatial localization task in a log-chrominance space, applying techniques from object detection and structured prediction. The method directly learns how to discriminate between correctly white-balanced images and poorly white-balanced ones scoring by convolution each tinted image, and then returning the highest-scoring tint as the estimated illumination of the input image.

*3) CNN-Based Algorithms:* In [4] two different approaches using CNNs were investigated: in the first one an ad-hoc CNN for the color constancy problem was trained; in the second one a pre-trained one was used by extracting a 4096-dimensional feature vector from each image using the Caffe [28] implementation of the deep CNN described by Krizhevsky *et al.* [3]. Features were computed by forward propagation of a mean-subtracted $227 \times 227$ RGB RAW image through five convolutional layers and two fully connected layers. More details about the network architecture can be found in [3] and [28]. The CNN was discriminatively trained on a large dataset (ILSVRC 2012) with image-level annotations to classify images into 1000 different classes. Features are obtained by extracting activation values of the last hidden layer. The extracted features were then used as input to a linear Support Vector Regression (SVR) [29] to estimate the illuminant color for each image. In [30] a deep CNN inspired by [3] is used. To overcome the lack of large scale training datasets, a three-step learning strategy is proposed. First, the CNN was discriminatively trained on ILSVRC; then, the CNN is fine-tuned on the same dataset with generated light sources, i.e. using labels coming from a color constancy algorithm in the state of the art; finally, the CNN is fine-tuned on datasets with real ground-truth labels. In [31] the illumination estimation problem is cast as a classification problem. The training images are first clustered according to the illumination color assigned to each image and the images with the new labels (i.e. the cluster ids) are used to train the CNN. The CNN is designed to output the probabilities of the given image belonging to each illumination cluster. The final estimate is obtained by linear combination of the cluser probabilities and the cluster centroids.

### *B. Multiple Illuminant Estimation*

The great majority of state-of-the-art illuminant estimation methods assumes that a uniform illumination is present in the scene. This assumption is often violated in real-world images. It is not trivial to extend the existing illuminant estimation algorithms to work locally instead of globally, since the spatial support on which they accumulate the statistics is reduced, and the final local estimate could be biased by local image properties. One of the first methods following this strategy is Retinex [9], which is able to deal with non-uniform illumination assuming that an abrupt change in pixel values is caused by a change in reflectance properties. This implies that the illuminant smoothly varies across the image and does not change between adjacent or nearby locations. Ebner [32] proposed a method that assumes that the illuminant transition is smooth. The method uses the local space average color for local estimation of the illuminant by convolving the image with a Gaussian kernel function. Bleier *et al.* [33] investigated whether existing color constancy methods, originally developed assuming uniform illumination, can be adapted to local illuminant color estimation using image sub-regions. Multiple independent estimations are then combined through regression to obtain a more robust final estimate. Gijsenij *et al.* [34] proposed a method that makes use of local image patches, which can be selected by any sampling method. After sampling of the patches, illuminant estimation techniques are applied to

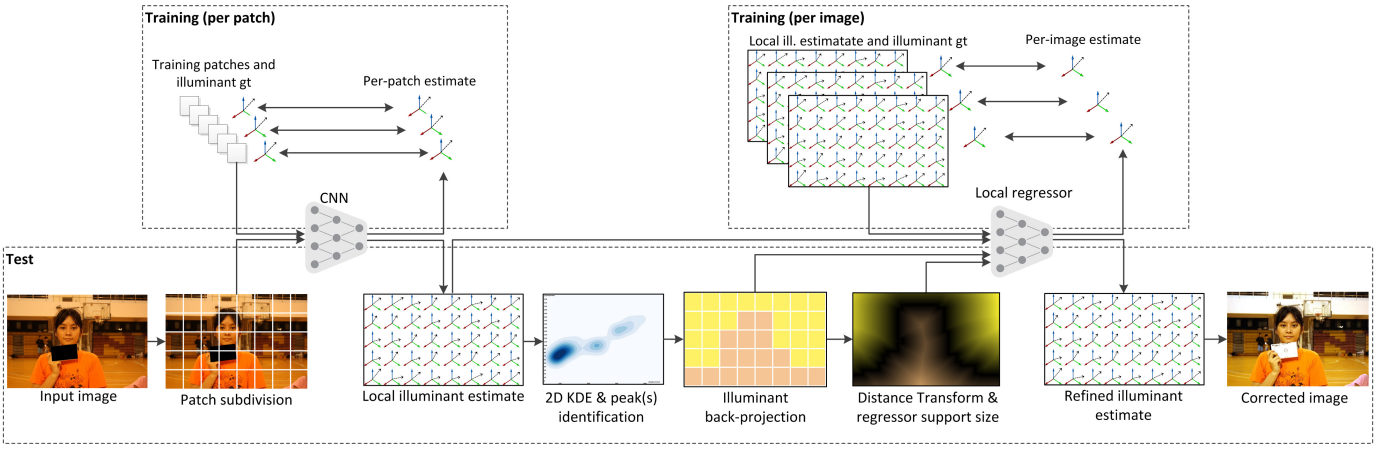



Fig. 1. Scheme of the proposed illuminant estimation method showing the connections among the modules during both training and test phases.

obtain local illuminant estimates, and these estimates are combined into more robust estimations, since it is assumed that the number of different lights is less than the number of patches. This combination of local estimates is done with two different approaches: clustering if the number of lights is known, segmentation otherwise. Recently Bianco and Schettini [21], and Joze and Drew [23] respectively extended the face-based and exemplar-based color constancy algorithms to deal with multiple illuminations. A different class of algorithms is based on user guidance to deal with the case of two [35] and multiple lights [36].

## III. The Proposed Approach

In the last years deep learning techniques have made it possible to achieve significant improvements in several computer vision tasks. Their success often depends on the availability of a large amount of annotated training data. Compared to other image-related problems, in illuminant estimation annotated data is scarce. Therefore, the straightforward procedure of learning the most probable illuminant color directly from the image pixels needs some major adjustments.

We propose a three-stage method: the first stage is patch based, that is, a CNN is trained to predict the illuminant color from a small square portion of the input image. A large training set of patches can be obtained even from a relatively small data set of images, making it possible the use of deep learning techniques. This first stage allows to obtain multiple local estimates of the illuminant across the input image.

The second stage determines the number of illuminants in the scene. This decision is taken on the basis of a statistical analysis of the local estimates produced by the first stage.

The third stage refines the illuminant estimates by non linear local aggregation. In the case the second stage determines that the scene has been taken under a single illuminant, it is better to aggregate the local estimates into a single prediction. For this purpose, in our previous work [4] we experimented with the mean and the per-channel median operators. In this work we propose a local aggregation procedure based on supervised learning. More in detail, statistical features are extracted from the local estimates, and then fed to a non-linear mapping whose output is the final global estimate of the color of the illuminant. The support of the local regression is thus the whole image. Differently from the first stage, this stage is image based. Therefore, its complexity is limited by the small number of annotated images. For this reason, instead of using a deep learning approach, we adopted a "shallow" non-linear regression scheme.

In case of multiple illuminants, the support of the local regression is smaller, thus performing a refinement of the local illuminant estimates. Local illuminant estimates are grouped together in the number of groups estimated in the second stage, and the group assignments are by back-projected on the original image. For each group the distance transform is computed, and the local regression support size is computed by calculating the minimum over all groups of the maximum values of the distance transforms. Note that the procedure used for the single illuminant is just a special case of this more general strategy.

Figure 1 shows a schematic view of the proposed method.

### A. Local Illuminant Estimation

In the first stage a convolutional neural network produces local estimates of the illuminant. The network, described in greater detail in Section VI, takes as input non-overlapping patches that have been previously subjected to a stretching of the histogram so that the output estimate is invariant with respect to the local contrast. The network is composed by the the following sequence of layers (see also Figure 2 for a graphical representation):

- input RGB patches of size $32 \times 32 \times 3$;
- a bank of 240 convolutional $1 \times 1 \times 3$ filters producing an output of size $32 \times 32 \times 240$;
- downsampling via an $8 \times 8$ max pooling layer to a size of $4 \times 4 \times 240$;
- reshaping of the result of pooling into a 3840-dimensional vector;
- a linear $3840 \times 40$ layer producing a 40-dimensional feature vector;
- a ReLU activation function;
- a linear $40 \times 3$ layer producing the output RGB estimate.

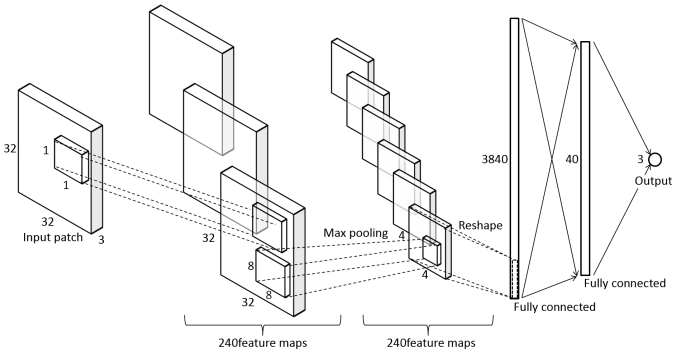



Fig. 2. The architecture of the CNN that produces the local estimates.

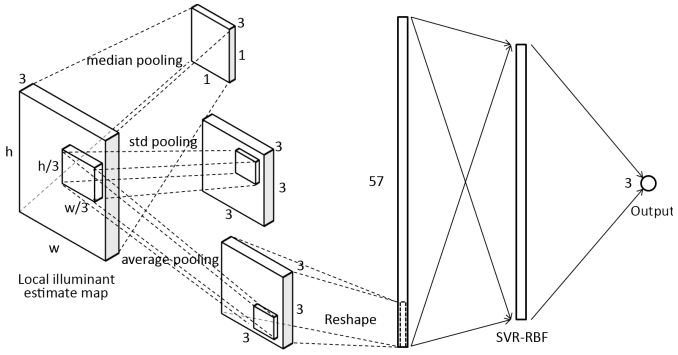



Fig. 3. The architecture of our local regressor.

Taking into account all the linear coefficients and the biases, the network include a total of 154,723 parameters that have been learned by applying the standard back propagation algorithm to minimize the average Euclidean squared difference between the estimated and the ground truth illuminant colors (we also tried to minimize the cosine loss without any improvement). Beside its size, compared to the networks used for scene and object recognition we notice two major differences: (i) $1 \times 1$ convolutional filters, and (ii) the large $8 \times 8$ pooling. These differences can be motivated by considering that with respect to object/scene recognition, illuminant estimation is a dual problem: instead of trying to identify the content of the image regardless the illuminant, here we need to estimate the illuminant regardless the content of the image. A detailed interpretation of the model from a color constancy point of view is given in Section VI-A.

### *B. Detection of Multiple Illuminants*

Since our CNN is applied to each patch independently, it can be easily used to predict local illuminants. However, local estimates tend to be noisy and sometimes (when there is a single illuminant, or when the color of all the light sources is very similar) it is better to replace them with a single global estimate. What we need is an automatic rule to switch between the two modalities. In order to decide if the image contains single or multiple illuminants, the per patch illuminant estimates are normalized and projected onto the normalized chromaticity plane ($r^c = R/G, b^c = B/G$). Then, an efficient 2D kernel density estimation (KDE) [37] is applied. The modes $(r_i, b_i)$, $i = 1, \ldots, n$, i.e. the red/blue chromaticities with the highest densities are identified using a scale-space filtering [38]. Only the modes with a value higher than $t$ times the maximum are retained:

$$J = \left\{ j \in \{1, \ldots, m\} : \frac{density(r_j^c, b_j^c)}{\max_{i=1,\ldots,n} density(r_i^c, b_i^c)} \geq t \right\}. \tag{5}$$

The angular difference between each pair of the retained modes $((r_j^c, 1, b_j^c),\ j \in J)$ is computed. If the maximum difference exceeds a set threshold then the scene is considered as taken under multiple illuminants. Otherwise, we proceed by assuming the presence of a single illuminant.

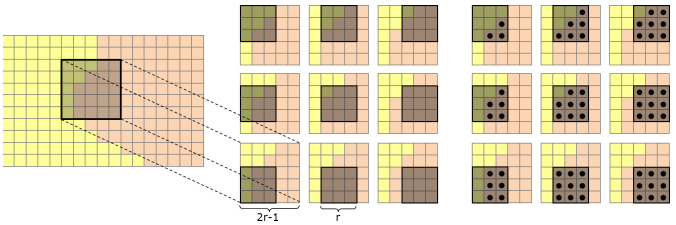



Fig. 4. Example of the use of the local regressor with a support $L = 5$ and two modes. Left: back-projected illuminants, where each estimate is assigned to the closest mode. Middle: pooling regions with size $L \times L$ and a stride of $(r - 1)/2$. Right: for each pooling region, the locations of the estimates assigned to the same mode of the central patch are indicated with black dots.

Following [21] and [39] we set the threshold to 3°, since it has been judged to be a noticeable but acceptable difference.

### *C. Local Aggregation of the Estimates*

In our previous work [4] we generated a single illuminant estimation per image by pooling the predicted illuminants on the image patches. By taking image patches as input, we have a much larger number of training samples compared to using the whole image on a given dataset, which particularly meets the needs of CNNs, but we loose the information that certain patches belong to the same image. Thus, we fine-tuned the learned net by adding knowledge about the way local estimates are pooled to generate a single global estimate for each image.

In this work we extend the per-channel average and median pooling operators used in [4] with a non-linear mapping based on a RBF kernel over local statistics of the CNN estimates. The parameters of the mapping are obtained by applying a regression procedure that minimizes the median angular error on the training set. Given as input the map of the per-patch illuminant estimates having a size of $w \times h$, the first step in this module is the smoothing via convolution with a $5 \times 5$ Gaussian filter. The response is then independently pooled in three different ways: average pooling and standard deviation pooling both with size $w/3 \times h/3$ (i.e. on a subdivision in nine rectangular regions), and median pooling with size $w \times h$ (i.e. on the whole image). These values are reshaped and given as input to a SVR (with RBF kernel) which predicts the global illuminant by minimizing the median angular error over the training set. The architecture of this module is reported in Figure 3.

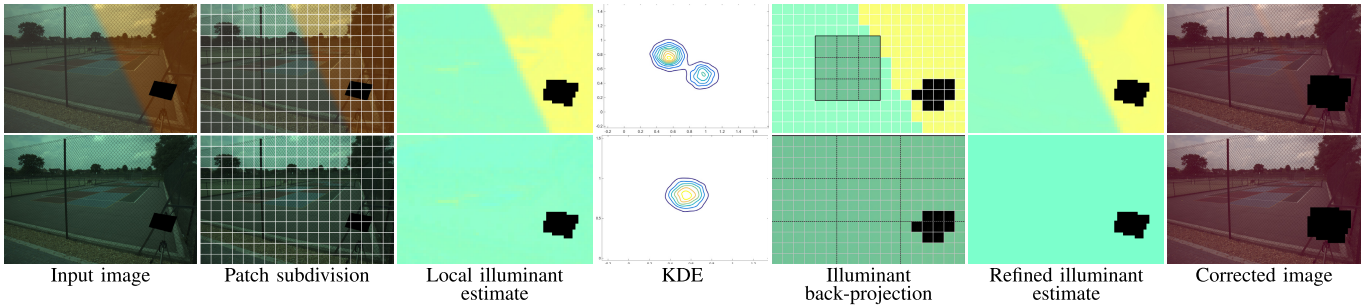



Fig. 5. Output of each stage of the proposed illuminant estimation method in the case of multiple (top row) and single illuminants (bottom row). From left to right: input image, subdivision in patches, local illuminant estimate, output of KDE where it is possible to see the peaks of the different illuminants found; illuminant back-projection with overlaid regressor support size; refined illuminant estimate; corrected images.

In case of multiple detected illuminants, the non-linear mapping described above for the single illuminant case is applied to refine the local patch-by-patch estimates. First, the estimates are assigned to the closest retained modes $M_j = (r_j^c, 1, b_j^c)$, $j \in J$ representing the detected illuminants. Then, for each local estimate, a neighborhood of $L \times L$ estimates is considered, where $L$ is adaptively determined on the basis of the spatial distribution of the illuminants. The median of the estimates in the neighborhood is computed together with the average and the standard deviation computed on nine square pooling regions, forming a vector of $3 + 9 \times 3 + 9 \times 3 = 57$ components. These statistics are computed by considering only the estimates assigned to the same illuminant of the central one. The sizes $L$ of the neighborhood and $r$ of the pooling regions are determined on the basis of the Distance Transform (DT) [40] of the back projection of each mode $M_j$ as follows:

$$s = \max\left\{\left\lfloor \frac{1}{2}\min_{j\in J}\left(\max DT\left(M_j\right)\right)\right\rceil, 3\right\}, \tag{6}$$

$$r = s + 1 - (s \mod 2), \tag{7}$$

$$L = 2r - 1. \tag{8}$$

Note that in this way the size $r$ of the pooling regions is ensured to be an odd number, greater than two. By taking the regions with a stride of $(r-1)/2$ estimates, we also obtain that the central estimate belongs to all the nine regions, making it sure that at least one estimate contributes to the statistics computed in the pooling regions, allowing us to deal with complicated cases such as the one in which all the surrounding estimates are assigned to a different illuminant than that of the central one. Figure 4 shows an example of local estimation in the case $L = 5, r = 3$.

In Figure 5 the output of each stage of the proposed illuminant estimation method is showed in the case of multiple and single illuminants.

## IV. Experimental Setup

### A. Image Datasets and Evaluation Procedure

To test the performance of the proposed algorithm for the global illuminant estimation, two standard datasets of RAW camera images having a known color target are used. In the first dataset, images have been captured using high-quality digital SLR cameras in RAW format, and are therefore free of any color correction. The dataset [14] was originally available in sRGB-format, but Shi and Funt [41] reprocessed the raw data to obtain linear images with a higher dynamic range (14 bits as opposed to standard 8 bits). The dataset has been acquired using a Canon 5D and a Canon 1D DSLR cameras and consists of a total of 568 images. The Macbeth ColorChecker (MCC) chart is included in every scene, and this allows to accurately estimate the actual illuminant of each acquired image. As suggested [41], the camera's black level offset is removed before any processing. The second dataset is the NUS dataset [42]. The dataset is similar to the previous one: it has been captured using digital SLR cameras in RAW format with a MCC included in every scene. The differences with the previous dataset are that it has been captured by 9 different cameras and that there is a larger number of images, i.e. 1853 with around 200 images for each camera.

To test the performance of the proposed algorithm for the multiple illuminant estimation, three different datasets have been used. The first one is synthetically generated from the Gehler-Shi dataset: each image is relighted using two, three and four random illuminants taken from the same datasets. This synthetic dataset thus contains a total of 1704 images. The second dataset used is a subset of the Milan portrait dataset [21]. It has been acquired in RAW format using four different DSLR cameras. The dataset is the union of different subsets that have been acquired in three different world locations: Italy, Taiwan, and Japan. The dataset includes portraits of a single person with a single MCC up to multiple persons with multiple MCCs. In this work we used the subset containing multiple MCCs, for a total of 197 images. Finally, the third one is the multiple illuminant dataset by Beigpour *et al.* [43]. It has been acquired using a Sigma SD10 single-lens reflex (SLR) digital camera which uses a Foveon X3 sensor and is available in linear RAW format. The dataset consist of two parts: the first one is taken in controlled laboratory setting for a total of 10 scenes taken under six distinct illumination conditions; the second one is taken in uncontrolled setting for a total of 20 indoor and outdoor scenes. The datasets comes with pixel-wise ground truth information.

The network has been trained on the Gehler-Shi dataset and adapted to the other datasets by re-training each time the local

Fig. 6. Examples of images within the image datasets considered. Top to bottom: Gehler-Shi, NUS, Milan portrait and Beigpour et al. datasets.

regressor to cope with the different cameras and sensor type used. Examples of images within the datasets considered are reported in Figure 6.

*Relighted Gehler-Shi Dataset:* We synthetically generated a relighted version of the Gehler-Shi dataset: each image is balanced using the corresponding ground truth illuminant and relighted using two, three and four random illuminants taken from the original dataset. Their position in the image was set randomly with the constraint of being at least $\min\{w, h\}/3$ apart, with $w$ and $h$ being image width and height respectively. The ground truth for each image has been generated by nearest-neighbor assignment followed by Gaussian smoothing to simulate illuminant mixing. This synthetic dataset thus contain a total of 1704 images.

### B. Benchmark Algorithms

Different benchmarking algorithms for color constancy are considered. Since each image of the dataset contains only one MCC, only global color constancy algorithms based on the assumption of uniform illumination can be compared. Six of them are generated varying the three variables $(n, p, \sigma)$ in Equation 4, and correspond to well known and widely used illuminant estimation algorithms. The values chosen for $(n, p, \sigma)$ are set as in [44]. The algorithms are used in the original authors' implementation which is freely available online (http://lear.inrialpes.fr/people/vandeweijer/code/ColorConstancy.zip). The seventh algorithm is the pixel-based Gamut Mapping [45]. The other algorithms considered are illumination chromaticity estimation via Support Vector Regression (SVR [46]); the Bayesian (BAY [14]); the Natural Image Statistics (NIS [17]); the High Level Visual Information [19]: bottom-up (HLVI BU), top-down (HLVI TD), and their combination (HLVI BU&TD); the Spatio-Spectral statistics [18]: with Maximum Likelihood estimation (SS ML), and with General Priors (SS GP); the Automatic color constancy Algorithm Selection (AAS) [16] and the Automatic Algorithm Combination (AAC) [16]; the Exemplar-Based color constancy (EB) [22]; the Face-Based (FB) color constancy algorithm [20] using GM or SS ML when no faces are detected; the CNN-based algorithms [4] and the AlexNet fine-tuned with a linear Support Vector Regression (SVR) [29] to estimate the illuminant color for each image [4] (AlexNet+SVR); the ensemble of regression trees applied to simple color features [25] (SF); the corrected-moment illuminant estimation [24] (CM); the one predicting chromaticity from pixel luminance (PCL) [26]; the one exploiting bright pixels (BP) [47] and the one exploiting both bright and dark pixels (BDP) [42]; the Convolutional Color Constancy (CCC) [27]; the color constancy by deep learning (CCDL) [30]; and the one approaching the computational color constancy as a classification problem (CCP) [31]. The last algorithm considered is the Do Nothing (DN) algorithm which gives the same estimation for the color of the illuminant ($\mathbf{I} = [1\ 1\ 1]$) for every image, i.e. it assumes that the image is already correctly balanced. A schematic comparison of the proposed method with the recent, most related, and best performing algorithms in the state of the art are reported in Table I.

### C. Learning of the Main Modules

We train our CNN on $32 \times 32$ patches randomly taken from training images of the Gehler-Shi dataset in RAW format (patches including portions of the reference MCC are excluded from training). Images have been resized to $\max(w, h) = 1200$ pixels. The net is learned using a three-fold cross validation on the folds provided with the dataset: for each run one is used for training, one for validation and the remaining one for test. For training, we assign each patch with the illuminant ground truth associated to the image to which it belongs. At testing time, we generate a single illuminant estimation per image by pooling the the predicted patch illuminants. By taking image patches as input, we have a much larger number of training samples compared to using the whole image on a given dataset, which particularly meets the needs of CNNs. Net parameters have been learned using Caffe [28] with Euclidean loss.

The learned net is then applied to each whole image in the training set by masking the MCC to obtain an illuminant estimation map. The pooled features computed from these maps are the input to our local regressor to give a single global illuminant estimate for each image. We train our regressor using the same three-fold cross validation as before using an $\epsilon$-SVR [29] with RBF kernel in which we used a modified cost function to minimize the median angular distance between illuminant estimates and ground-truths. The regressor is able to give a more accurate global estimate than a simple average or median pooling [4] for two reasons: (i) it is learning-based and is able to leverage the different local estimates coming from the patches belonging to the same image; (ii) it is trained by explicitly minimizing the error metric using in the evaluation of illuminant estimation methods. The regressor is learned on single illuminant images, thus using as support size equal to the whole image.

TABLE I

SCHEMATIC COMPARISON OF THE PROPOSED METHOD WITH THE RECENT, MOST RELATED, AND BEST PERFORMING METHODS IN THE STATE OF THE ART

| Algorithm | Features | Classifier/estimator | Task | CNN architect. | Single/Multi illuminant | Dataset tested[1] |
|---|---|---|---|---|---|---|
| Illuminant estimation using simple features (SF) [25] | **Hand-crafted**: chromaticity of the average, the brightest and the dominant color, and chromaticity mode of the color palette. | Ensemble of regression trees. The estimate is found by consenus among the trees. | Regr. | – | •/○ | ••○○○ |
| Predicting chromaticity from pixel luminance (PCL) [26] | **Hand-crafted**: luminance of individual pixels to compute the conditional likelihood on a discretized domain of true pixel chromaticity given observed luminance. | Aggregation of the conditional likelihood over all image pixels. The estimate is the chromaticity with the highest likelihood. | Class. | – | •/○ | •○○○○ |
| Convolutional Color Constancy (CCC) [27] | **Hand-crafted**: histograms in the log-chrominance space of the input image itself, and its sharpened and rectified, its soft max-filtered, and its standard-deviation-filtered versions. | Convolution of the histograms with a discriminatively learned filter. The estimate is the chromaticity with the highest soft-max probability. | Class. | – | •/○ | ••○○○ |
| Color constancy by deep learning (CCDL) [30] | **Fine-tuned**: CNN pre-trained for object classification, and then fine-tuned in two steps: first on generated light sources and then on real data. | Global estimate produced by the CNN itself. | Regr. | Inspired from [3] | •/○ | •○○○○ |
| Color constancy as a classification problem (CCP) [31] | **Fine-tuned**: CNN pre-trained for object classification, and then fine-tuned to predict the illuminant class. | The estimate is the linear combination of the probabilities of the illuminant to belong to different illuminant clusters and the clusters' centroids. The final estimate is averaged over 200 passes of the CNN on transformations of the input image. | Class. | Inspired from [3] | •/○ | ••○○○ |
| Proposed | **Learned**: supervised feature learning in a deep learning framework using a CNN especially designed for color constancy. The features are learned from scratch. | Local estimates produced by the CNN itself, and then locally aggregated with a SVR regressor. | Regr. | Custom | •/• | ••••• |

[1] Gehl.-Shi / NUS / Milan portr./Beigpour et al./cross-db

## V. RESULTS AND DISCUSSION

We evaluated the proposed method in both single and multiple illuminant estimation.

### A. Global Illuminant Estimation

In Table II the median, the average, the $90^{th}$-percentile, and the maximum of the angular errors obtained by the considered state-of-the-art algorithms and the proposed approach on the Gehler-Shi dataset are reported. The table is divided into three blocks and for each of them the best result for each statistic is reported in bold. The first block includes statistic-based algorithms, the second one learning-based algorithms, and the third one the different variants of the proposed approach. From the results it is possible to see that the deep CNN pre-trained on ILSVRC 2012 [3] coupled with SVR (i.e. AlexNet+SVR) is already able to outperform most statistic-based algorithms and some learning-based ones. The CNN introduced in our previous work [4] in its various instantiations allowed to obtain a median angular error below 2 degrees which is better than almost all the other methods considered. Even better results have been obtained with the recent method by Barron [27] for which the median error is 1.22 degrees. The method proposed here obtained the second lowest error (1.44 degrees if we consider the median). The ranking of the algorithms does not change if we consider the mean error instead of the median; the best maximum error, instead has been obtained by the fine-tuned CNN [4]).

Note that for this experiment we did not apply the multiple illuminant detection module and we always performed the local estimate aggregation using a support size for the regressor equal to the whole image. This last step brings a significant improvement. In fact, without it the median error raises by more than one degree, reaching the 2.69 degrees corresponding to the "CNN per patch" result. It is also a significant improvement with respect to the other aggregation methods considered in our previous work: average pooling, median pooling and fine tuning, that obtained median errors of 2.44, 2.32 and 1.98, respectively.

Figure 7 reports some examples of images on which the proposed illuminant estimation method makes the largest errors. Even if during the illuminant estimation phase, the patches overlapping the MCC are ignored, they are left unmasked in the figure to better appreciate the results. Once we have an estimate of the global illuminant color $\mathbf{I}$, each pixel in the image is color corrected using the von Kries model [48], i.e.: $\boldsymbol{\rho}_{out}(x, y) = diag(\mathbf{I}^{-1})\boldsymbol{\rho}_{in}(x, y)$. Although this dataset is used to evaluate global illuminant estimation methods, Xu and Funt [49] showed that it contains images with mixed-light conditions. They manually split this dataset into images with uniform and non-uniform illumination, and compare on them the performance of different global illuminant estimation methods. We run the same experiment here, obtaining a median angular error on their splits of 1.43 and 1.49 degrees respectively. This demonstrates that,

TABLE II
ANGULAR ERROR STATISTICS OBTAINED BY THE STATE-OF-THE-ART ALGORITHMS CONSIDERED ON THE GEHLER-SHI DATASET. ALGORITHMS ARE DIVIDED IN THREE GROUPS (STATISTIC-BASED, LEARNING-BASED, INVESTIGATED IN THIS PAPER) AND, FOR EACH GROUP, THEY ARE SORTED BY DECREASING MEDIAN ANGULAR ERROR

| Algorithm | Med | Avg | $90^{th}$prc | Max |
|---|---|---|---|---|
| DN | 13.55 | 13.62 | 16.45 | 27.37 |
| GW | 6.30 | 6.27 | 10.12 | 24.84 |
| WP | 5.61 | 7.46 | 15.68 | 40.59 |
| GE1 | 4.55 | 5.21 | 9.78 | 19.69 |
| GE2 | 4.43 | 5.01 | **8.93** | **16.87** |
| SoG | 4.04 | 4.85 | 9.71 | 19.93 |
| gGW | 3.45 | 4.60 | 9.68 | 22.21 |
| GM [45] | **2.28** | **4.10** | 11.08 | 23.18 |
| BAY [14] | 3.44 | 4.70 | 10.21 | 24.47 |
| SVR [46] | 3.23 | – | – | 24.20 |
| AAS [16] | 3.16 | 4.18 | 9.15 | 22.21 |
| NIS [17] | 3.13 | 4.09 | 8.57 | 26.20 |
| SS ML [18] | 2.93 | 3.55 | 7.23 | 15.25 |
| SS GP [18] | 2.90 | 3.47 | 7.00 | **14.80** |
| AAC [16] | 2.90 | 3.74 | 7.93 | 14.98 |
| HLVI TD [19] | 2.63 | 3.65 | 7.53 | 25.24 |
| BP [47] | 2.61 | 3.98 | – | – |
| FB+SS GP [20] | 2.57 | 3.18 | 6.67 | **14.80** |
| HLVI BU [19] | 2.54 | 3.30 | 6.59 | 17.51 |
| HLVI BU&TD [19] | 2.47 | 3.38 | 6.97 | 25.24 |
| CCDL [30] | 2.30 | 3.10 | – | – |
| EB [22] | 2.24 | 2.77 | **5.52** | 19.44 |
| BDP [42] | 2.14 | 3.52 | – | 28.35 |
| CM [24] | 2.04 | 2.86 | – | – |
| FB+GM [20] | 2.01 | 3.67 | 9.50 | 23.18 |
| PCL [26] | 1.67 | 2.56 | 5.56 | – |
| SF [25] | 1.65 | 2.42 | – | – |
| CCP [31] | 1.47 | 2.16 | – | – |
| CCC [27] | **1.22** | **1.95** | – | – |
| AlexNet + SVR [4] | 3.09 | 4.74 | 11.18 | 29.15 |
| CNN per patch [4] | 2.69 | 3.67 | 7.79 | 30.93 |
| CNN average-pooling [4] | 2.44 | 3.18 | 6.37 | 14.84 |
| CNN median-pooling [4] | 2.32 | 3.07 | 6.15 | 19.04 |
| CNN fine-tuned [4] | 1.98 | 2.63 | **5.54** | **14.77** |
| CNN + SVR (this paper) | **1.44** | **2.36** | 5.72 | 16.98 |

as for other learning-based methods [49], our method appears to be mostly unaffected by the presence of image parts with different illuminants.

In Table III the median angular errors obtained by the considered state-of-the-art algorithms and the proposed approach on the NUS dataset are reported. As commonly done, results are reported separately for each camera and aggregated with geometric mean. From the results it is possible to notice that our method outperforms the other algorithms on all cameras with the exception of CCC [27].

### *B. Local Illuminant Estimation*

Our CNN predicts the illumination on small image patches, so it can be easily used to predict local illuminants as well as giving a global illuminant estimate for the entire image. Given the performance of the per patch error in Table II we expect our CNN to perform well even on local estimation. We perform here a preliminary test by using our learned CNN as-is on the synthetically relighted Geheler-Shi dataset.


Fig. 7. Examples of images on which the method makes the largest estimation errors, in the case of a single illuminant. Left to right: input RAW image, correction with the ground truth illuminant, correction with the illuminant estimated by the proposed method (with the local-to-global regressor enabled), and correction with the algorithm in the state-of-the-art making the best estimate on the given image.

Among the algorithms in the state-of-the-art able to deal with non-uniform illumination, e.g. [21], [23], [32], [33], [50] we report as comparison the results of the Multiple Light Sources (MLS) [34] using White Patch (WP) and Gray World (GW) algorithms, grid based sampling, in the clustering version setting the number of clusters equal to the number of lights in the scene; RETINEX [9], and Random Spray RETINEX [51] (in the light random spray version [52]); the Local Space Average Color (LSAC) [32]; fusion by Gradient Tree Boosting and fusion by Random Forest Regression [33]. The numerical results are reported in Table IV, while some examples are given in Figure 8. It is clear that the proposed method obtains significantly better results than all the other methods considered; the second best obtains more than twice the median error (5.92 degrees) than the proposed one (2.86 degrees).

Note that this comparison has been made by disabling the detection of multiple illuminant and by always taking the local estimates. In a further experiment we evaluated the performance in a mixed single/multi illuminant scenario. The dataset used is the single illuminant version of the Gehler-Shi and one-third of the synthetically relighted version so that the numbers of images having single and multiple illuminants are equal. The numerical results are reported in Table V, where the performance of the four variants of the proposed method are reported: i) single illuminant, that always applies the local regressorwith a support size equal to the whole image; ii) multi illuminant, that always keeps the local estimates; iii) the fully automatic, that uses the multiple illuminant detector to decide the size of the local regressor; iv) the oracle, that applies the local regressor with a support size taken from the ground truth, i.e. produces a global estimate when the image present a single illuminant. The results obtained show that the use of the multiple illuminant detector allows to obtain better results with respect to adopting a single strategy. Its performance are very close to those that can be obtained by exploiting the ground truth information about the presence of single or multiple illuminants (i.e. the oracle version).

TABLE III

MEDIAN ANGULAR ERRORS OBTAINED BY THE STATE-OF-THE-ART ALGORITHMS CONSIDERED ON THE NUS DATASET

| | GW | WP | SoG | gGW | BP | GE1 | GE2 | GM(P) | GM(E) | GM(I) | BAY | SS ML | SS GP | NIS | BDP | CCP | CCC | Prop. |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Canon1 | 4.15 | 6.19 | 2.73 | 2.35 | 2.45 | 2.48 | 2.44 | 4.30 | 4.68 | 4.72 | 2.80 | 2.80 | 2.67 | 3.04 | 2.01 | 2.18 | – | **1.71** |
| Canon2 | 2.88 | 12.44 | 2.58 | 2.28 | 2.48 | 2.07 | 2.29 | 14.83 | 15.92 | 14.72 | 2.35 | 2.32 | 2.03 | 2.46 | 1.89 | **1.75** | – | 1.85 |
| Fuji | 3.30 | 10.59 | 2.81 | 2.60 | 2.67 | 1.99 | 2.00 | 8.87 | 8.02 | 5.90 | 3.20 | 2.70 | 2.45 | 2.95 | 2.15 | 2.75 | – | **1.75** |
| Nikon1 | 3.39 | 11.67 | 2.56 | 2.31 | 2.30 | 2.22 | 2.19 | 10.32 | 12.24 | 9.24 | 3.10 | 2.43 | 2.26 | 2.40 | 2.08 | 2.00 | – | **1.88** |
| Oly | 2.58 | 9.50 | 2.42 | 2.15 | 2.18 | 2.11 | 2.18 | 4.39 | 8.55 | 4.11 | 2.81 | 2.24 | 2.21 | 2.17 | 1.87 | 2.22 | – | **1.65** |
| Pan | 3.06 | 18.00 | 2.30 | 2.23 | 2.15 | 2.16 | 2.04 | 4.74 | 4.85 | 4.23 | 2.41 | 2.28 | 2.22 | 2.28 | 2.02 | **1.53** | – | 1.59 |
| Sam | 3.00 | 12.99 | 2.33 | 2.57 | 2.49 | 2.23 | 2.32 | 7.91 | 6.12 | 6.37 | 3.00 | 2.51 | 2.29 | 2.77 | 2.03 | 1.65 | – | **1.88** |
| Sony | 3.46 | 7.44 | 2.94 | 2.56 | 2.62 | 2.58 | 2.70 | 4.26 | 3.30 | 3.81 | 2.36 | 2.70 | 2.58 | 2.88 | 2.33 | 3.11 | – | **1.63** |
| Nikon2 | 3.44 | 15.32 | 3.24 | 2.92 | 3.13 | 2.99 | 2.95 | 10.99 | 11.64 | 11.32 | 3.53 | 2.99 | 2.89 | 3.51 | 2.72 | – | – | **2.00** |
| Geo.mean | 3.22 | 11.03 | 2.64 | 2.43 | 2.48 | 2.30 | 2.33 | 7.09 | 7.46 | 6.40 | 2.81 | 2.54 | 2.39 | 2.69 | 2.11 | 2.09 | **1.48** | 1.76 |

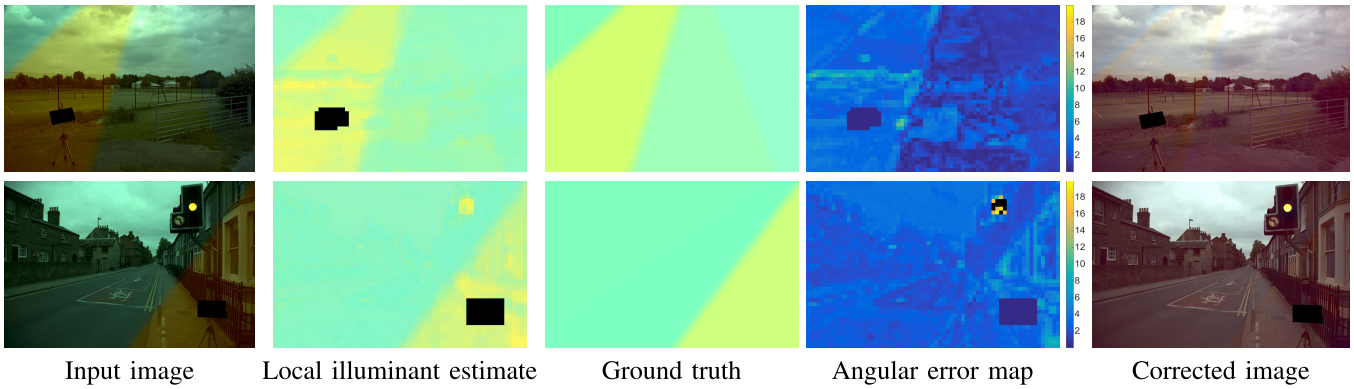



Fig. 8. Examples of the illuminant estimation on the relighted Gehler-Shi dataset. From left to right: relighted input image, local illuminant estimate, illuminant ground truth, angular error map between estimate and ground truth, corrected image.

TABLE IV

ANGULAR ERROR STATISTICS OBTAINED ON THE SYNTHETIC RELIGHTED GEHLER-SHI DATASET WITH SPATIALLY VARYING ILLUMINATION

| Algorithm | Med | Avg | $90^{th}$prc | Max |
|---|---|---|---|---|
| DN | 13.47 | 13.49 | 15.53 | 26.75 |
| LSAC [32] | 8.60 | 9.00 | 13.78 | 32.78 |
| RETINEX [9] | 8.61 | 9.03 | 13.75 | 32.76 |
| LRS RETINEX [52] | 7.54 | 7.98 | 12.46 | 31.44 |
| MLS+WP [34] | 5.92 | 6.90 | 12.55 | 31.09 |
| MLS+GW [34] | 8.91 | 9.35 | 14.43 | 33.33 |
| Fusion Grad. Tree Boost. [33] | 8.45 | 8.94 | 13.56 | 32.07 |
| Fusion Rand. Forest Regr. [33] | 5.96 | 6.84 | 12.39 | 31.23 |
| Proposed multiple estimate | 2.86 | 3.48 | 6.13 | 19.95 |

TABLE V

ANGULAR ERROR STATISTICS OBTAINED BY VARIANTS OF THE PROPOSED METHOD ON A MIXTURE OF THE ORIGINAL AND THE RELIGHTED GEHLER-SHI DATASET

| Algorithm | Med | Avg | $90^{th}$prc | Max |
|---|---|---|---|---|
| single illuminant | 2.73 | 3.54 | 7.62 | 23.61 |
| multi illuminant | 2.99 | 3.96 | 7.34 | 22.12 |
| fully automatic | 2.34 | 2.85 | 5.71 | 20.10 |
| oracle | 2.30 | 2.77 | 5.62 | 20.10 |

TABLE VI

ANGULAR ERROR STATS OBTAINED ON THE MILAN PORTRAIT DATASET

| Algorithm | Med | Avg | $90^{th}$prc | Max |
|---|---|---|---|---|
| DN | 17.30 | 17.53 | 19.74 | 28.60 |
| WP | 12.16 | 11.39 | 19.08 | 28.60 |
| GW | 4.26 | 4.86 | 9.26 | 20.04 |
| SoG | 4.39 | 5.93 | 14.03 | 20.02 |
| gGW | 5.25 | 6.42 | 15.07 | 20.80 |
| GE1 | 4.59 | 5.08 | 9.50 | 18.22 |
| GE2 | 4.93 | 5.39 | 9.69 | 15.36 |
| SS ML | 2.94 | 3.72 | 8.03 | 16.28 |
| LSAC [32] | 4.23 | 4.79 | 8.66 | 18.99 |
| RETINEX [9] | 4.28 | 4.83 | 8.39 | 20.54 |
| LRS RETINEX [52] | 3.97 | 4.26 | 7.69 | 18.03 |
| MLS + WP [34] | 3.21 | 4.04 | 7.55 | 17.19 |
| MLS + GW [34] | 3.33 | 4.18 | 8.82 | 17.97 |
| Fusion Grad. Tree Boost. [33] | 4.48 | 5.29 | 9.95 | 31.26 |
| Fusion Rand. Forest Regr. [33] | 3.23 | 3.96 | 7.61 | 27.76 |
| Face-based [21] | **2.11** | **2.66** | **5.15** | **11.43** |
| Proposed (fully automatic) | 2.63 | 3.14 | 6.09 | 15.02 |

The first experiment on real world data is performed on the subset of the Milan portrait dataset containing multiple MCCs. The numerical results are reported in Table VI, where the performance of the proposed method are reported enabling the multiple illuminant detector to decide the support size of the regressor. The results obtained show that the proposed method performs better than all the single illuminant estimation algorithms as well as all the general purpose multiple illuminant estimation ones. The only algorithm able to outperform the one proposed here is the face-based [21], which is specifically designed to leverage skin properties in images containing faces. An example taken from the Milan portrait dataset is reported in Figure 9. Since ground truth illuminant is available

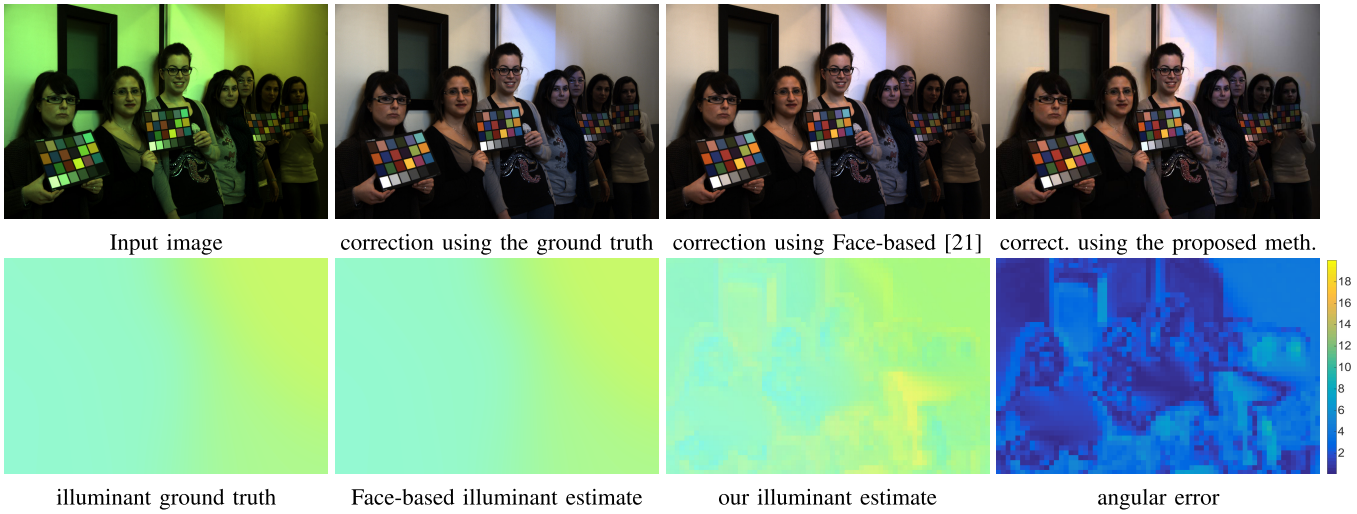



Fig. 9. Example image with multiple illuminants taken from the Milan portrait dataset.

TABLE VII

AVERAGE AND MEDIAN ANGULAR ERRORS OBTAINED ON THE TWO PARTS OF THE BEIGPUR ET AL. DATASET [43]: LABORATORY (LEFT) AND REAL-WORLD (RIGHT)

| Algorithm | Avg | Med | Avg | Med |
|---|---|---|---|---|
| DN | 10.6 | 10.5 | 8.8 | 8.9 |
| GW | 3.2 | 2.9 | 5.2 | 4.2 |
| WP | 7.8 | 7.6 | 6.8 | 5.6 |
| GE1 | 3.1 | 2.8 | 5.3 | 3.9 |
| GE2 | 3.2 | 2.9 | 6.0 | 4.7 |
| IEbv | 8.5 | 8.3 | 6.0 | 4.9 |
| LSAC [32] | 6.2 | 5.4 | 5.3 | 5.2 |
| RETINEX [9] | 6.3 | 5.4 | 5.2 | 5.2 |
| LRS RETINEX [52] | 5.8 | 4.8 | 4.6 | 4.0 |
| Fusion Grad. Tree Boost. [33] | 6.4 | 5.7 | 5.5 | 5.4 |
| Fusion Rand. Forest Regr. [33] | 5.0 | 3.9 | 4.1 | 3.5 |
| MLS + GW [34] | 6.4 | 5.9 | 4.4 | 4.3 |
| MLS + WP [34] | 5.1 | 4.2 | 4.2 | 3.8 |
| MLS + GE1 [34] | 4.8 | 4.2 | 9.1 | 9.2 |
| MLS + GE2 [34] | 5.9 | 5.7 | 12.4 | 12.4 |
| MIRF + GW [43] | 3.1 | 2.8 | 3.7 | 3.4 |
| MIRF + WP [43] | 3.0 | 2.8 | 4.1 | 3.3 |
| MIRF + GE1 [43] | 2.7 | 2.6 | 4.0 | 3.4 |
| MIRF + GE2 [43] | 2.6 | 2.6 | 4.9 | 4.5 |
| MIRF + IEbV [43] | 4.5 | 3.0 | 5.6 | 4.3 |
| Proposed (fully automatic) | **2.2** | **2.0** | **3.1** | **3.0** |

only on the MCCS, pixel-level ground truth is obtained by linear interpolation. As usual, MCCs are ignored during illuminant estimation but are left unmasked in the figure to better understand the results.

The last experiment concerning local illuminant estimation is performed on the multiple illuminant dataset by Beigpour et al. The numerical results are reported in Table VII, where the performance of the proposed method are reported enabling the multiple illuminant detector to decide the support size of the regressor. The results are reported separately for the laboratory and real-world settings. In both cases the results obtained show that the proposed method performs better than all the algorithms considered with an average reduction of the median error of more than 16%.

TABLE VIII

MEDIAN ANGULAR ERRORS OBTAINED BY THE PROPOSED CNN-BASED COLOR CONSTANCY ALGORITHM IN THE INTER-DATASET CROSS VALIDATION EXPERIMENT

| Train on: \ Test on: | Gehler-Shi | NUS | Milan Portait | Beigpour et al. |
|---|---|---|---|---|
| Gehler-Shi [14] [41] | **1.44** | 1.89 | 2.76 | 3.31 |
| NUS [42] | 1.58 | **1.76** | 2.82 | 3.14 |
| Milan Portrait [21] | 1.52 | 1.94 | **2.63** | 3.39 |
| Beigpour et al. [43] | 3.40 | 3.27 | 3.45 | **2.50** |

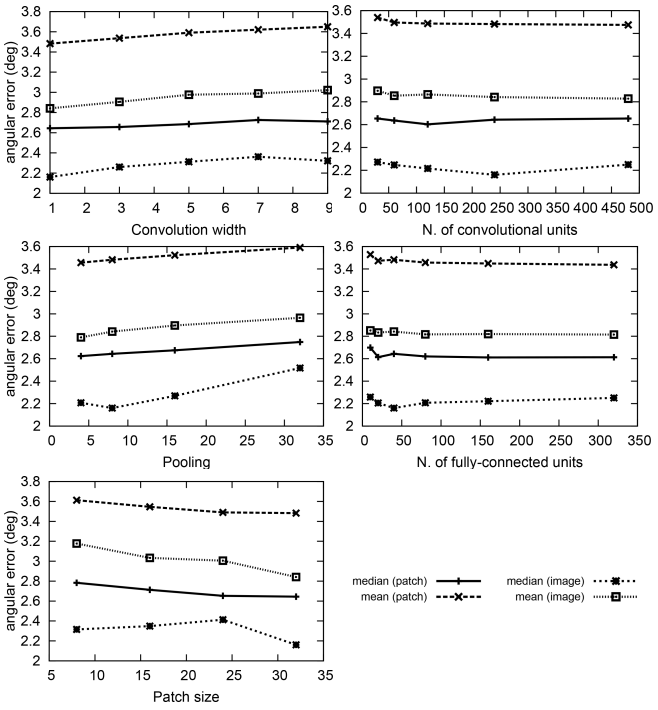



Fig. 10. Effect of the parameters on the CNN performance.

## C. Inter-Dataset Cross Validation

Although we have shown state-of-the-art performance for our CNN-based color constancy on the standard color constancy data sets studied, following [23] we also investigate

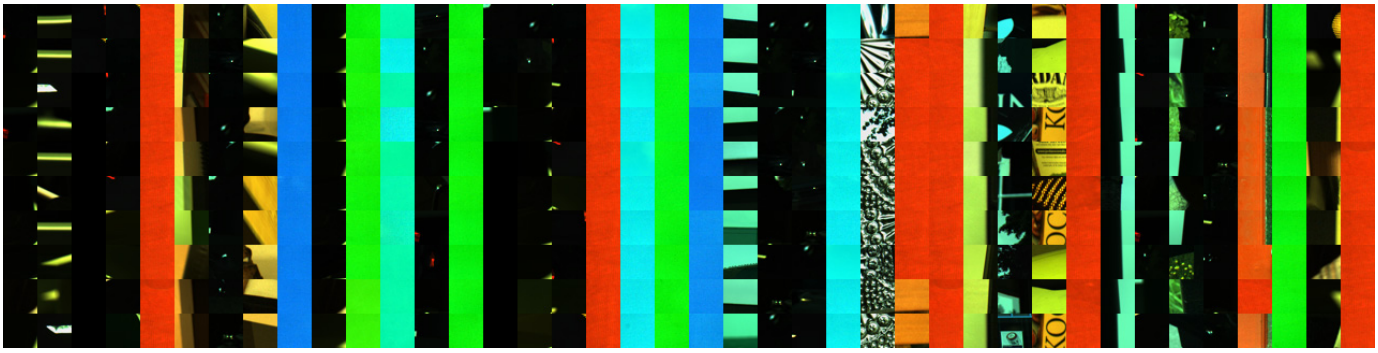


Fig. 11. Image patches producing the highest activations of the 40 neurons in the fully connected layer: each column represents a different neuron and reports in decreasing order the patches corresponding to its top ten activations.

here whether our proposed method also work well in an inter-dataset setup. This means that the model is learned on a dataset and tested as is on a different one. All the previous experiments were run doing cross-validation in an intra-dataset setup. However, images from the same dataset could be corretaled with each other to some degree due to the way image data is gathered. Therefore cross validating between different datasets is a more challenging experiment.

For this purpose, we run our proposed method, CNN-based color constancy, on all the four datasets considered by learning the model on one dataset and testing it on all the others. The median angular errors for the inter-dataset cross validation are reported in Table VIII. For completeness, the results for the intra-dataset experiment are also reported taking them from Table II, III, VI and VII where cross-validation on the same dataset was used. It is possible to see that for each dataset, the best performance is obtained when the model is learned on the same dataset used for testing. On average results for the inter-dataset cross validation experiment are only almost 0.6 degrees worse than the intra-dataset ones, still achieving state-of-the-art performance on all the four datasets considered.

### D. Computational Time

The average computational time of our method for one tested image is 208$ms$ on a PC with Intel i5-2500K 3.3GHz CPU, Nvidia k40 GPU, using Matlab 2014b including image read time, resizing, and patch by patch histogram stretching.

## VI. Network Architecture

In this section we discuss the design of the network, how its performance is affected by the parameters, and how we can relate the behavior of the learned model to that of other methods for computational color constancy.

The architecture of the network has been designed by starting from a deep CNN similar to the LeNet [53] and by removing layers until no further improvement in performance was possible. The final model is a simplified convolutional neural network with a single convolutional layer, max pooling, and two fully connected layers. Differently from other computer vision tasks, deepening the network causes slightly worse results. This fact probably depends on the small variability in content provided by the annotated data sets for computational color constancy. In fact, our training patches come from a few hundreds of images, while deep networks are often trained on millions of annotated images.

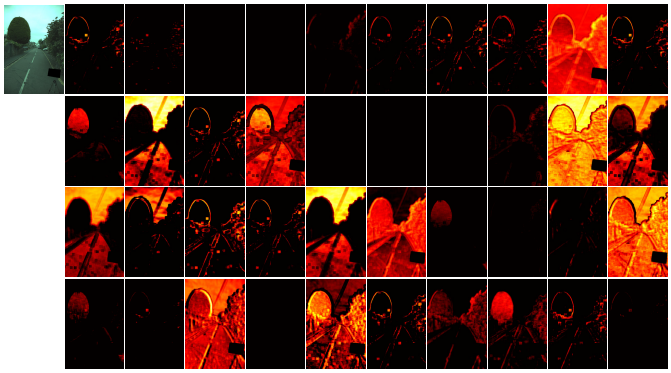


Fig. 12. Activation maps of the 40 neurons in the fully connected layer on all the patches of an entire image.

The performance of the network are quite robust with respect to its parameters as shown in Figure 10, that reports the variation in accuracy as a function of the size of the input patches, of the width and number of convolutional kernels, of the size of the receptive field of the pooling units, and of the number of fully connected units in the second to last layer. The plots are obtained by changing one parameter at a time while setting all the others as in the optimal configuration. In additional tests, not reported here, we also measured the performance obtained by varying multiple parameters without obtaining any surprising result.

The most striking element of the final network is the use of $1 \times 1$ "convolutional" units. At first this could be surprising, since in different domains larger kernels are preferred. However, it is not the first time that such small kernels are used, see [54]. In our case, networks built with larger convolutions failed to reproduce the spatial filters (edge detectors etc.) that are usually observed in CNNs trained for image classification. The number of the convolutional kernels seems less important and we found that the optimal value was around 240.

Another interesting element is represented by the relatively large ($8 \times 8$) receptive fields of the pooling units.

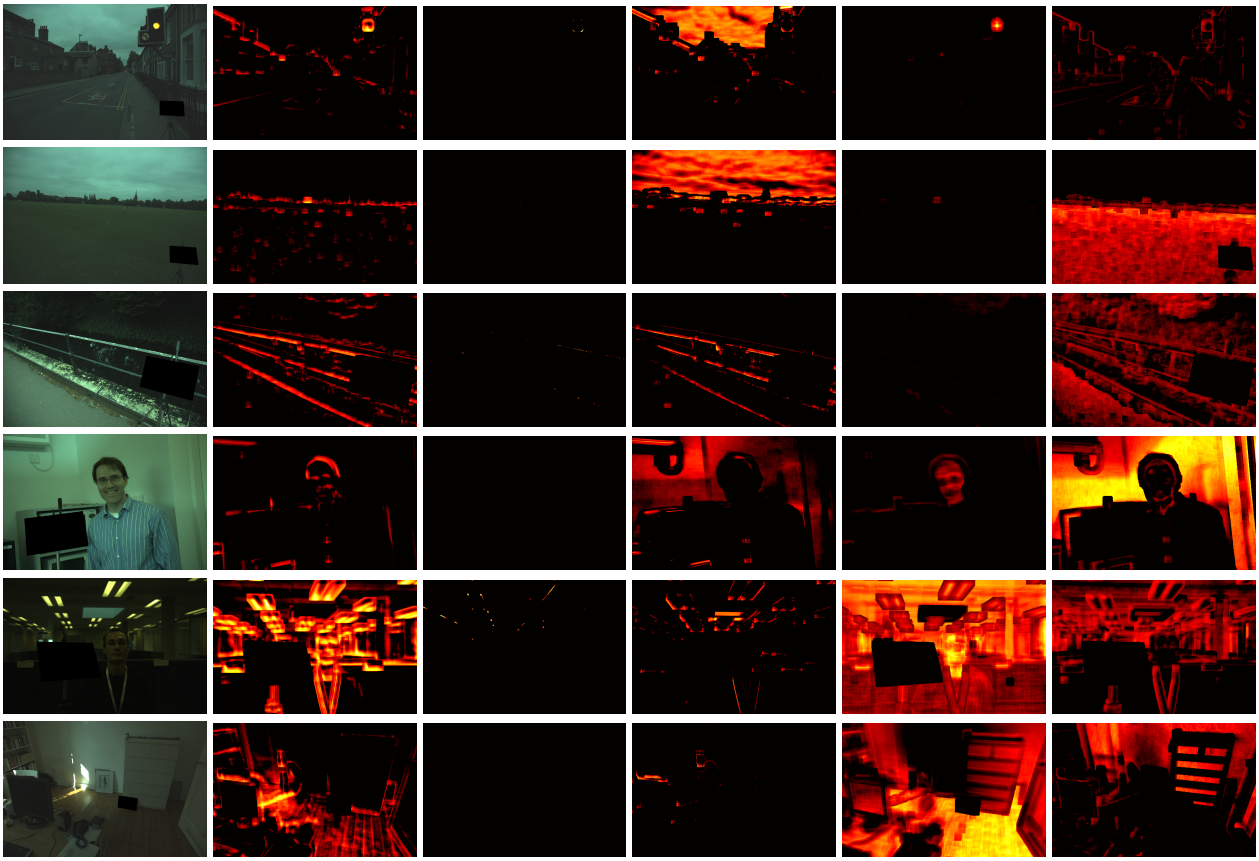



Fig. 13. Activation maps of five selected neurons in the fully connected layer: neuron 8, 17, 22, 27 and 38.

As a consequence the max pooling layer strongly reduces the dimensionality of the incoming data, while retaining just some spatial information. Smaller receptive fields resulted in a decrease in the performance of the network. We observe a sort of duality with respect to the parameters used for CNNs for image classification that usually prefer large convolutional kernels and small pooling units.

Concerning the remaining parameters, we found that the optimal number of fully-connected units was intermediate (40) and that the network prefers large $32 \times 32$ patches over smaller ones.

### A. Model Interpretation

After training the network, we analyzed the resulting weights for the three layers with learning capabilities. The last layer maps the 40 intermediate values ("features", in the following) in the three components of the illuminant estimated for the input patch. The transformation is affine and is represented by a matrix of $40 \times 3$ coefficients and by 3 biases. A layer of this kind has been already shown to perform well by Funt *et al.* [12], where it was used to process the responses of indicator functions over a regular quantization of the image chromaticities. It is also similar to combinational methods [55] where the outputs of different color constancy algorithms are combined to give the final illuminant estimate.

Differently from the work by Funt *et al.* [12], our network exploits some spatial information encoded in the 40 features that are computed as linear combinations of the 240 convolutions after that they have been pooled according to a $4 \times 4$ spatial grid. According to Cheng *et al.* [42], local spatial information does not provide any additional information that cannot be obtained directly from the color distributions. We found, instead, that some spatial information is still beneficial since it allows to detect useful patterns, as better discussed in the following. This also agrees with Gijsenij and Gevers [56] who noted that the use of spatial information brings an improvement over the application of color constancy to the entire image.

To better understand the role of the 40 features, we report in Figure 11 the ten patches producing their highest values. The patches are taken from the first fold of the Gehler-Shi dataset and are shown after the stretching of the color channels. It can be seen how different neurons are activated by different kinds of patches. Some of them are specialized in finding uniform patches of a given dominant color (blue, red, green…) that often correspond to specific content in the input images (sky, vegetation…). Several neurons are able to identify highlights, an element that has been previously exploited for color constancy [57]. There are also neurons specialized in detecting strong edges (that have also been used in the past [7]) and patches with complex textures. Figure 12 shows the

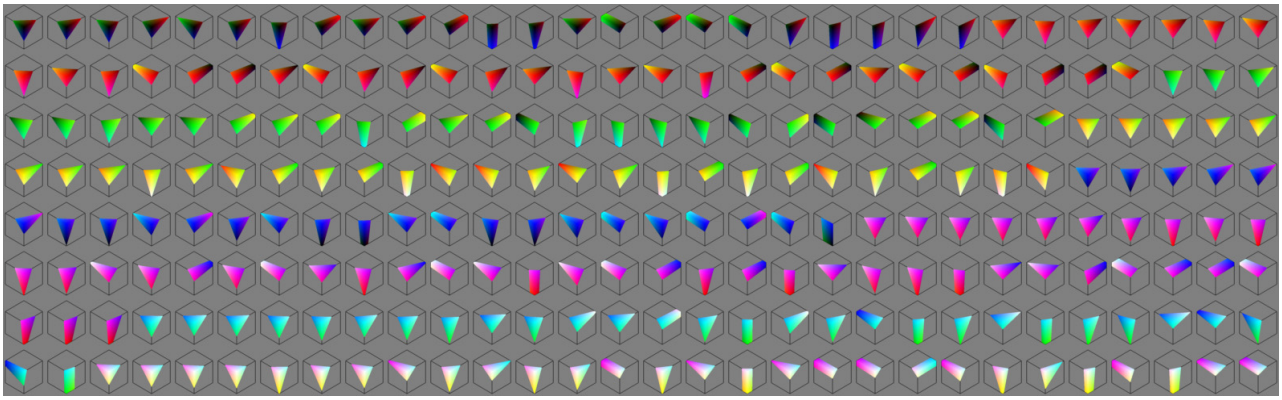


Fig. 14. Surface of the RGB cube resulting in the higher 10% responses of the 240 convolutional filters.

40 activations on the patches of a whole image, while those of five selected neurons on six different images are shown in Figure 13. These figures suggest that the network performs a rough analysis of the content of the image by identifying the main elements of the scene or by selecting elements that may be useful for the estimation of illuminant. For instance, neuron #8 seems to fire on image edges, neuron #17 on highlights, neuron #22 on sky and bluish texture, neuron #27 on skin and orange/reddish texture, neuron #38 on vegetation and greenish texture. The use of semantic concepts share some similarities with the work of van de Weijer *et al.* [19] where the illuminant is estimated by maximizing the likelihood of the colors associated to each semantic class.

Finally, the first layer is of the convolutional kind, and it consists of 240 units with $1 \times 1$ kernels. The activation of each convolutional unit can be seen as the projection over a specific direction in the RGB cube. Note that while $1 \times 1$ convolutions do not exploit spatial information, they also preserve it unaltered for the subsequent layers. The combination of the 240 units forms a sort of "soft" quantization of the color space that can be combined by the pooling units to represent the local color distribution. Figure 14 shows how different regions of the RGB color cube activate the 240 convolutional units. Since each unit corresponds to a linear projection, the maximum activation always occur on a vertex of the RGB cube (to improve visualization, cubes are rotated so that the region of maximum activation is always front-facing). It is possible to see that for all the eight vertexes there are several units with high activations. In practice this means that the quantization learned by the network covers the whole color space instead of being focused on specific colors. Several units seem redundant as they activate in presence of very similar colors. We observed, in fact, small differences in performance when we reduced the number of convolutional units (see Figure 10).

## VII. Conclusion

In this work we have developed a CNN-based color constancy algorithm that combines feature learning and regression as a complete optimization process, which enables us to employ modern training techniques to boost performance. The network has been specially designed to work on image patches in order to estimate the local illuminant color. Local estimates are then refined by local non-linear regression. The size of the support of the regression is automatically determined by a multiple illuminant detector: if only a single illuminnt is detected, the support size is the whole image and a global illuminant estimate is produced. The experimental results showed that our algorithm is the second best performing algorithm on images with a single illuminant. Experiments on a synthetically relighted dataset with multiple illuminants showed that our method outperforms all the general purpose local illuminant estimation methods in the state of the art. Results are further confirmed on two real-world datasets with multiple illuminants, where our method is outperformed only by an illuminant estimation method exploiting the presence of faces. The robustness of the proposed method is confirmed by inter-dataset cross validation. The results obtained suggest that a possible future research direction is that of feeding additional semantic information in the form of scene category or detected objects to further improve illuminant estimation performance.

## References


[1] D. Lee and K. N. Plataniotis, "A taxonomy of color constancy and invariance algorithm," in *Advances in Low-Level Color Image Processing*. Dordrecht, The Netherlands: Springer, 2014, pp. 55–94.

[2] K. Kavukcuoglu, P. Sermanet, Y.-L. Boureau, K. Gregor, M. Mathieu, and Y. L. Cun, "Learning convolutional feature hierarchies for visual recognition," in *Proc. NIPS*, 2010, pp. 1090–1098.

[3] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. NIPS*, 2012, pp. 1097–1105.

[4] S. Bianco, C. Cusano, and R. Schettini, "Color constancy using CNNs," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 81–89.

[5] S. D. Hordley and G. D. Finlayson, "Re-evaluating color constancy algorithms," in *Proc. ICPR*, Aug. 2004, pp. 76–79.

[6] B. Funt, K. Barnard, and L. Martin, "Is machine colour constancy good enough?" in *Proc. 5th Eur. Conf. Comput. Vis.*, 1998, pp. 445–459.

[7] J. van de Weijer, T. Gevers, and A. Gijsenij, "Edge-based color constancy," *IEEE Trans. Image Process.*, vol. 16, no. 9, pp. 2207–2214, Sep. 2007.

[8] G. Buchsbaum, "A spatial processor model for object colour perception," *J. Franklin Inst.*, vol. 310, pp. 1–26, Jul. 1980.

[9] E. H. Land *et al.*, "The retinex theory of color vision," *Sci. Amer.*, vol. 237, pp. 108–128, Dec. 1977.

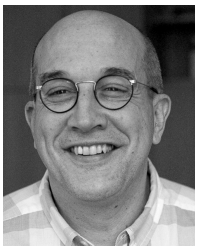
[10] D. A. Forsyth, "A novel algorithm for color constancy," *Int. J. Comput. Vis.*, vol. 5, no. 1, pp. 5–36, 1990.

[11] G. D. Finlayson, S. D. Hordley, and P. M. Hubel, "Color by correlation: A simple, unifying framework for color constancy," *IEEE Trans. Pattern Anal. Mach. Intell*, vol. 23, no. 11, pp. 1209–1221, Nov. 2001.

[12] B. Funt, V. Cardei, and K. Barnard, "Learning color constancy," in *Proc. Color Imag. Conf.*, Jan. 1996, pp. 58–60.
[13] R. Stanikunas, H. Vaitkevicius, and J. J. Kulikowski, "Investigation of color constancy with a neural network," *Neural Netw.*, vol. 17, no. 3, pp. 327–337, 2004.
[14] P. V. Gehler, C. Rother, A. Blake, T. Minka, and T. Sharp, "Bayesian color constancy revisited," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2008, pp. 1–8.
[15] S. Bianco, G. Ciocca, C. Cusano, and R. Schettini, "Improving color constancy using indoor–outdoor image classification," *IEEE Trans. Image Process.*, vol. 17, no. 12, pp. 2381–2392, Dec. 2008.
[16] S. Bianco, G. Ciocca, C. Cusano, and R. Schettini, "Automatic color constancy algorithm selection and combination," *Pattern Recognit.*, vol. 43, no. 3, pp. 695–705, 2010.
[17] A. Gijsenij and T. Gevers, "Color constancy using natural image statistics and scene semantics," *IEEE Trans. Pattern Anal. Mach. Intell*, vol. 33, no. 4, pp. 687–698, Apr. 2011.
[18] A. Chakrabarti, K. Hirakawa, and T. Zickler, "Color constancy with spatio-spectral statistics," *IEEE Trans. Pattern Anal. Mach. Intell*, vol. 34, no. 8, pp. 1509–1519, Aug. 2012, doi: 10.1109/TPAMI.2011.252.
[19] J. van de Weijer, C. Schmid, and J. Verbeek, "Using high-level visual information for color constancy," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2007, pp. 1–8.
[20] S. Bianco and R. Schettini, "Color constancy using faces," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 65–72.
[21] S. Bianco and R. Schettini, "Adaptive color constancy using faces," *IEEE Trans. Pattern Anal. Mach. Intell*, vol. 36, no. 8, pp. 1505–1518, Aug. 2014.
[22] H. R. V. Joze and M. S. Drew, "Exemplar-based colour constancy," in *Proc. BMVC*, 2012, pp. 1–12.
[23] H. R. V. Joze and M. S. Drew, "Exemplar-based color constancy and multiple illumination," *IEEE Trans. Pattern Anal. Mach. Intell*, vol. 36, no. 5, pp. 860–873, May 2014.
[24] G. D. Finlayson, "Corrected-moment illuminant estimation," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 1904–1911.
[25] D. Cheng, B. Price, S. Cohen, and M. S. Brown, "Effective learning-based illuminant estimation using simple features," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 1000–1008.
[26] A. Chakrabarti, "Color constancy by learning to predict chromaticity from luminance," in *Proc. NIPS*, 2015, pp. 163–171.
[27] J. T. Barron, "Convolutional color constancy," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 379–387.
[28] Y. Jia *et al.*, "Caffe: Convolutional architecture for fast feature embedding," in *Proc. ACM Int. Conf. Multimedia*, 2014, pp. 675–678.
[29] H. Drucker, C. J. Burges, L. Kaufman, A. Kaufman, and V. Kaufman, "Support vector regression machines," in *Proc. NIPS*, vol. 9. 1997, pp. 155–161.
[30] Z. Lou, T. Gevers, N. Hu, and M. P. Lucassen, "Color constancy by deep learning," in *Proc. Brit. Mach. Vis. Conf.*, Sep. 2015, pp. 76.1–76.12.
[31] S. W. Oh and S. J. Kim, "Approaching the computational color constancy as a classification problem through deep learning," *Pattern Recognit.*, vol. 61, pp. 405–416, Jan. 2017.
[32] M. Ebner, "Color constancy based on local space average color," *Mach. Vis. Appl.*, vol. 20, no. 5, pp. 283–301, 2009.
[33] M. Bleier *et al.*, "Color constancy and non-uniform illumination: Can existing algorithms work?" in *Proc. ICCV Workshop*, Nov. 2011, pp. 774–781.
[34] A. Gijsenij, R. Lu, and T. Gevers, "Color constancy for multiple light sources," *IEEE Trans. Image Process.*, vol. 21, no. 2, pp. 697–707, Feb. 2012.
[35] E. Hsu, T. Mertens, S. Paris, S. Avidan, and F. Durand, "Light mixture estimation for spatially varying white balance," *ACM Trans. Graph.*, vol. 27, no. 3, p. 70, 2008.
[36] I. Boyadzhiev, K. Bala, S. Paris, and F. Durand, "User-guided white balance for mixed lighting conditions," *ACM Trans. Graph.*, vol. 31, no. 6, p. 200, 2012.
[37] Z. I. Botev, J. F. Grotowski, and D. P. Kroese, "Kernel density estimation via diffusion," *Ann. Statist.*, vol. 38, no. 5, pp. 2916–2957, 2010.
[38] A. Witkin, "Scale-space filtering: A new approach to multi-scale description," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, vol. 9. Mar. 1984, pp. 150–153.
[39] S. D. Hordley, "Scene illuminant estimation: Past, present, and future," *Color Res. Appl.*, vol. 31, no. 4, pp. 303–314, Aug. 2006.
[40] A. Rosenfeld and J. L. Pfaltz, "Sequential operations in digital picture processing," *J. ACM*, vol. 13, no. 4, pp. 471–494, 1966.
[41] L. Shi and B. V. Funt. *Re-Processed Version of the Gehler Color Constancy Database of 568 Images*, accessed on Apr. 26, 2016. [Online]. Available: http://www.cs.sfu.ca/~colour/data/shi_gehler/
[42] D. Cheng, D. K. Prasad, and M. S. Brown, "Illuminant estimation for color constancy: Why spatial-domain methods work and the role of the color distribution," *J. Opt. Soc. Amer. A, Opt. Image Sci.*, vol. 31, no. 5, pp. 1049–1058, 2014.
[43] S. Beigpour, C. Riess, J. van de Weijer, and E. Angelopoulou, "Multi-illuminant estimation with conditional random fields," *IEEE Trans. Image Process.*, vol. 23, no. 1, pp. 83–96, Jan. 2014.
[44] A. Gijsenij, T. Gevers, and J. van de Weijer, "Computational color constancy: Survey and experiments," *IEEE Trans. Image Process.*, vol. 20, no. 9, pp. 2475–2489, Sep. 2011.
[45] A. Gijsenij, T. Gevers, and J. van de Weijer, "Generalized gamut mapping using image derivative structures for color constancy," *Int. J. Comput. Vis.*, vol. 86, nos. 2–3, pp. 127–139, Jan. 2010.
[46] B. Funt and W. Xiong, "Estimating illumination chromaticity via support vector regression," in *Proc. Color Imag. Conf.*, Jan. 2004, pp. 47–52.
[47] H. R. V. Joze, M. S. Drew, G. D. Finlayson, and P. A. T. Rey, "The role of bright pixels in illumination estimation," in *Proc. Color Imag. Conf.*, Jan. 2012, pp. 41–46.
[48] J. von Kries, "Chromatic adaptation," *Selection Translated and Reprinted in Sources Of Color Science*, D. L. MacAdam, Ed. Cambridge, MA, USA: MIT Press, 1970, pp. 109–119.
[49] L. Xu and B. Funt, "How multi-illuminant scenes affect automatic colour balancing," in *Proc. Color Image*, 2015, pp. 62–67.
[50] E. Provenzi, C. Gatta, M. Fierro, and A. Rizzi, "A spatially variant white-patch and gray-world method for color image enhancement driven by local contrast," *IEEE Trans. Pattern Anal. Mach. Intell*, vol. 30, no. 10, pp. 1757–1770, Oct. 2008.
[51] E. Provenzi, M. Fierro, A. Rizzi, L. De Carli, D. Gadia, and D. Marini, "Random spray retinex: A new retinex implementation to investigate the local properties of the model," *IEEE Trans. Image Process.*, vol. 16, no. 1, pp. 162–171, Jan. 2007.
[52] N. Banić and S. Lončarić, "Light random sprays retinex: Exploiting the noisy illumination estimation," *IEEE Signal Process. Lett.*, vol. 20, no. 12, pp. 1240–1243, Dec. 2013.
[53] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.
[54] C. Szegedy *et al.* (2014). "Going deeper with convolutions." [Online]. Available: https://arxiv.org/abs/1409.4842
[55] B. Li, W. Xiong, W. Hu, and B. Funt, "Evaluating combinational illumination estimation methods on real-world images," *IEEE Trans. Image Process.*, vol. 23, no. 3, pp. 1194–1209, Mar. 2014.
[56] A. Gijsenij and T. Gevers, "Color constancy using image regions," in *Proc. ICIP*, vol. 3. Sep./Oct. 2007, pp. III-501–III-504.
[57] H.-C. Lee, "Method for computing the scene-illuminant chromaticity from specular highlights," *J. Opt. Soc. Amer. A, Opt. Image Sci.*, vol. 3, no. 10, pp. 1694–1699, 1986.

**Simone Bianco** received the B.Sc. and M.Sc. degrees in mathematics from the University of Milano-Bicocca, Italy, in 2003 and 2006, respectively, and the Ph.D. degree in computer science from the Dipartimento di Informatica, Sistemistica e Comunicazione, University of Milano-Bicocca, in 2010. He is currently an Assistant Professor. His current research interests include computer vision, machine learning, optimization algorithms, and color imaging.

**Claudio Cusano** received the Ph.D. degree from the University of Milano-Bicocca, Italy, in 2006. He has been a Researcher with a grant at the ITC Institute of the Italian National Research Council and at the Imaging and Vision Laboratory, University of Milano-Bicocca. He is an Associate Professor with the Department of Electrical, Computer, and Biomedical Engineering, University of Pavia. His research interests include 2-D and 3-D imaging, with a particular focus on image analysis and classification.



**Raimondo Schettini** is a Full Professor with the University of Milano-Bicocca, Italy. He is the Vice Director of the Department of Informatics, Systems and Communication and the Head of the Imaging and Vision Laboratory. He has been associated with the Italian National Research Council, since 1987, where he has led the Color Imaging Laboratory, from 1990 to 2002. He has been the Team Leader on several research projects. He has authored or co-authored over 300 refereed papers and holds several patents on color reproduction, 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.