IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received January 21, 2021, accepted February 1, 2021, date of publication February 4, 2021, date of current version February 17, 2021.
Digital Object Identifier 10.1109/ACCESS.2021.3057072

Iterative Convolutional Neural Network-Based Illumination Estimation

KARLO KOŠČEVIĆ, (Graduate Student Member, IEEE), MARKO SUBAŠIĆ, (Member, IEEE), AND SVEN LONČARIĆ, (Senior Member, IEEE)
Faculty of Electrical Engineering and Computing, University of Zagreb, 10000 Zagreb, Croatia
Corresponding author: Karlo Koščević (karlo.koscevic@fer.hr)

This work was supported by the Croatian Science Foundation under Project IP-06-2016-2092.

ABSTRACT In the image processing pipelines of digital cameras, one of the first steps is to achieve invariance in terms of scene illumination, namely computational color constancy. Usually, this is done in two successive steps which are illumination estimation and chromatic adaptation. The illumination estimation aims at estimating a three-dimensional vector from image pixels. This vector represents the scene illumination, and it is used in the chromatic adaptation step, which aims at eliminating the bias in image colors caused by the color of the illumination. An accurate illumination estimation is crucial for successful computational color constancy. However, this is an ill-posed problem, and many methods try to comprehend it with different assumptions. In this paper, an iterative method for estimating the scene illumination color is proposed. The method calculates the illumination vector by a series of intermediate illumination estimations and chromatic adaptations of an input image using a convolutional neural network. The network has been trained to iteratively compute intermediate incremental illumination estimates from the original image. Incremental illumination estimates are combined by per element multiplication to obtain the final illumination estimation. The approach is aimed to reduce large estimation errors usually occurring with highly saturated light sources. Experimental results show that the proposed method outperforms the vast majority of illumination estimation methods in terms of median angular error. Moreover, in terms of worst-performing samples, i.e., the samples for which a method errs the most, the proposed method outperforms all other methods by a margin of more than 18% with respect to the mean of estimation errors in the third quartile.

INDEX TERMS Chromatic adaptation, color constancy, convolutional neural networks, illumination estimation, image color analysis.

I. INTRODUCTION

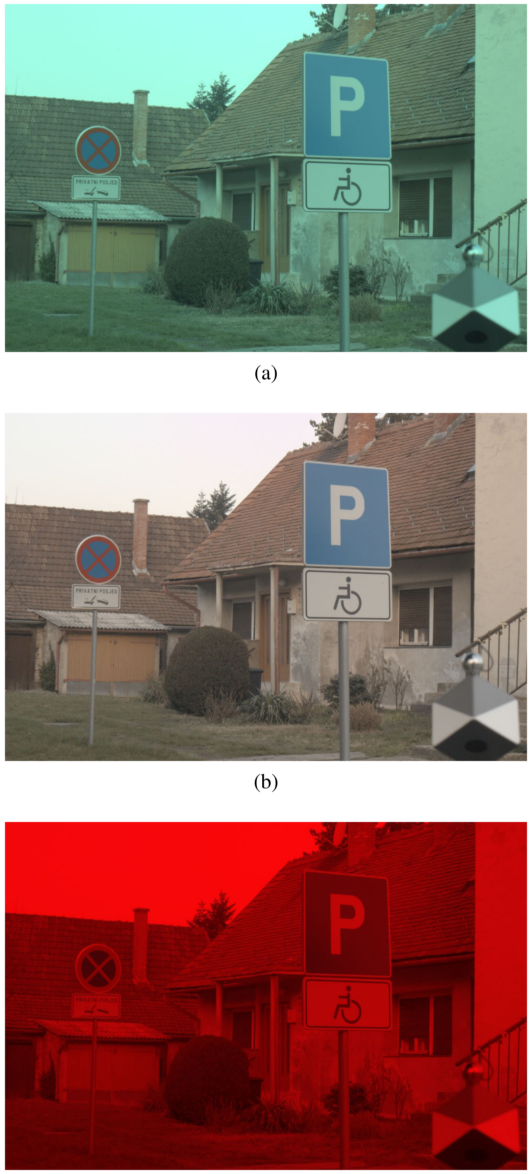
In digital photography, any illumination present in the scene of interest significantly impacts the colors of the objects in digital images. According to the image formation model [1], the value of a pixel in an image is determined by three functions: the spectrum of the light source, the reflectance of the object surface, and the spectral sensitivity of the camera sensor. If the same scene is captured with the same camera (i.e., the reflectance of the object surface and the spectral sensitivity of the camera sensor are constant) whereas the spectrum of the light source changes, the colors in the captured images will most likely differ. The reason for this behavior is that the camera sensor is a device that can only capture the incident light but cannot detect changes in illumination itself. Therefore, for most digital cameras, one of the first steps in the image processing pipeline is dedicated to achieving illumination invariance. This process can be associated with the ability of the human visual system to adapt to changes in scene illumination, namely color constancy [2]. Achieving computational color constancy has proven to be beneficial in many image-related areas such as object recognition, scene comprehension, digital photography, and image reproduction [3]. In order to achieve computational color constancy, two steps are usually required. First, the scene illumination color is estimated based on the image pixel values, and then, in the second step, its influence on the image colors is eliminated. Color constancy is not yet fully understood and

The associate editor coordinating the review of this manuscript and approving it for publication was Shiqi Wang.

modeled, and estimating the scene illumination from the image pixels is an ill-posed problem, which is regularized by various assumptions. During the last few years, many methods for estimating the illumination color have been proposed, with the general assumption that the illumination is uniform in the scene [1]. Since only one illumination vector per image is estimated, a simple diagonal matrix with reciprocal illumination values on the diagonal is usually used to eliminate color distortion.

For a successful computational color constancy, both illumination estimation and chromatic adaptation should be as accurate and similar to the image formation model as possible. However, even though the simple diagonal matrix for chromatic adaptation is computationally efficient and sufficient for a somewhat satisfactory computational color constancy, it is still an approximation. Illumination estimates can be either imprecise or out of the range of illuminations for which color images can be properly corrected using the current chromatic adaptation model. It is expected that the error in computational color constancy is higher for images that are captured in scenes illuminated with highly colored light sources than for scenes affected by near-white illuminations. Such illuminations can corrupt object colors, and if their estimates are imprecise high errors in corrected images can be expected. In [4], it was shown that camera manufacturers bound illuminations to a narrow region in chromaticity space so that chromatic adaptation is never performed with highly colored illuminations. It can be speculated that the cause for this is the inadequacy of the chromatic adaptation model that is unfit for the highly colored illuminations. Therefore, in this paper, a multistage illumination color estimation combined with the current simple chromatic adaptation model is proposed. The individual stages' estimations are restricted from highly colored estimations so that the used chromatic adaptation model is operating in the range of slightly colored illuminations. The final illumination estimation is obtained by combining all of the stage illuminations so that the final illumination estimations can still be highly colored. With this approach, the occurrence of high estimation errors should be alleviated, as shown in experimental results.

For the evaluation of illumination estimation methods, the angular error is used. It is calculated as the angle between the ground-truth illumination vector and the estimated illumination vector. Usually, the RGB color space is used so that both vectors have three components corresponding to the red, green, and blue image channels. The median error value of a test dataset is usually considered the most representative statistic. Nowadays, illumination estimation methods can achieve median error values of less than 2°, which can be regarded as a threshold for a sufficiently accurate illumination estimation [5]. However, even such accurate methods in terms of median or mean error value tend to be flawed in some cases. The maximum error values can be as large as 10° or more. Correcting an image with a highly incorrect illumination color vector can distort the image colors to such an extent that the actual information they carry is effectively lost.



(a)

(b)

(c)

FIGURE 1. Chromatic adaptation example with highly inaccurate illumination vector: (a) original raw image with the influence of illumination; (b) the result of the chromatic adaptation of image (a) with ground-truth illumination vector $(0.1624\ 0.4533\ 0.3843)^T$; (c) the result of the chromatic adaptation of image (a) with inaccurate illumination vector $(0.0001\ 0.6528\ 0.3471)^T$. The angle between the ground-truth vector and inaccurate illumination vector is 19.54°. For display purposes, images were tone mapped by using the Flash tone mapping operator [6].

An example of a chromatic adaptation with a highly incorrect illumination vector is shown in Fig. 1.

In this paper, an illumination estimation method that reduces maximum estimation errors, which can occur when highly colored illuminations are present in the scene, is proposed. The proposed method combines both illumination estimation and chromatic adaptation, which are usually two distinct steps in the image processing pipeline, to obtain more precise illumination estimates. The global illumination vector is estimated through a series of consecutive intermediate illumination estimations, and chromatic adaptations of an input image. In each step, intermediate illumination estimation is forced to a subset of illuminations that are close to the white light, i.e., a light that does not alter image colors. Chromatic adaptation of the input image with an estimated intermediate illumination vector is performed, and such a corrected image is then passed as a new input. This procedure was embedded in a deep neural network which uses convolutional architecture for the estimation of intermediate illuminations, and simple matrix multiplications for chromatic adaptation of input images and aggregation of intermediate estimates into one final illumination estimate.

The rest of the paper is structured as follows: In Section II, an overview of related methodology is given, Section III describes the proposed illumination estimation method, experimental results are presented and discussed in Section IV, and in Section V, the conclusion is given.

II. RELATED WORK

The image formation model, commonly used in computational color constancy, which assumes Lambertian reflectance can be formulated as

$$f_c(\mathbf{x}) = \int_{\omega} I(\lambda, \mathbf{x}) R(\lambda, \mathbf{x}) \rho_c(\lambda) d\lambda, \tag{1}$$

where each pixel $\mathbf{x}$ in the image $\mathbf{f}$ with three color channels $c \in \{R, G, B\}$ is computed as the integral of the product of light source spectrum $I(\lambda, \mathbf{x})$, surface reflectance $R(\lambda, \mathbf{x})$, and camera sensor sensitivity $\rho_c(\lambda)$ across all wavelengths λ in the visible light spectrum ω.

A. ILLUMINATION ESTIMATION

The first step in computational color constancy is illumination estimation, which aims to estimate the vector of the scene illumination from image pixels. From (1), it can be observed that illumination can be determined by knowing the light source spectrum $I(\lambda, \mathbf{x})$ and camera sensor sensitivity $\boldsymbol{\rho}(\lambda)$. In the case of global illumination estimation methods, i.e., when it is assumed that there is one dominant light source present in the scene, spatial information $\mathbf{x}$ is disregarded, and the illumination vector is defined as

$$\mathbf{e} = \begin{pmatrix} e_R \\ e_G \\ e_B \end{pmatrix} = \int_{\omega} I(\lambda) \boldsymbol{\rho}(\lambda) d\lambda. \tag{2}$$

The estimation of $\mathbf{e}$ is an ill-posed problem as usually there is no prior knowledge about $I(\lambda)$ and $\boldsymbol{\rho}(\lambda)$ values.

To make the problem of illumination estimation feasible, illumination estimation methods are often based on some assumptions. One group of illumination estimation methods are methods such as White-Patch [7], [8] and its improvements [9]–[11], and gray world assumption-based methods that include Gray-World [12], Shades-of-Gray [13], Gray-Edge [14], Weighted-Gray-Edge [15]. Although simple and do not generalize well, these methods are suitable for hardware implementation since they use simple image features and statistics, which are fast to calculate and have insignificant computational complexity.

On the other hand, there are machine-learning based illumination estimation methods that require computational models to be trained on data. The most recent examples are methods based on deep learning. These methods achieve the most accurate estimates of scene illumination but are highly dependent on training data distribution. Large and diverse datasets are prerequisites for creating deep learning methods that can generalize well. In comparison with illumination estimation methods in the first group, learning-based methods require more computational resources and have more complex structures. The earliest deep learning architectures for illumination estimation were very shallow, containing only a few convolutional and fully connected layers [16], [17]. Content-based convolutional neural networks that combine weighted local illumination estimations have been proposed in [18]–[20]. In [21], [22], illumination estimation was cast into a deep learning classification problem. In [23], from an image, two illuminations were estimated using one convolutional neural network, and then using another convolutional neural network, a more probable one was chosen. The problem of dependency of illumination estimation methods on the camera sensor was tackled in [24], where two convolutional networks were used for sensor space mapping and illumination estimation, respectively. Other learning-based methods use Bayesian learning [25], color moments [26], gamut mapping [27]–[29], spatial localizations [30], [31], visual information of high level [32], illumination space restrictions [4], [33]–[35], gray pixel detection [36], regression trees with simple color features [37], and others.

B. CHROMATIC ADAPTATION

The second step in computational color constancy is chromatic adaptation, which is used to change the color cast in images due to the illumination color. It was shown that using a diagonal matrix can be sufficient for a successful chromatic adaptation [38]. Namely, following this simplification, which is also known as the *von Kries model* [39], camera sensor responses are considered independent. Then, for an image pixel $\mathbf{p} = \begin{pmatrix} p_R \; p_G \; p_B \end{pmatrix}^T$, a new color corrected pixel $\hat{\mathbf{p}} = \begin{pmatrix} \hat{p}_R \; \hat{p}_G \; \hat{p}_B \end{pmatrix}^T$ can be computed as

$$\hat{\mathbf{p}} = \mathbf{C}\mathbf{p}, \tag{3}$$

where $\mathbf{C}$ denotes the correction matrix. In general, the correction matrix $\mathbf{C}$ can be computed as

$$\mathbf{C} = \begin{pmatrix} \bar{e}_R/e_R & 0 & 0 \\ 0 & \bar{e}_G/e_G & 0 \\ 0 & 0 & \bar{e}_B/e_B \end{pmatrix}, \tag{4}$$

where $\mathbf{e} = \begin{pmatrix} e_R \; e_G \; e_B \end{pmatrix}^T$ denotes the illumination vector that should be removed from an image, and $\bar{\mathbf{e}} = \begin{pmatrix} \bar{e}_R \; \bar{e}_G \; \bar{e}_B \end{pmatrix}^T$ denotes the vector of the desired illumination. In computational color constancy, the input image should be processed so that it appears as it was captured while illuminated with a white light source, i.e., the light source for which $e_R = e_G = e_B$. Therefore, $\bar{\mathbf{e}} = \begin{pmatrix}1 & 1 & 1\end{pmatrix}^T$ is used.

III. PROPOSED METHOD

The proposed method estimates the illumination vector from a raw input image in multiple iterations. In each iteration, a restricted intermediate illumination vector is computed from the input image. The estimated vector is then used for chromatic adaptation of the input image according to (3). In the next iteration, the corrected image is used as input. In the end, intermediate illumination vectors estimated in the iterations are element-wise multiplied to produce the final illumination vector that corresponds to the scene illumination captured in the original raw input image. The pseudocode of the proposed illumination estimation method is given in Algorithm 1.

Algorithm 1 Iterative Illumination Estimation

Input: image $\mathbb{I}$, convolutional neural network CNN, iteration number N

Output: illumination vector $\mathbf{e}$

1: $\mathbf{e} \leftarrow \begin{pmatrix}e_R & e_G & e_B\end{pmatrix} \leftarrow \begin{pmatrix}1 & 1 & 1\end{pmatrix}$
2: **for** $k \leftarrow 1$ to N **do**
3: $\quad \mathbf{e}^{(k)} \leftarrow CNN.estimate(\mathbb{I})$
4: $\quad \mathbf{e} \leftarrow \mathbf{e} \circ \mathbf{e}^{(k)}$
5: $\quad \mathbf{C} \leftarrow \mathrm{diag}(1/e_R^{(k)}, 1/e_G^{(k)}, 1/e_B^{(k)})$ ▷ Eq. (4)
6: $\quad \mathbb{I}_{x,y} \leftarrow \mathbf{C}\mathbb{I}_{x,y}\ \forall x, y$ ▷ Eq. (3)
7: $\quad \mathbb{I} \leftarrow \frac{1}{\max_{x,y} \mathbb{I}} \cdot \mathbb{I}$
8: **end for**
9: $\mathbf{e} \leftarrow \frac{1}{e_R + e_G + e_B} \cdot \mathbf{e}$

In each iteration, an intermediate illumination vector is estimated using the convolutional neural network. Network parameters are the same in each iteration. Convolutional blocks of the VGG16 [40] network architecture were used as a feature extractor,[1] on top of which one additional convolutional layer was placed. This layer has three filters with a kernel of size 1×1. Each filter corresponds to one of three color channels in the RGB image: red, green, and blue. Output activation was a sigmoid function. Global average pooling, which calculates the average across feature maps, was used to accumulate feature maps computed by the last convolutional layer, thus producing one value for each color channel. Global average pooling yields the intermediate illumination vector. On top of this, chromatic adaptation was implemented, which uses the current network input and illumination estimate to compute the network input in the next iteration.

A. DATA NORMALIZATION

The last convolutional layer in the proposed network architecture uses a sigmoid activation function that ensures that intermediate illumination estimates are all in the first octant in three-dimensional illumination solution space. However, the codomain of a sigmoid function is in the range from zero to one. When such values are used for chromatic adaptation, due to the division, the values in the corrected image may span in a different range than original image values. Therefore, in each iteration, the input image is normalized by dividing every image value by the image maximum. Moreover, input normalization was shown beneficial for efficient backpropagation [43].

Estimated intermediate illumination vectors in each iteration were not normalized using the standard normalization in computational color constancy research, i.e., the division of illumination vector with its sum. The reasoning behind this is that the proposed method combines illumination estimation, chromatic adaptation, and the abovementioned image normalization. Namely, if chromatic adaptation is performed with normalized illumination vector and the resulting image is then normalized as well, the factor which would be used to normalize the illumination would be canceled out. Therefore, normalizing intermediate illumination vectors would not have any effect.

B. NETWORK TRAINING

For the training of the proposed illumination estimation network architecture, a custom loss function was used. It is based on the cosine of the angle[2] between two vectors and consists of two parts. The first part of the custom loss function is dedicated to computing the error between ground-truth illuminations and the end-result of the network. The second part is used to control the behavior of intermediate illumination estimates in each iteration by forcing them to be close to the white light. This is achieved by minimizing the angle between intermediate illumination estimates and the vector of the white light. However, the extent of bounding to the white light is not the same in each iteration. With each subsequent iteration, intermediate illuminations have to be closer to white light. That is achieved by assigning the weight to the loss value in each iteration as

$$w_k = \frac{2^{k-1}}{\sum_{j=0}^{N-1} 2^j}, \tag{5}$$

where $k \in \{1, \ldots, N\}$ denotes the current iteration, and N denotes the number of iterations.

[1] It was experimentally determined to use the VGG16 network as a feature extractor. The architecture of SqueezeNet [41] convolutional neural network was also considered, which matches the accuracy of AlexNet [42] architecture but with fewer weights. However, the VGG16 network outperformed such simpler architectures.

[2] The most direct measure of error in illumination estimation is the angle between the ground-truth illumination value and the estimated illumination value. Taking into account that both the ground-truth and the estimation are vectors, the angle between them, once they are both normalized to unit length, is computed as the inverse cosine ($\cos^{-1}$) of their dot product. According to [44], using $\cos^{-1}$ makes the derivative of the loss function more complex and infinite when the absolute value of the dot product is equal to one, and therefore, using $1 - \cos\theta$ as loss function is more appropriate.

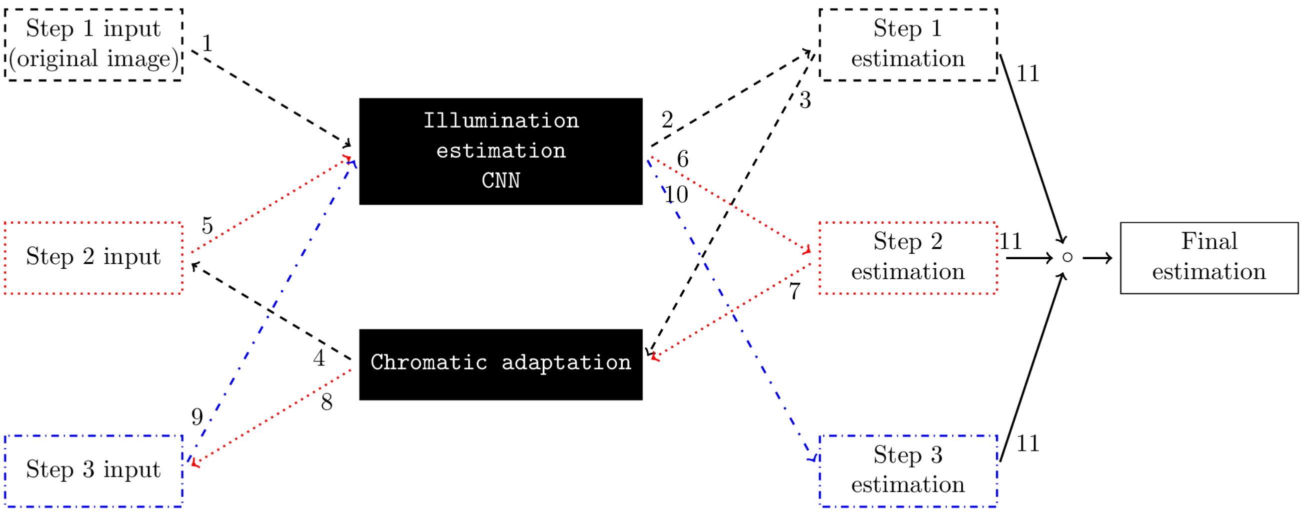


FIGURE 2. The illustration of the forward pass of the proposed method for three iterations. Arrows are enumerated in the order of execution, starting from 1. Different line styles denote different iterations: – – – denotes the first iteration steps, ··· denotes the second iteration steps, and – · – denotes the third iteration steps. The final estimation in step 11 is computed in parallel once the last iteration ends.

In both parts of the loss function, for a mini-batch of M input samples, the loss L was calculated as

$$\mathrm{L}(\mathbb{E}, \hat{\mathbb{E}}) = \frac{1}{M}\sum_{m=1}^{M}\left(1 - \frac{\mathbb{E}^{(m)} \cdot \hat{\mathbb{E}}^{(m)}}{\|\mathbb{E}^{(m)}\|_2 \|\hat{\mathbb{E}}^{(m)}\|_2}\right), \tag{6}$$

where $\mathbb{E}$ and $\hat{\mathbb{E}}$ denote batches of ground-truth and estimated illumination vectors, respectively, m^{th} ground-truth and estimated illumination vectors in the mini-batch are denoted as $\mathbb{E}^{(m)}$ and $\hat{\mathbb{E}}^{(m)}$, respectively, '$\cdot$' is the vector dot product, and $\|.\|_2$ is vector L2 norm.

The total loss for a mini-batch of images is the sum of the end-result loss and weighted intermediate estimation losses as follows

$$\mathrm{L}(\mathbb{E}, \hat{\mathbb{E}}) + \sum_{k=1}^{N} w_k \mathrm{L}(\mathbb{U}, \hat{\mathbb{E}}_k), \tag{7}$$

where $\mathbb{U}$ and $\hat{\mathbb{E}}_k$ denote batches of white illumination vectors and illumination vectors estimated in k^{th} iteration, respectively.

The forward pass in the proposed approach follows the steps in Algorithm 1. It is crucial to emphasize that the forward pass consists of multiple iterations and that the weights of the network are shared across iterations, i.e., the same set of network weights is used in each iteration in the forward pass. This method of the forward pass can be thought of as recurrent since the network is gradually computing the solution from multiple variations of the input image while keeping the set of weights unchanged. Each iteration results in an image with a slight modification of colors obtained by performing the chromatic adaptation of the input in that iteration with illumination estimate, which is also computed in that iteration. The modified image is the input for the succeeding iteration. An illustration of the flow of the proposed method for three iterations is shown in Fig. 2. The only form of supervision during network training is imposed with the loss function, and, in each iteration, in the forward pass, the network estimates intermediate illuminations, which result in a more accurate final estimate.

With the complex form of the forward pass, the backward pass in the proposed approach is complex as well. This is because the final illumination estimate in the forward pass is the product of intermediate estimates, the loss function penalizes each intermediate estimate, and network weights are shared across iterations. Therefore, the gradients propagating through a network layer consist of the gradients induced by the error of the final estimate and by the error of each intermediate estimate with respect to the white light.

IV. EXPERIMENTAL RESULTS

A. EXPERIMENTAL SETUP

Cube+ dataset [45] was used to train and test the proposed illumination estimation network and the iterative procedure. It is a dataset containing 1707 images labeled for global illumination estimation. It consists of images of outdoor scenes in day and night and images of indoor scenes with artificial illuminations. Raw images in the Cube+ dataset are 2601 pixels wide and 1732 pixels high. For the reduction of the computational cost and to utilize as many resources as possible, all images have been resized to the size of 224×224 pixels. Additionally, by resizing the images to the specified shape, the input shape of the pre-trained VGG16 network was matched. Apart from image resizing, standard pre-processing steps for the Cube+ dataset were applied. Pre-processing steps include calibration object masking, black level subtraction, and overexposed pixel removal.

The angular error was used to evaluate the network accuracy. It is computed as the angle between the ground-truth illumination vector and the estimated illumination vector as follows

$$A(\mathbf{e}, \hat{\mathbf{e}}) = \cos^{-1}\left(\frac{\mathbf{e} \cdot \hat{\mathbf{e}}}{\|\mathbf{e}\|_2 \|\hat{\mathbf{e}}\|_2}\right) \tag{8}$$

For comparison with existing methods, a standard evaluation procedure for the evaluation of illumination estimation methods was followed. Mean, median, trimean, best 25%, worst 25%, and average [30] error statistics were computed on the test set. However, the focus of this paper is on reducing maximum estimation errors which can occur in cases of images with highly colored illuminations. By forcing the intermediate illumination estimates to be as close to the white light as possible, the reduction of maximal errors is expected. Therefore, the worst cases were additionally explored. Since other illumination estimation methods do not have maximal estimation errors reported, comparison with them could only be conducted by using the worst 25% statistic.

The following convolutional neural network parameters were used: learning rate 1×10^{-4}, number of epochs 200, min-batch size 8. The feature extraction part that corresponds to the VGG16 network was initialized with weights from the Keras Applications module [46] which were pre-trained on the ImageNet [47] dataset. The newly added convolutional layer was initialized by using the Xavier initialization [48].

B. DETERMINING THE NUMBER OF ITERATIONS

The optimal number of iterations for the proposed method was experimentally determined. Cube+ dataset was used for this purpose. It was split into three parts: train, test, and validation. The train part of the dataset was used to train the proposed network architecture for a different number of iterations. In each experiment, training parameters were the same, as described in subsection IV-A. The optimal number of iterations was obtained by evaluating the trained models on the validation part of the dataset and looking for the one with the lowest median angular error. Once determined, the model with the optimal number of iterations was evaluated on the test part of the dataset, and these results are reported in subsection IV-C.

An important role in determining the optimal number of iterations is the model complexity, which increases in accordance with the number of iterations. The higher the number of iterations is, the more computational memory is needed. Since the proposed method was trained and tested by using the GPU, the size of the GPU memory was a limiting factor for the conducted experiments.

Taking into account method accuracy and GPU memory limits, models with the number of iterations in the range from one to nine were considered, and, as the optimal one, the model with seven iterations was chosen. Therefore in the proposed method and experimental results the number of iterations and, thus, the number of intermediate illumination estimations is set to seven. For comparison, the model performances for a different number of iterations on the test part of the dataset are shown in Fig. 3.

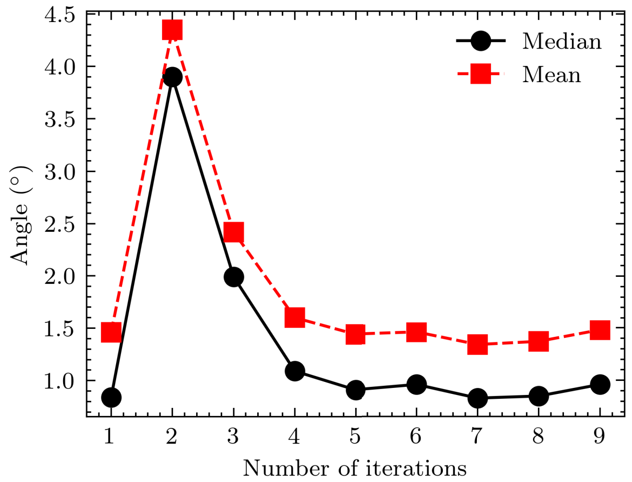


FIGURE 3. Performance of the proposed method for a different number of iterations with respect to the median and mean error statistics.

The proposed multistage approach aimed to achieve the asymptotic convergence of the illumination correction towards no correction. In other words, the preliminary limiting factor was only the amount of the available GPU memory. However, from the experiments, it can be seen that such convergence was not achieved since both mean and median errors start to increase after seven iterations. There are several possible factors for such behavior, with the main one being the imperfection of the simple chromatic adaptation model. Other possible factors include floating-point arithmetic rounding and neural network capacity. Therefore, the proposed search for determining the optimal number of iterations was conducted.

C. METHOD PERFORMANCE

1) COMPARISON WITH EXISTING ILLUMINATION ESTIMATION METHODS

In Table 1, the illumination estimation methods' accuracy on the Cube+ dataset is shown. For evaluation and comparison of the proposed method, final network estimation, i.e., the product of intermediate illumination estimates is used. It can be seen that the proposed method outperforms all other methods on average and in worst-case scenarios. Additionally, both the proposed method and Color Beaver [4] have comparable median and average error statistics that outperform other methods by a notable margin.

The proposed method was tested on a system with Intel(R) Core(TM) i7-8700K CPU @ 3.70GHz central processing unit. The average execution time on the test set using only one core was 2.04 seconds per input image. The proposed model has 14,716,227 weights which is less compared to deep learning-based illumination estimations methods evaluated on the Cube+ dataset in [19], [20], [22], which all use VGG16 network structure for feature extraction, but have more complex additional layer structures, such as attention

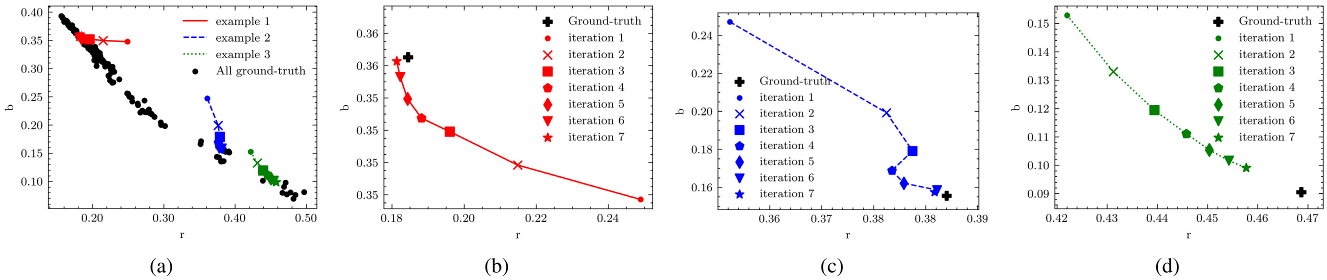


FIGURE 4. Examples of cumulative estimation trajectories with respect to the ground-truth in *rb*-chromaticity space for the proposed approach with seven iterations: (a) cumulative estimation trajectories in comparison to all ground-truth chromaticities; (b), (c), and (d) magnified trajectories for examples 1, 2, and 3 in (a), respectively.

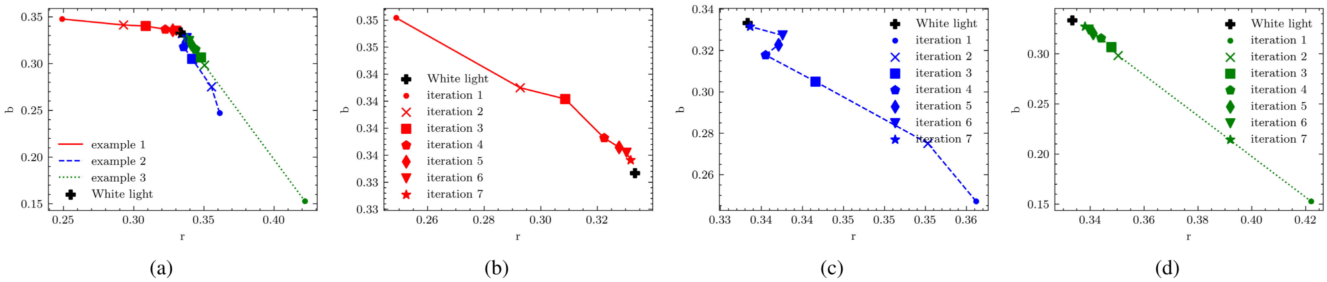


FIGURE 5. Examples of intermediate illumination estimation trajectories with respect to the white light in *rb*-chromaticity space for the proposed approach with seven iterations: (a) estimation trajectories in comparison to white light chromaticities; (b), (c), and (d) magnified trajectories for examples 1, 2, and 3 in (a), respectively. Trajectories correspond to the same examples as in Fig. 4.

TABLE 1. The comparison of angular error statistics of different color constancy methods on the Cube+ dataset [45] (sorted by Avg., lower is better).

Algorithm	Mean	Med.	Tri.	Best 25%	Worst 25%	Avg.
White-Patch [8]	9.69	7.48	8.56	1.72	20.49	7.38
Gray-world [12]	7.71	4.29	4.98	1.01	20.19	5.08
Double-opponency (max pooling) [49]	6.76	3.44	4.15	0.79	18.54	4.27
Using gray pixels [36]	6.65	3.26	3.95	0.68	18.75	4.05
Color Tiger [45]	3.91	2.05	2.53	0.98	10.00	2.88
Color Mule [50]	5.16	1.30	2.03	0.25	16.93	2.25
Shades-of-Gray [13]	2.59	1.73	1.93	0.46	6.19	1.90
2nd-order Gray-Edge [14]	2.50	1.59	1.78	0.48	6.08	1.83
1st-order Gray-Edge [14]	2.41	1.52	1.72	0.45	5.89	1.76
Color Dog [35]	3.32	1.19	1.60	0.22	10.22	1.70
General Gray-World [3]	2.38	1.43	1.66	0.35	6.01	1.64
Attention CNN [20]	2.05	1.32	1.53	0.42	4.84	1.54
Light Source Classification [22]	1.86	1.27	1.39	0.42	4.31	1.43
RGB Attention CNN [19]	1.95	1.13	1.37	0.32	4.92	1.37
Proposed approach	**1.34**	0.83	**0.97**	0.28	**3.20**	**0.99**
Color Beaver (Gray-world) [4]	1.49	**0.77**	0.98	**0.21**	3.94	**0.99**

blocks, or have multiple instances of the same network structure with different weights.

2) METHOD BEHAVIOR VALIDATION

For the rest of the paper, it is important to define the term cumulative estimate. A cumulative estimate in iteration k is the element-wise product of all intermediate estimates up to and including the iteration k. In other words, cumulative estimate in the iteration k can be thought of as the final output of the network if the total number of iterations is equal to k.

The proposed method introduces iterative illumination estimation which forces intermediate illumination estimates computed in each iteration to be close to the white light and when multiplied element-wise altogether to be equal to the scene illumination. By the construction of the method, it is expected for intermediate estimates to be closer to the white light with each iteration. Also, it is expected for cumulative estimates to be closer to the ground-truth as iterations progress. Neither intermediate estimates nor cumulative estimates should fluctuate in illumination space. Such behavior can be verified in Fig. 4, and Fig. 5 where few examples of estimation trajectories for different input images with respect to the ground-truth and white light are shown. A trajectory represents the path enclosed by either intermediate or cumulative estimates through iterations. In Fig. 4 cumulative estimations with respect to the ground-truth are considered, and in Fig. 5 intermediate estimations with respect to the white light are considered.

Since the proposed method uses estimates from multiple versions of an input image to compute the color of scene illumination, naturally, a question of the benefit of using more estimations compared to a single estimate arises. Therefore, the proposed network architecture was also trained for one iteration only. The same set of parameters was used as described in subsection IV-A: learning rate 1×10^{-4}, epoch 200, and mini-batch size 8. When only one iteration is used, chromatic adaptation is not performed, and the first intermediate estimate is actually the final network estimate.

TABLE 2. The comparison of angular error statistics of the proposed method and the baseline (lower is better).

Algorithm	Min	Max	Mean	Med.	Tri.	Best 25%	Worst 25%	Avg.
Baseline	0.02	9.47	1.46	0.84	0.98	0.23	3.73	1.01
Proposed approach	0.03	7.36	1.34	0.83	0.97	0.28	3.20	0.99

TABLE 3. The comparison of angular error statistics of the proposed method and the baseline on worst-performing samples for the baseline on the test set (lower is better).

Algorithm	Min	Max	Mean	Med.	Tri.	Best 25%	Worst 25%	Avg.
Baseline	3.93	9.47	5.71	5.55	5.51	4.24	7.57	5.62
Proposed approach	0.76	7.36	3.57	3.39	3.44	1.36	5.93	3.20

TABLE 4. The comparison of angular error statistics of the proposed method and the baseline on worst-performing samples for the proposed method on the test set (lower is better).

Algorithm	Min	Max	Mean	Med.	Tri.	Best 25%	Worst 25%	Avg.
Baseline	0.50	9.47	4.11	4.33	4.09	1.18	7.17	3.61
Proposed approach	3.29	7.36	4.55	4.34	4.39	3.42	6.07	4.48

In other words, illumination is estimated from the original image directly. Consequently, calculating the loss during the network training consisted only of the first part of the loss calculation, which is based on the cosine of the angle between the ground-truth and final illumination estimation. In further text, this experiment with one iteration will be referred to as the baseline. In Table 2, the comparison of the angular error statistics of the baseline with the proposed method is shown. It can be seen that the proposed method outperforms the baseline, especially in the case of the mean statistic and worst-performing samples.

To further validate the benefit of the proposed method, additional comparisons were made. In Table 3, estimation error statistics for the proposed method and the baseline method on worst performing samples for the baseline are shown. Worst performing samples are samples with estimation angular error higher than the value of the worst 25% statistic on the whole test set. For the baseline method, that value is 3.73°, and 33 samples have a higher error value. For 90.01% of such samples, the proposed method outperforms the baseline. Considering only the samples for which the proposed method is more accurate, the mean absolute error difference between estimates of the proposed method and estimates of the baseline is 2.43°, and when only the samples for which the baseline is more accurate are considered the difference is 0.73°. The same experiment was repeated with a different set of worst-performing samples. In Table 4, estimation error statistics for the proposed method and the baseline method on worst performing samples for the proposed method are shown. Worst performing samples were sampled using the same criterion as in the previous

TABLE 5. The comparison of angular error statistics of the proposed method and the baseline on the worst-performing samples for both the proposed method and the baseline on the test set (lower is better).

Algorithm	Min	Max	Mean	Med.	Tri.	Best 25%	Worst 25%	Avg.
Baseline	4.08	9.47	5.92	5.55	5.56	4.35	8.02	5.77
Proposed approach	3.29	7.36	4.82	4.77	4.68	3.47	6.33	4.73

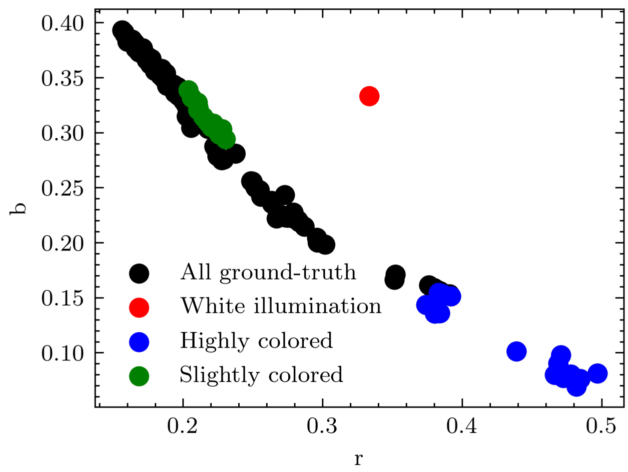


FIGURE 6. The distribution of highly colored ground-truth illuminations and slightly colored ground-truth illuminations in the test set.

example. This time the threshold value was 3.20° since that is the value of the worst 25% statistic on the whole test set for the proposed method. Even though these samples were the ones for which the proposed method had the lowest accuracy, for 45.71% of samples the proposed method outperformed the baseline. The mean absolute error difference between proposed method estimates and baseline estimates when considering only the samples for which the proposed method was more accurate was 1.45°, and 2.04° when considering only the samples for which the baseline was more accurate. Finally, estimation error statistics for the proposed method and the baseline method on the intersection of worst-performing samples for both the proposed method and the baseline are given in Table 5. It can be seen that the proposed method outperforms the baseline by a significant margin.

Further method validation includes the comparison of method performance on images in two extrema. One extreme is images of scenes in artificial illuminations where scene illumination significantly differs from white illumination (in further text highly colored images). The second extreme contains images in daylight where the illumination was near white, i.e., illumination did not have a significant effect on image colors (in further text slightly colored images). To sample highly and slightly colored images, firstly, the angular distances between the ground-truth illuminations in the test set and a white illumination were computed according to (8). Then, highly colored images were sampled by taking images

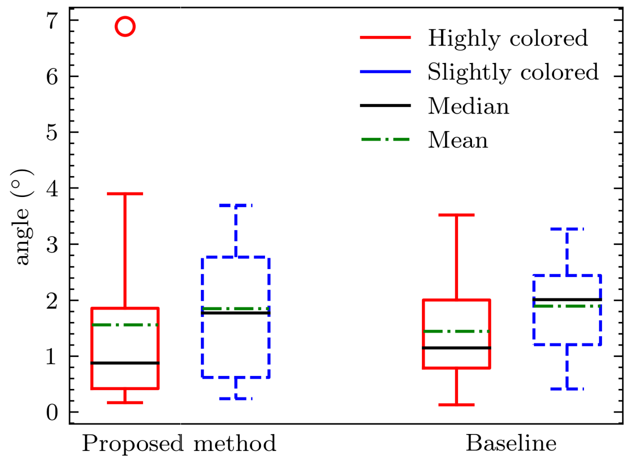


FIGURE 7. Box plot of angular errors of the proposed method and the baseline on highly colored images and slightly colored images.

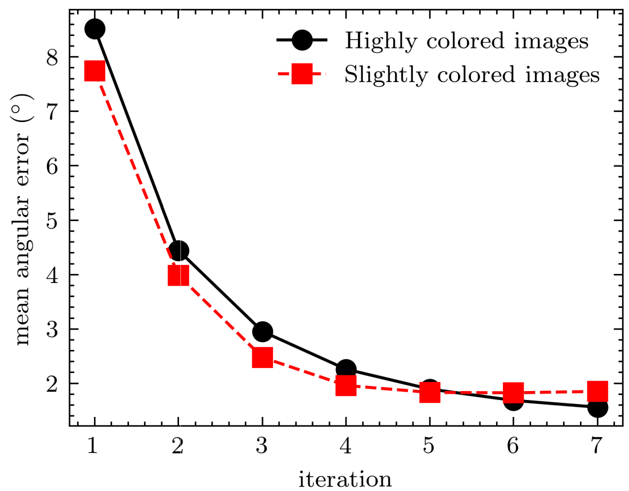


FIGURE 8. Mean angular error between cumulative estimates in each iteration and ground-truth illuminations for highly colored images and slightly colored images.

with corresponding angular distance within the 5% highest values, and slightly colored images were sampled by taking images with corresponding angular distance within the 5% lowest values. In Fig. 6, *rb*-chromaticities of ground-truth illuminations separated based the classification of highly and slightly colored images are shown.

In Fig. 7, the box plot of angular errors for the proposed method and the baseline on highly colored images and slightly colored images is given. For both groups of images, the proposed method outperforms the baseline with median angular errors 0.88° and 1.78° for highly colored images and slightly colored images, respectively. Median angular errors for the baseline were 1.15° for highly colored images and 2.01° for slightly colored images.

Since the proposed method reduces maximal estimation errors by forcing the intermediate illumination estimations to

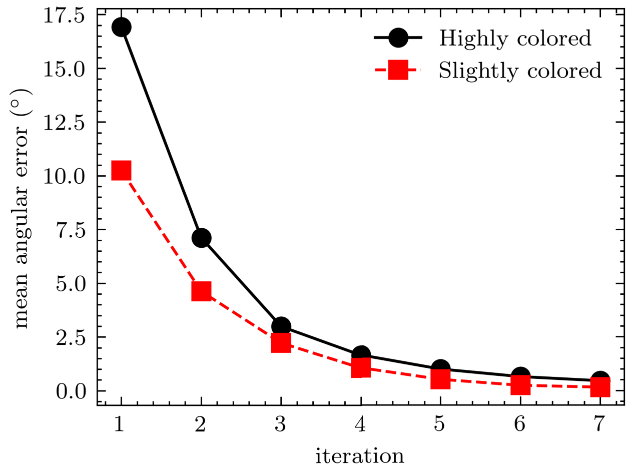


FIGURE 9. Mean angular error between intermediate estimates in each iteration and a white light illumination for highly colored images and slightly colored images.

be close to the white light, it is expected that the convergence to the ground-truth illumination is slower on highly colored images than on slightly colored images. Such behavior is shown in Fig. 8 and Fig. 9. In Fig. 8, it can be seen that for slightly colored images cumulative illumination estimates approach close to ground-truth values much faster than for highly colored images and, what is more important, after the convergence the angular error does not increase in remaining iterations. In Fig. 9, the same trend can be observed with respect to the convergence of intermediate illumination estimates on highly colored images and slightly colored images towards the white light.

V. CONCLUSION

Illumination estimation is an ill-posed problem and as such, it can not be explicitly solved. Moreover, in computational color constancy, it is usually followed by a chromatic adaptation that uses an illumination estimation expressed as a diagonal matrix which assumes independence of image color channels. Both processes are simple and may fail in some cases but when combined together in a controlled manner they could be used for iterative illumination estimation. In this paper, such an illumination estimation method is proposed. It combines illumination estimation and chromatic adaptation in a sequence. The convolutional neural network is used to compute multiple intermediate illumination estimates from an input image, which, when multiplied, correspond to the real scene illumination. By forcing the intermediate illumination estimates to be close to the white light, the proposed method avoids the estimation of highly inaccurate illuminations. The experimental results successfully validate the proposed method and its accuracy, especially in the case of worst-performing samples. Future research will include looking for an early stopping mechanism that should stop the method from entering further iterations if it already converged to the best solution it can calculate.

ACKNOWLEDGMENT

The authors would like to thank the support of NVIDIA Corporation with the donation of the Titan Xp GPU used for this research.

REFERENCES

[1] 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.

[2] M. Ebner, "Color constancy," in *The Wiley-IS&T Series in Imaging Science and Technology*. Hoboken, NJ, USA: Wiley, 2007.

[3] K. Barnard, V. Cardei, and B. Funt, "A comparison of computational color constancy algorithms. I: Methodology and experiments with synthesized data," *IEEE Trans. Image Process.*, vol. 11, no. 9, pp. 972–984, Sep. 2002.

[4] K. Koščević, N. Banić, and S. Lončarić "Color beaver: Bounding illumination estimations for higher accuracy," in *Proc. 14th Int. Joint Conf. Comput. Vis., Imag. Comput. Graph. Theory Appl.*, 2019, pp. 183–190.

[5] S. D. Hordley, "Scene illuminant estimation: Past, present, and future," *Color Res. Appl.*, vol. 31, no. 4, pp. 303–314, 2006.

[6] N. Banić and S. Loncarić, "Flash and storm: Fast and highly practical tone mapping based on naka-rushton equation," in *Proc. 13th Int. Joint Conf. Comput. Vis., Imag. Comput. Graph. Theory Appl.*, 2018, pp. 47–53.

[7] E. H. Land, *The Retinex Theory of Color Vision*. Boston, MA, USA: Scientific America, 1977.

[8] B. Funt and L. Shi, "The rehabilitation of MaxRGB," in *Proc. 18th Color Imag. Conf. Final Program Color Imag. Conf.*, vol. 2010, no. 1. Springfield, VA, USA: Society for Imaging Science and Technology, 2010, pp. 256–259.

[9] N. Banić and S. Lončarić, "Using the random sprays Retinex algorithm for global illumination estimation," in *Proc. The 2nd Croatian Comput. Vis. Workshopn (CCVW)*. Zagreb, Croatia: Univ. Zagreb Faculty of Electrical Engineering and Computing, 2013, pp. 3–7.

[10] N. Banic and S. Loncaric, "Color rabbit: Guiding the distance of local maximums in illumination estimation," in *Proc. 19th Int. Conf. Digit. Signal Process.*, Aug. 2014, pp. 345–350.

[11] N. Banic and S. Loncaric, "Improving the white patch method by subsampling," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2014, pp. 605–609.

[12] G. Buchsbaum, "A spatial processor model for object colour perception," *J. Franklin Inst.*, vol. 310, no. 1, pp. 1–26, Jul. 1980.

[13] G. D. Finlayson and E. Trezzi, "Shades of gray and colour constancy," in *Proc. Color Imag. Conf.*, vol. 2004, no. 1. Springfield, VA, USA: Society for Imaging Science and Technology, 2004, pp. 37–41.

[14] 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.

[15] A. Gijsenij, T. Gevers, and J. van de Weijer, "Improving color constancy by photometric edge weighting," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 5, pp. 918–929, May 2012.

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

[17] Z. Lou, T. Gevers, N. Hu, and M. P. Lucassen, "Color constancy by deep learning," in *Proc. Brit. Mach. Vis. Conf.*, 2015, pp. 1–76.

[18] Y. Hu, B. Wang, and S. Lin, "Fully Convolutional Color Constancy with Confidence-weighted Pooling," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4085–4094.

[19] K. Koscevic, M. Subasic, and S. Loncaric, "Attention-based convolutional neural network for computer vision color constancy," in *Proc. 11th Int. Symp. Image Signal Process. Anal. (ISPA)*, Sep. 2019, pp. 372–377.

[20] K. Koščević, M. Subašić, and S. Lončarić, "Guiding the illumination estimation using the attention mechanism," in *Proc. 2nd Asia Pacific Inf. Technol. Conf.*, Jan. 2020, pp. 143–149, doi: 10.1145/3379310.3379329.

[21] 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.

[22] K. Koscevic, M. Subasic, and S. Loncaric, "Deep learning-based illumination estimation using light source classification," *IEEE Access*, vol. 8, pp. 84239–84247, 2020.

[23] W. Shi, C. C. Loy, and X. Tang, "Deep specialized network for illuminant estimation," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 371–387.

[24] M. Afifi and M. S. Brown, "Sensor-independent illumination estimation for DNN models," 2019, *arXiv:1912.06888*. [Online]. Available: http://arxiv.org/abs/1912.06888

[25] 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.

[26] G. D. Finlayson, "Corrected-moment illuminant estimation," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 1904–1911.

[27] D. A. Forsyth, "A novel algorithm for color constancy," *Int. J. Comput. Vis.*, vol. 5, no. 1, pp. 5–35, Aug. 1990.

[28] K. Barnard, "Improvements to gamut mapping colour constancy algorithms," in *Proc. Eur. Conf. Comput. Vis.* Berlin, Germany: Springer, 2000, pp. 390–403.

[29] G. D. Finlayson, S. D. Hordley, and I. Tastl, "Gamut constrained illuminant estimation," *Int. J. Comput. Vis.*, vol. 67, no. 1, pp. 93–109, Apr. 2006.

[30] J. T. Barron, "Convolutional color constancy," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 379–387.

[31] J. T. Barron and Y.-T. Tsai, "Fast Fourier color constancy," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 886–894.

[32] J. van de Weijer, C. Schmid, and J. Verbeek, "Using high-level visual information for color constancy," in *Proc. IEEE 11th Int. Conf. Comput. Vis.*, Oct. 2007, pp. 1–8.

[33] N. Banic and S. Loncaric, "Color cat: Remembering colors for illumination estimation," *IEEE Signal Process. Lett.*, vol. 22, no. 6, pp. 651–655, Jun. 2015.

[34] N. Banic and S. Loncaric, "Using the red chromaticity for illumination estimation," in *Proc. 9th Int. Symp. Image Signal Process. Anal. (ISPA)*, Sep. 2015, pp. 131–136.

[35] N. Banic and S. Loncaric, "Color dog–guiding the global illumination estimation to better accuracy," in *Proc. 10th Int. Conf. Comput. Vis. Theory Appl.*, Mar. 2015, pp. 129–135.

[36] K.-F. Yang, S.-B. Gao, and Y.-J. Li, "Efficient illuminant estimation for color constancy using grey pixels," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 2254–2263.

[37] 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. (CVPR)*, Jun. 2015, pp. 1000–1008.

[38] G. D. Finlayson, M. S. Drew, and B. V. Funt, "Diagonal transforms suffice for color constancy," in *Proc. 4th Int. Conf. Comput. Vis.*, May 1993, pp. 164–171.

[39] G. West and M. H. Brill, "Necessary and sufficient conditions for von kries chromatic adaptation to give color constancy," *J. Math. Biol.*, vol. 15, no. 2, pp. 249–258, Oct. 1982.

[40] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*. [Online]. Available: http://arxiv.org/abs/1409.1556

[41] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size," 2016, *arXiv:1602.07360*. [Online]. Available: http://arxiv.org/abs/1602.07360

[42] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017.

[43] Y. A. LeCun, L. Bottou, G. B. Orr, and K.-R. Müller, "Efficient backprop," in *Neural Networks: Tricks Trade*. Berlin, Germany: Springer, 2012, pp. 9–48.

[44] O. Sidorov, "Artificial color constancy via GoogleNet with angular loss function," *Appl. Artif. Intell.*, vol. 34, no. 9, pp. 643–655, 2020.

[45] N. Banić, K. Koščević, and S. Lončarić, "Unsupervised learning for color constancy," 2017, *arXiv:1712.00436*. [Online]. Available: http://arxiv.org/abs/1712.00436

[46] F. Chollet. (2015). *Keras*. [Online]. Available: https://keras.io

[47] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.

[48] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. 13th Int. Conf. Artif. Intell. Statist.*, 2010, pp. 249–256.

[49] S.-B. Gao, K.-F. Yang, C.-Y. Li, and Y.-J. Li, "Color constancy using double-opponency," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 10, pp. 1973–1985, Oct. 2015.

[50] N. Banić and S. Lončarić, "A perceptual measure of illumination estimation error," in *Proc. 10th Int. Conf. Comput. Vis. Theory Appl.*, Mar. 2015, pp. 136–143.

KARLO KOŠČEVIĆ (Graduate Student Member, IEEE) received the B.Sc. and M.Sc. degrees in computer science in 2016 and 2018, respectively. He is currently in his second year of the technical sciences in the scientific field of computing Ph.D. program with the Faculty of Electrical Engineering and Computing, University of Zagreb, Croatia. His research interests include image processing, image analysis, and deep learning. His current research is in the area of color constancy with a focus on learning-based methods for illumination estimation.

MARKO SUBAŠIĆ (Member, IEEE) received the Ph.D. degree from the Faculty of Electrical Engineering and Computing, University of Zagreb, in 2007. Since 1999, he has been working with the Department for Electronic Systems and Information Processing, Faculty of Electrical Engineering and Computing, University of Zagreb, where he is currently an Associate Professor. He teaches several courses at the graduate and undergraduate levels. His research interests include image processing and analysis and neural networks, with a particular interest in image segmentation, detection techniques, and deep learning. He is also a member of the IEEE Computer Society, the Croatian Center for Computer Vision, the Croatian Society for Biomedical Engineering and Medical Physics, and the Centre of Research Excellence for Data Science and Advanced Cooperative Systems.

SVEN LONČARIĆ (Senior Member, IEEE) received the Ph.D. degree in electrical engineering from the University of Cincinnati, Cincinnati, OH, USA, in 1994, as a Fulbright Scholar. He was an Assistant Professor with the New Jersey Institute of Technology, Newark, NJ, USA, from 2001 to 2003. He is currently a Professor of Electrical Engineering and Computer Science at the Faculty of Electrical Engineering and Computing, University of Zagreb, Croatia. He was the principal investigator on a number of R&D projects. He is the Director of the Center for Computer Vision, University of Zagreb and the Head of the Image Processing Group. He is a Co-Director of the Center of Excellence in Data Science and Cooperative Systems. He has coauthored more than 250 publications in scientific journals and conferences. His research interests include image processing and computer vision. He was the Chair of the IEEE Croatia Section. He is a member of the Croatian Academy of Technical Sciences. He received several awards for his scientific and professional work.