https://doi.org/10.2352/CIC.2023.31.1.18

Learning color constancy: 30 years later

Marco Buzzelli, Raimondo Schettini, Simone Bianco; Department of Informatics Systems and Communication, University of Milano – Bicocca; Milan 20126, Italy

Abstract

The first paper investigating the use of machine learning to learn the relationship between an image of a scene and the color of the scene illuminant was published by Funt et al. in 1996. Specifically, they investigated if such a relationship could be learned by a neural network. During the last 30 years we have witnessed a remarkable series of advancements in machine learning, and in particular deep learning approaches based on artificial neural networks. In this paper we want to update the method by Funt et al. by including recent techniques introduced to train deep neural networks. Experimental results on a standard dataset show how the updated version can improve the median angular error in illuminant estimation by almost 51% with respect to its original formulation, even outperforming recent illuminant estimation methods.

Introduction

The chromatic attributes of the objects we perceive in a visual scene are primarily influenced by three distinct elements: i) the spectral reflectance characteristics of the object surfaces; ii) the spectral power distribution of the illuminating light source; iii) the relative spatial arrangement of the objects and the illuminant. The primary objective of computational color constancy, also known as automatic white balancing, is to render the objects within the scene as if they were observed under a chosen neutral illuminant. Consequently, it is evident why this process holds significant importance in digital camera workflows and why numerous computer vision problems employ color constancy as a preliminary pre-processing step. Computational color constancy is often addressed as a two-step procedure: the first step is illuminant estimation, and it aims to estimate the color of the illuminant or the lighting conditions present in the scene; the second step is illuminant correction, that is typically done by applying a chromatic adaptation transform, which scales the color channels of the image to compensate for the estimated illuminant color.

However, despite its apparent simplicity, color constancy is an ill-posed problem. In order to solve it, additional assumptions on the imaged content have to be made: common assumptions are for example the gray-world hypothesis, which assumes that the average reflectance in a scene is achromatic, or the white patch hypothesis, based on the assumption that the maximum values recorded for each color channel correspond to the color of the illuminant. Given an input image, if the assumptions are satisfied, the image will be properly corrected. Since these assumptions may be not always satisfied, Funt et al. in 1996 [1, 2] tested a surprisingly simple hypothesis: that the relationship between image content and the chromaticity of scene illuminant could be learned by a neural network, paving the way for the development of learning-based illuminant estimation algorithms. Nowadays, about 30 years later, learning-based illuminant estimation methods are the state of the art and nearly all recently published methods belong to this category.

In this paper, starting from the work by Funt et al., we modernize it by incorporating state-of-the-art techniques introduced for deep neural network training, to see how much its illuminant estimations can be improved.

Related Works

Illuminant estimation methods can be divided into two categories: learning-free, and learning-based. In this paper we focus on the learning-based group.

The first learning-based illuminant estimation algorithm is by Funt et al. [1, 2], where a Fully Connected network (i.e., a Multi-Layer Perceptron) is used to map from the scene chromaticities to the color of the illuminant. Inspired by this work, in the next years several different machine learning algorithms have been employed with the aim of further improving illuminant estimation accuracy. Bayesian approaches [3] are used to model the variability of reflectance and of illuminant as random variables, and then estimate illuminant from the posterior distribution conditioned on image data. Random Forests are used to drive the selection of the best algorithm (or the best combination of algorithms) for a given image using as input low-level properties of the images [4, 5].

Inspired by the success obtained by deep Convolutional Neural Networks (CNN) on several Computer Vision tasks, they have also been successfully applied to the problem of illuminant estimation. The first work using CNNs is by Bianco et al. [6], then followed for example by [7, 8, 9]. Most recent approaches use different deep neural architectures and training procedures, such as Generative Adversarial Networks (GANs), e.g. [10, 11], or change the learning paradigm from supervised learning to unsupervised learning [12].

Proposed approach

The original work by Funt et al. [1, 2] uses a Multi-Layer Perceptron (MLP) to map the chromaticities of the colors present in the original scene into the illuminant chromaticity. The chromaticity space adopted is the $rg = (R/(R+G+B), G = G/(R+G+B))$ sampled in steps of 0.020. The input is encoded as a 1250-dimensional vector and its values are binary, indicating the presence or absence of that particular chromaticity in the input image. The input dimension 1250 derives from the fact that the rg chromaticity space is actually a triangle, and therefore there is not need to encode the remaining 1250 values. The architecture of the MLP is 1250-200-40-2, thus consisting of two hidden layers, with sigmoid activation functions for a total of about 260k trainable parameters. The error function used for training the network is the Euclidean distance between the target and the estimated illuminant in the rg-chromaticity space. This original formulation is referred to as LCC-v0 in this paper, and constitutes our starting point to build upon.

The first modernization step, that we call LCC-v1, substitutes the sigmoid activation functions with the Rectifying Linear Unit (ReLU) [13], which has been shown to reach faster training times. It also changes the number of outputs from two to three, thus directly estimating the RGB color of the scene illuminant. It also changes the used training loss, replacing the Euclidean distance in the chromaticity space with an angular distance in RGB space, thus measuring training errors in the same way they are used in test to compare the performance of different color constancy algorithms.

The second modernization step, i.e., LCC-v2, introduces the use of dropout [14, 15] after each fully connected layer during the training phase. Dropout reduces overfitting preventing complex co-adaptations on the training data, and therefore improves the performance on new unseen data.

Starting from the third modernization step, we introduce a change in the neural network type, moving from the shallow fully-connected architecture of a MLP to a deep Convolutional Neural Network (CNN). In particular a ResNet-18 [16] is used, which accepts 224×224 inputs and has a total of about 11M trainable parameters. For LCC-v3 the input now consists in the *rg*-chromaticity histogram sampled in steps of 0.020 as in LCC-v0 to perform a fair comparison (see Figure 1). This results in a 50×50 histogram that is then normalized to unitary maximum, upsampled to 224×224 pixels by bilinear interpolation, multiplied by 255, and saved as an 8-bit single-channel image. During training, random erase augmentation is used, which randomly selects a rectangle region in the input image and erases its pixels setting them to zero.

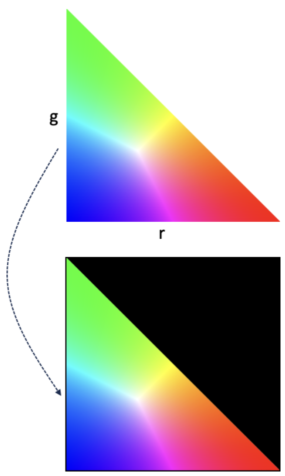

Figure 1. *rg-chromaticity space (top) and its occupation of the input image fed to LCC-v3 (bottom).*

The last two models, i.e., LCC-v4 and LCC-v5, stem from the observation that the *rg*-chormaticity space is a triangle that uses just half of the input image. We therefore include a second chromaticity histogram in the remaining triangle, by respectively rotating by 180 degrees the *rb*-chromaticity histogram from LCC-v4 and the *gb*-chromaticity histogram for LCC-v5 (see Figure 2).

Table 1 summarizes the main characteristics of the original

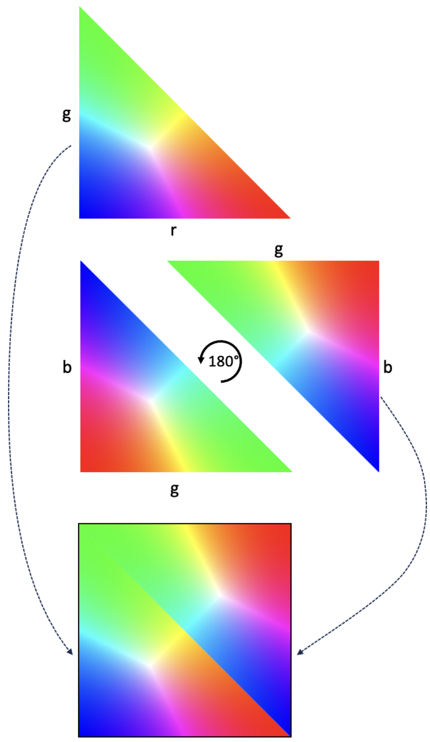

Figure 2. *rg-chromaticity space (top); gb-chromaticity space and its 180 degrees rotated version (middle); how rg- and gb-chromaticity spaces are spatially combined to form the input image fed to LCC-v5 (bottom).*

version of LCC and the different updated versions here introduced.

Experimental setup

All the experiments are performed on the Gehler-Shi dataset [3] also known as the ColorCheker dataset, using the REC groudtruth [17, 18], and following the fair comparison procedure defined in [19]. All the different LCC architectures are trained in PyTorch on this dataset for a total of 1000 epochs, batch size equal to 16, Adam optimizer [20] with a learning rate equal to $3e-4$, and a weight decay equal to $5e-5$. No other augmentations are used besides those described in the previous section. The ResNet-18 architecture used by LCC-v3, v4, and v5 is initialized with ImageNet weights; the first and last layers are then replaced to match the input and output characteristics of our illuminant estimation problem (see Figure 3).

Experimental results

The performance in terms of recovery angular errors for the different versions of LCC proposed in this paper are reported in Table 2. Considering the median angular error, we observe that the replica of the work by Funt et al. [1, 2], i.e. LCC-v0, is the one having the largest error and is considered as the baseline for the subsequent comparisons. Changing the activation function to ReLU and using an angular loss (LCC-v1), improves the median angular error over the baseline by 13.8%. On top of this, the use of dropout regularization improves the median angular error over the baseline by 22.5%. Changing the neural architecture to a CNN,

Table 1. Summary of the main characteristics of the original work by Funt et al. [1, 2] (LCC-v0) and the updates proposed in this paper (LCC-v1 to LCC-v5). They are grouped on the basis of the architecture used: MLP (a), and CNN (b).

Method	Architecture	Input type	Color space	Activation	Dropout	Loss
LCC-v0 (replica [1, 2])	MLP	Binarized histogram	rg	Sigmoid	No (did not exist)	MSE between chromaticities
LCC-v1	MLP	Binarized histogram	rg	ReLU	No	Angle between RGB vectors
LCC-v2	MLP	Binarized histogram	rg	ReLU	Yes (0.5)	Angle between RGB vectors

a)

Method	Architecture	Input type	Color space	Loss
LCC-v3	CNN (ResNet-18)	Histogram (normalized)	rg	Angle between RGB vectors
LCC-v4	CNN (ResNet-18)	Histogram (normalized)	rg & rb	Angle between RGB vectors
LCC-v5	CNN (ResNet-18)	Histogram (normalized)	rg & gb	Angle between RGB vectors

b)

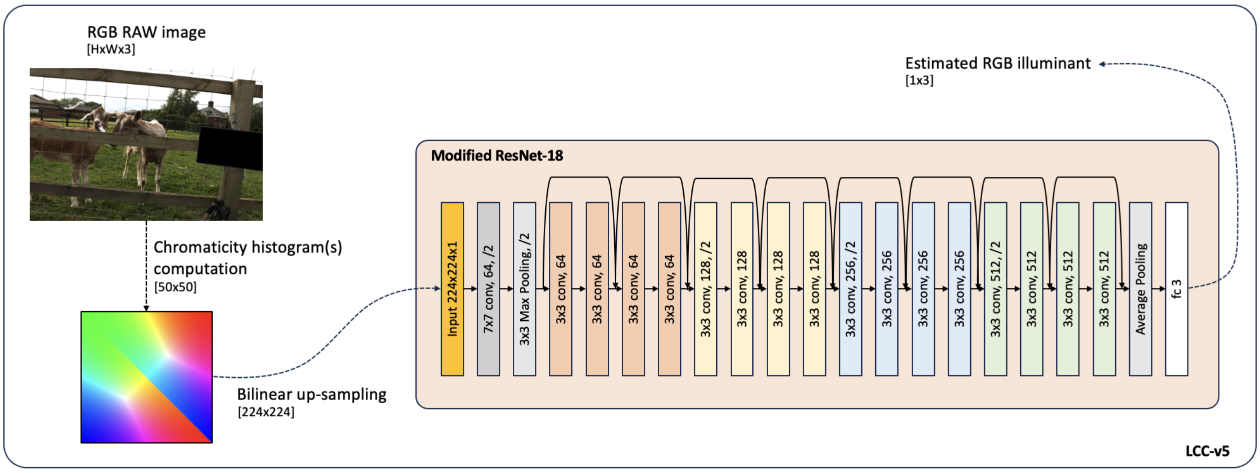

Figure 3. *Schematic representation of the proposed LCC-v5 (LCC-v3 and v4 can be obtained with slight modifications: starting from an input RAW RGB image, chromaticity histograms are computed (rg for LCC-v3, rg and rb for LCC-v4, rg and gb for LCC-v5); histograms are bilinearly interpolated to match the ResNet-18 input size; the CNN estimates the RGB illuminant.*

and changing also the input type to a bi-dimensional chromaticity histogram (LCC-v3), improves the median angular error over the baseline by 44.3%. Filling the empty area of the input with a second chromaticity histogram improves the median angular error over the baseline by 49.8% and 51.0% for LCC-v4 and LCC-v5 respectively.

From the previous analysis we can notice how the largest improvement is due to the change of the neural architecture used: the use of a deep CNN in fact permits to exploit the spatial relationship existing among the bins of the input chromaticity histogram. We can also observe that LCC-v0 and v1 are those reaching the lowest errors in terms of 99th percentile and maximum error. The CNN-based solution able to limit this error increase in the highest percentiles is LCC-v5, with a 99th percentile and a maximum error that are respectively 6.1% and 7.5% worse than the baseline. We can therefore conclude that the overall best performing version is LCC-v5.

In Table 3 we compare the performance of LCC-v5 with several learning-based methods in the state of the art. Performance of other methods in the state of the art can be found for example in [12].

From the results reported in Table 3 we can see how the proposed LCC-v5 outperforms the considered methods in the state of the art in terms of the average, the median, the tri-mean, and the best 25% angular errors. It obtains the second best statistics in terms of the worst 25%, 95th and 99th percentile. Finally, it obtains the third best maximum error.

Conclusion

The first learning-based color constancy algorithm was published about 30 years ago, and was based on a Multi-Layer Perceptron to map from the chromaticities present in the scene to the illuminant chromaticity. Neural Networks were largely disregarded in the period known as AI winter, and learning-based color constancy algorithms were proposed exploiting different machine learning techniques. In the last ten years, with advancements in computing resources, the accumulation of large-scale datasets, and breakthroughs in training techniques, neural networks experienced a resurgence in popularity, leading to their current prominence in the field of artificial intelligence and machine learning. In this paper therefore, we have proposed different update steps of the original work by Funt et al. including recent deep learning techniques and architectures.

Experimental results on a standard dataset show how the updated version can improve the median angular error in illuminant estimation by almost 51% with respect to its original formulation,

Table 2. Performance comparison in terms of estimation angular error of the different versions of LCC proposed in this paper on the ColorChecker dataset [3] following the fair comparison procedure [19]. The best result for each statistic is reported in blue, the second one in red, and the third one in orange.

Method	Mean	Med.	Tri-m.	B-25	W-25	95-P	99-P	Max
LCC-v0 (replica [1, 2])	3.28	2.53	2.72	0.92	6.80	8.68	11.49	15.37
LCC-v1	2.96	2.18	2.32	0.66	6.66	8.99	12.12	14.62
LCC-v2	2.84	1.96	2.10	0.51	6.73	8.95	13.97	20.66
LCC-v3	2.38	1.41	1.60	0.35	6.07	8.35	13.35	19.12
LCC-v4	2.13	1.27	1.45	0.32	5.37	7.24	12.19	19.74
LCC-v5	2.12	1.24	1.46	0.33	5.39	7.14	12.19	16.52

Table 3. Performance comparison of LCC-v5 with several learning-based illuminant estimation algorithms in the state of the art in terms of estimation angular error on the ColorChecker dataset [3] following the fair comparison procedure [19]. The best result for each statistic is reported in blue, the second one in red, and the third one in orange.

Method	Mean	Med.	Tri-m.	B-25	W-25	95-P	99-P	Max
Cheng et al. [21]	4.03	2.49	2.92	0.52	9.89	12.70	16.78	28.21
Corrected Moments (9 Edge Mom.)[22]	2.84	2.00	2.23	0.71	6.32	7.56	12.38	16.60
Cheng et al. [23]	2.56	1.66	1.83	0.36	6.36	8.14	13.17	20.36
FFCC (model j) [24]	2.23	1.45	1.59	0.35	5.46	7.33	10.85	17.27
FC^4 [9]	2.14	1.44	1.57	0.40	5.08	6.50	12.75	15.28
Convolutional Mean [25]	2.50	1.73	1.90	0.51	5.81	7.93	12.42	16.13
QU [12]	3.26	2.07	2.38	0.44	7.98	10.78	14.32	21.68
QU+ft [12]	3.02	2.07	2.26	0.50	7.15	9.22	12.92	17.05
LCC-v5 (this paper)	2.12	1.24	1.46	0.33	5.39	7.14	12.19	16.52

even outperforming recent learning-based illuminant estimation methods.

References

[1] B. Funt, V. Cardei, and K. Barnard, "Learning color constancy," in *Fourth Color Imaging Conference: Color Science, Systems and Applications*. Simon Fraser University, 1996, pp. 58–60.

[2] V. C. Cardei, B. Funt, and K. Barnard, "Estimating the scene illumination chromaticity by using a neural network," *JOSA a*, vol. 19, no. 12, pp. 2374–2386, 2002.

[3] P. V. Gehler, C. Rother, A. Blake, T. Minka, and T. Sharp, "Bayesian color constancy revisited," in *2008 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2008, pp. 1–8.

[4] S. Bianco, G. Ciocca, C. Cusano, and R. Schettini, "Automatic color constancy algorithm selection and combination," *Pattern recognition*, vol. 43, no. 3, pp. 695–705, 2010.

[5] A. Gijsenij and T. Gevers, "Color constancy using natural image statistics and scene semantics," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 4, pp. 687–698, 2010.

[6] S. Bianco, C. Cusano, and R. Schettini, "Color constancy using cnns," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2015, pp. 81–89.

[7] Z. Lou, T. Gevers, N. Hu, M. P. Lucassen *et al.*, "Color constancy by deep learning." in *BMVC*, 2015, pp. 76–1.

[8] S. Bianco, C. Cusano, and R. Schettini, "Single and multiple illuminant estimation using convolutional neural networks," *IEEE Transactions on Image Processing*, vol. 26, no. 9, pp. 4347–4362, 2017.

[9] Y. Hu, B. Wang, and S. Lin, "Fc4: Fully convolutional color constancy with confidence-weighted pooling," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4085–4094.

[10] P. Das, A. S. Baslamisli, Y. Liu, S. Karaoglu, and T. Gevers, "Color constancy by gans: An experimental survey," *arXiv preprint arXiv:1812.03085*, 2018.

[11] M. Buzzelli, R. Riva, S. Bianco, and R. Schettini, "Consensus-driven illuminant estimation with gans," in *Thirteenth International Conference on Machine Vision*, vol. 11605. SPIE, 2021, pp. 578–584.

[12] S. Bianco and C. Cusano, "Quasi-unsupervised color constancy," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 12 212–12 221.

[13] A. F. Agarap, "Deep learning using rectified linear units (relu)," *arXiv preprint arXiv:1803.08375*, 2018.

[14] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *arXiv preprint arXiv:1207.0580*, 2012.

[15] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[16] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[17] G. Hemrit, G. D. Finlayson, A. Gijsenij, P. Gehler, S. Bianco, B. Funt, M. Drew, and L. Shi, "Rehabilitating the colorchecker dataset for illuminant estimation," *arXiv preprint arXiv:1805.12262*, 2018.

[18] G. Hemrit, G. D. Finlayson, A. Gijsenij, P. Gehler, S. Bianco, M. S. Drew, B. Funt, and L. Shi, "Providing a single ground-truth for illuminant estimation for the colorchecker dataset," *IEEE transactions on pattern analysis and machine intelligence*, vol. 42, no. 5, pp.

1286–1287, 2019.

[19] M. Buzzelli, G. D. Finlayson, A. Gijsenij, P. Gehler, M. S. Drew, L. Shi, and S. Bianco, "On the fair use of the ColorChecker dataset for illuminant estimation," *IEEE Transactions on Pattern Analysis and Machine Intelligence (submitted)*, 2023.

[20] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[21] 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," *JOSA A*, vol. 31, no. 5, pp. 1049–1058, 2014.

[22] G. D. Finlayson, "Corrected-moment illuminant estimation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 1904–1911.

[23] D. Cheng, B. Price, S. Cohen, and M. S. Brown, "Effective learning-based illuminant estimation using simple features," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1000–1008.

[24] J. T. Barron and Y.-T. Tsai, "Fast fourier color constancy," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 886–894.

[25] H. Gong, "Convolutional mean: A simple convolutional neural network for illuminant estimation," *arXiv preprint arXiv:2001.04911*, 2020.