Color Constancy by Deep Learning

Zhongyu Lou
z.lou@uva.nl
Theo Gevers
Th.Gevers@uva.nl
Ninghang Hu
n.hu@uva.nl
Marcel P Lucassen
marcel@lucr.nl

Intelligent System Laboratory
Amsterdam, University of Amsterdam,
Science Park 904, 1098 XH Amsterdam,
The Netherlands

Abstract

Computational color constancy aims to estimate the color of the light source. The performance of many vision tasks, such as object detection and scene understanding, may benefit from color constancy by using the corrected object colors. Since traditional color constancy methods are based on specific assumptions, none of those methods can be used as a universal predictor. Further, shallow learning schemes are used for training-based color constancy, possibly suffering from limited learning capabilities.

In this paper, we propose a new framework using Deep Neural Networks (DNNs) to obtain accurate light source estimation. We reformulate color constancy as a DNN-based regression approach to estimate the color of the light source. The model is trained using datasets of more than a million images. Experiments show that the proposed algorithm outperforms the state-of-the-art by 9%. Especially in cross dataset validation, our approach reduces the median angular error by 35%. Our algorithm operates at more than 100 fps during testing.

1 Introduction

The appearance of the same object under different light sources may vary due to the different color of the light sources. Computational color constancy [1, 2, 4, 5, 10, 13, 14, 16, 22, 27, 30, 34] aims to recover the color of the light source under which an image is recorded and subsequently correct for it such that the same objects appear the same. Many computer vision tasks may benefit from color constancy, such as stereo vision, object recognition, and tracking.

In general, there are two groups of color constancy algorithms. The first group of algorithms are statistics-based methods, such as the Gray-World [2], the White-Patch [24, 26] and the Gray-Edge algorithm [32]. These algorithms are based on a number of imaging assumptions. For example, the Gray-World algorithm assumes that the average reflectance in a scene, from which an image is taken, is gray. The White-Patch algorithm is based on the assumption that the maximum response of the color channels is caused by a perfect white reflector. Because statistics-based algorithms are based on restrictive assumptions of the imaging conditions, they are limited in their applicability. The second group of algorithms

Pages 76.1-76.12
DOI: https://dx.doi.org/10.5244/C.29.76

are learning-based approaches, such as the gamut mapping algorithm [13], the svr-based algorithm [15], neural networks [29] and the exemplar-based algorithm [23]. In general, existing learning-based methods are constrained to shallow learning models based on hand-crafted, low-level visual features such as pixels and edges. However, image features are intrinsically hierarchical and should automatically be learnt from the image data to avoid any bias in (hand-crafted) feature construction.

To this end, instead of learning shallow feature representations, we exploit deep learning architectures by means of Convolutional Neural Networks (CNN). Different from existing methods which rely on predefined low-level features, we propose to use CNNs to learn feature hierarchies to achieve robust color constancy. A deep CNN model is used consisting of eight (hidden) layers. Such a deep model will yield multi-scale image features composed of pixels, edges, object parts and object models. Our deep learning approach needs large amounts of data with ground-truth for training. Unfortunately, there are no such datasets available for color constancy. Therefore, we propose a different training approach. Our approach consists of learning a deep architecture using a sequence of training data for the estimation of the color of the light source. First, a hierarchy of visual features is automatically learnt to capture the essence of image structures of generic images. To this end, the model is trained on the ImageNet, which contains more than 1.2 million images, for the task of image classification. Then, the obtained generic feature presentation is adjusted to the color constancy problem by training on the ImageNet dataset using existing color constancy algorithms to provide the labels. Finally, the obtained deep learning architecture is retrained on existing (publicly available) ground truth datasets. Although, the (off-line) training is computationally expensive, the (online) testing, to estimate the light source color, is real-time (i.e. more than 100 fps).

In summary, the novel contributions are as follows. Our approach 1) is not constrained by any imaging assumption, 2) provides different deep learning frameworks for color constancy, 3) learns image features instead of hand-crafted ones, 4) provides different hierarchical visual features rather than low-level ones, and 5) provides real time color constancy.

2 Color Constancy

For a Lambertian surface, the image value $f_c(x) = \{f_R, f_G, f_B\}$ is defined by the light source $e(\lambda)$, the surface reflection $s(x,\lambda)$ and the camera sensitivity function $c(\lambda)$:

$$f_c(x) = \int_{\omega} e(\lambda)s(x,\lambda)c(\lambda)d\lambda, \tag{1}$$

where ω is the visible spectrum of the wavelength λ, $c = \{R,G,B\}$ and x is the spatial coordinate. With the assumption that the recorded color of the light source depends on the color of the light source $e(\lambda)$ and the camera sensitivity function $c(\lambda)$, the color of the light source is defined as follows:

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

Following the literature [12, 17, 20], we use the diagonal model to represent the change of the color of the light source:

$$\begin{pmatrix} R_u \\ G_u \\ B_u \end{pmatrix} = \begin{pmatrix} e_R & 0 & 0 \\ 0 & e_G & 0 \\ 0 & 0 & e_B \end{pmatrix} \begin{pmatrix} R_c \\ G_c \\ B_c \end{pmatrix}, \tag{3}$$

where $\{R_u, G_u, B_u\}$ is the color taken under a unknown light source, $\{R_c, G_c, B_c\}$ is the transformed color so as it appears as it has been recorded by a canonical light source. $\{e_R, e_G, e_B\}$ is the color of the light source to be estimated. In this paper, we use perfect white (i.e. $(\frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}})^T$ as the canonical illuminant. To obtain the image under the canonical illuminant, the diagonal model is used.

3 Deep Learning for Color Constancy

A three dimensional vector $\mathbf{Y} = (y_R, y_G, y_B)^T$ is used to represent the light source. An image with a labeled ground-truth light source is denoted by $(x, \mathbf{Y})$ where x is the image data and $\mathbf{Y}$ is the light source. Note that the light source used in our algorithm is normalized as follows:

$$\mathbf{Y} \longleftarrow \sqrt{3}\frac{\mathbf{Y}}{\sqrt{y_R^2 + y_G^2 + y_B^2}}, \tag{4}$$

ensuring that light sources with different intensities have the same scale.

3.1 DNN-based Regression

We formulate color constancy as a regression problem:

$$\hat{\mathbf{Y}} = \psi(x; \theta), \tag{5}$$

where $\hat{\mathbf{Y}}$ is the output estimation of the light source, x is the input image, ψ is the model and θ represents the parameters of the model. In this paper, model ψ is a deep convolutional neural network. We build model ψ based on the architecture of [25] as it provides outstanding performance for image classification and object localization. Our architecture has eight layers including five convolutional and three fully connected layers. The input of the first layer of the model is an image with a predefined size which is equal to the number of pixels multiplied by the number of channels (i.e. 3). Hence, the model uses raw images as input. No hand crafted features are extracted beforehand. As shown in Figure 1, the last layer of the model is the output target value of the regression which is a three dimensional vector.

The model consists of eight layers denoted by $C1$, $C2$, $C3$, $C4$, $C5$, $F6$, $F7$, $F8$. The first five layers are convolutional layers. The last three layers are fully collected layers. The last layer $F8$ is the output of the model which has three dimensions. Combining all the layers, the total number of parameters in this model is very large. Therefore, large scale datasets with ground-truth light source labels are required to directly apply this model to the color constancy problem. However, such large scale dataset are not available. To this end, we propose an alternative training procedure consisting of different training steps in the following section.

3.2 Sequential Training

In this section, we discuss our sequential training procedure. In the first step, the model is trained on the dataset of ImageNet LSVRC-2010 [7] which contains 1.2 million images. The

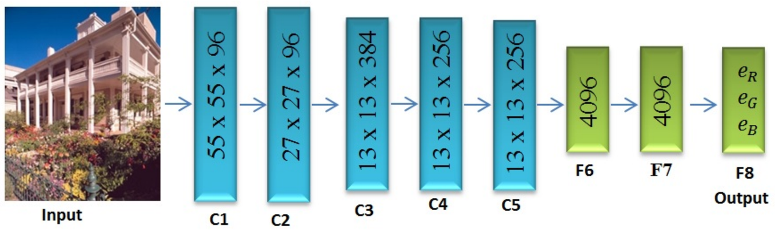


Figure 1: The architecture of the convolutional neural network. This architecture has eight layers in total. The first five layers are convolutional layers and the last three layers are fully connected layers. The input of the model is a raw image with three channels (i.e. R, G, B). The output of the model is a three dimension vector corresponding to the estimated color of the light source. The numbers in the figure represent the output dimensions of each layer.

aim is to generate a hierarchy of visual features to encompass image structures of generic images. Then, this model is adjusted to the color constancy problem by retraining on the ImageNet dataset using labels computed by existing color constancy algorithms. Finally, the obtained deep learning architecture is retrained on (real, but smaller) ground-truth datasets.

3.2.1 $Net1$: Training on ImageNet using Object Labels

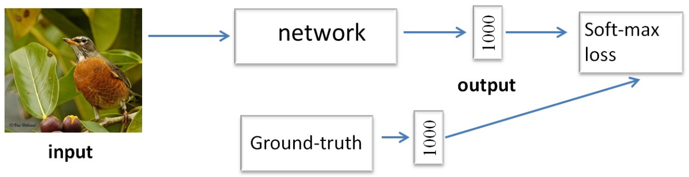


Figure 2: The first step of our training process: Training on ImageNet using Object Labels.

The ImageNet dataset contains 1.2 millions of labeled images. Each image has a label indicating which object is present in the image. There are 1000 object categories. In this step, we train the model on ImageNet to derive features for object description. In this way, a rich and generic feature hierarchy is learnt to capture the complex visual patterns in generic, real-world images. As shown in Figure 2, the 'network' is the first seven layers of the model shown in Figure 1. The last layer is replaced by a 1000 dimensional vector. The soft-max loss function is used for training. The aim of the first training step is to obtain a pre-trained feature model representing general images. Since the ImageNet dataset contains 1000 object categories, it provides a variant back-propagation information module for training. We denote the parameters obtained by training on ImageNet as the $Net1$ network.

3.2.2 $Net2$: Training on ImageNet with $Net1$ Parameters using Labels from Existing Color Constancy Methods

Now, the obtained model is retrained on ImageNet in the context of color constancy. Images in ImageNet do not have corresponding ground-truth light source labels. Therefore, existing color constancy methods are used to estimate the color of the light source $\hat{\mathbf{Y}}$ for each image in ImageNet. As shown in Figure 3, the estimated light source $\hat{\mathbf{Y}} = (\hat{y}_r, \hat{y}_g, \hat{y}_b)^T$ is used as label to train the model. We use the Euclidian loss, which results in the following optimization

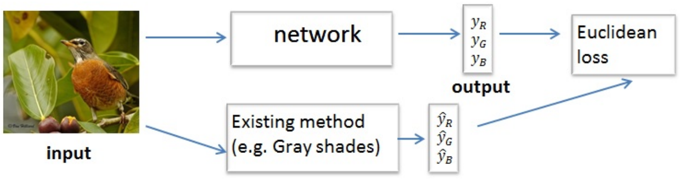

network
y_R y_G y_B
output
Euclidean loss
input
Existing method (e.g. Gray shades)
$\hat{y}_R$ $\hat{y}_G$ $\hat{y}_B$

Figure 3: The second step of our training process: Training on ImageNet with $Net1$ Parameters using labels of existing color constancy methods.

function:

$$\arg\min_{\theta}\sum_{i}||\hat{\mathbf{Y}}_i-\psi(x_i;\theta)||_2^2, \tag{6}$$

where $\hat{\mathbf{Y}}_i$ is the estimated light source for the $i-th$ image x_i using different existing color constancy methods such as the gray-shades, gray-edge or gamut mapping algorithm. $\psi(x_i;\theta)$ is the light source prediction using the model, as defined by Equation 5.

In this stage, the aim is to retrain and adjust the parameters of $Net1$ for the purpose of color constancy. We perform retraining of the (initial) parameters of $Net1$ based on light source estimation obtained by existing color constancy algorithms as labels. Although any other or combination of color constancy algorithms can be used to generate the labels for the ImageNet dataset, the gray-shades and gray-edge algorithms are used due to their efficiency and good performance. The resulting sets of parameters are denoted by $Net2-GrayShades$ and $Net2-GrayEdge$ respectively. The obtained feature representations $Net2-GrayShades$ and $Net2-GrayEdge$ are merely adopted (color constancy) versions of $Net1$. It is hypothesized that the obtained models $Net2-GrayShades$ and $Net2-GrayEdge$ replicate the performance of the color constancy methods used to provide the labels i.e. the gray-shades and gray-edge algorithms.

3.2.3 $Net3$: Retraining $Net2$ parameters on datasets with real ground-truth labels

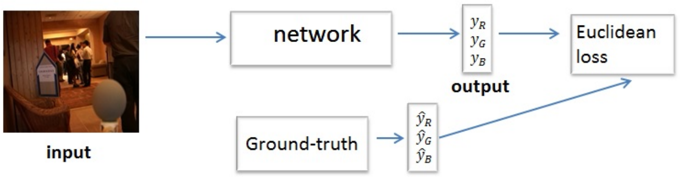


Figure 4: The parameters of the deep model, corresponding to the coefficients of the feature hierarchy obtained in the previous learning stages, are fine-tuned using existing (publicly available) datasets with ground-truth label sets (e.g. Grayball [6] and ColorChecker [19]).

The parameters of the obtained deep model, corresponding to the coefficients of the feature hierarchy obtained in the previous learning stages, are retrained using existing (publicly available) datasets with (real) ground-truth label sets (e.g. Grayball [6] and ColorChecker [19]). In these datasets, the ground-truth color of the light source is given for each image under which it has been recorded. In the experiments, it will be shown that $Net2-GrayShades$ outperforms $Net2-GrayEdge$. Therefore, we use $Net2-GrayShades$ as initial parameters to retrain the model with a Euclidian loss. The parameters of this network are denoted by $Net3$.

4 Experiments

In this section, we assess the performance of the proposed deep learning framework and compare it with state-of-the-art color constancy algorithms.

4.1 Dataset and Evaluation Criterion

Two standard benchmark datasets, the Grayball [6] and the ColorChecker [19], are used. The Grayball dataset contains $11,346$ real-world images. In each image, a gray ball is placed in the right-bottom of the image to obtain the ground-truth light color. During training and testing, the gray ball has been removed from the image. Gamma correction has been applied. The ColorChecker dataset contains 568 real-world images. Since each image has a ColorChecker placed in the scene, the illuminant ground truth is known. The ColorChecker has been removed from the image during training and testing.

To evaluate the performance of the different algorithms, the angular error is used:

$$\varepsilon = \cos^{-1}(\hat{e}, e), \tag{7}$$

where e is the ground-truth light source and $\hat{e}$ is the estimated one. The mean, median and standard deviation of the angular errors are reported for each algorithm.

For the Grayball [6] dataset, following previous papers, we split the dataset into 15 subsets. Each time, one subset is used as the testing set and all the remaining images are used as training sets. The experiment is completed after each subset has been used as testing set. For the ColorChecker [19] dataset, we split the dataset into 3 subsets. Each time, one subset is used as testing and the other two sets are used as training. The final result is reported by averaging the result for each image.

4.2 Experiment 1: Grayball Dataset

As stated above, for the Grayball dataset, 15-fold cross validation is used. We report on the results of Gray-Shades, Gray-Edge, the proposed model trained on ImageNet (i.e. $Net2 - GrayShades$ and $Net2 - GrayEdge$ in Section 3.2.2) and the proposed model retrained on the Grayball dataset (i.e. $Net3$ in Section 3.2.3).

Table 1: The results on the Grayball dataset.

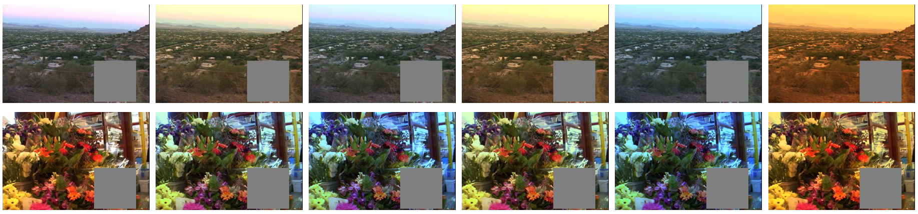
Methods	mean	median	std
Gray-shades [11]	5.4°	4.6°	3.8°
$Net2 - GrayShades$ (This paper)	5.5°	4.7°	3.8°
$Net3 - GrayShades$ (This paper)	**4.8°**	**3.7°**	3.9°
Gray-edge [32]	6.2°	4.6°	5.0°
$Net2 - GrayEdge$ (This paper)	6.5°	5.6°	4.4°
$Net3 - GrayEdge$ (This paper)	**5.2°**	**3.9°**	4.5°

As described in section 3.2.2, we use the resulting parameters of the Gray-Shades and Gray-Edge as labels to retrain the model on ImageNet to obtain $Net2 - GrayShades$ and $Net2 - GrayEdge$ respectively. Table 1 shows that $Net2 - GrayShades$ and $Net2 - GrayEdge$

have similar performance as the Gray-shades respectively the Gray-Edge algorithm. As hypothesized, the obtained convolutional neural network $Net2$ replicates the performance of the color constancy method(s) used to provide the labels. Then, after retraining $Net2 - GrayShades$ on real data (i.e. Grayball dataset) the obtained deep model $Net3$ significantly outperforms the previous CNN's. In the remainder, $Net2$ is equal to $Net2 - GrayShades$, because $Net2 - GrayShades$ outperforms $Net2 - GrayEdge$.

In conclusion, the proposed deep learning approach is able to automatically learn feature hierarchies to capture the essence of visual patterns in images to achieve color constancy.

4.3 Experiment 2: Extended Grayball Dataset: Data Argumentation



(a) GT(0°) (b) 4.14° (c) 6.80° (d) 11.22° (e) 22.78°

Figure 5: A number of images generated by applying simulated light sources on the original images.

Table 2: Results obtained on the Grayball dataset. *NetFinal* is trained on the extended Grayball dataset.

Methods	mean	median	std
Elfiky et al. (TIP 2014) [9]	5.4°	4.5°	-
Prinet et al. (ICCV 2013) [28]	5.4°	4.6°	-
Gao et al. (ECCV 2014) [18]	6.0°	5.1°	-
Joze et al. (PAMI 2014) [23]	4.4°	3.3°	-
Net3 (This paper)	4.8°	3.7°	3.9°
NetFinal (This paper)	**3.9°**	**3.0°**	3.3°

Deep learning approaches greatly benefit from large training datasets. Since ground-truth labels are expensive to get (especially for the color constancy problem), data augmentation is widely exploited by different deep learning approaches. For example, in image denoising, extra training data is generated by applying simulated noise [33]. In deep learning-based image classification, Krizhevsky et.al [25] use image translations and horizontal reflections to generate more training images.

Inspired by these data augmentation methods, in this paper, we use data augmentation to obtain more training data for color constancy. Specifically, for each training image, we correct for the color of the light source using the diagonal model of the ground-truth. Using Eq. 3, the canonical image is obtained, as shown in Fig. 5(a). Then, simulated light sources can be applied to the corrected image using the diagonal model in Eq. 3. Any simulated light source color can be used i.e. the Spectral Power Distribution (SPD) of different light sources such as tungsten halogen, fluorescent lamp, high pressure sodium, or daylight. However, many of them are less frequently present than others depending on the scenes from which the images are recorded. Therefore, in this paper, simulated light sources are derived from the training dataset. By clustering the ground-truth light color of the training set into k clusters, we obtain k simulated light sources by collecting the means of each cluster. In this way, per image, k additional training images are obtained with different ground-truth illuminant ($k = 10$ in our experiments). Note that data augmentation is only performed on the training images. We test the algorithm on the original testing images. We denote the proposed model trained on the extended Grayball dataset as $NetFinal$. Fig. 5 shows a number of images obtained by the proposed data augmentation method.

It can be derived from Table 2, that our deep learning model $NetFinal$ significantly outperforms all previous CNN's models, and all existing state-of-the-art algorithms. It reduces the median angular error by 9%, from 3.3 ([23]) to 3.0. Further, our algorithm is more efficient and general than Joze et al. [23]. Their method is computational expensive due to e.g. image segmentation. Furthermore, the method of Joze et al. [23] requires nearest neighbour classification for each segment of the testing image. Our algorithm processes images in more than 100 fps with GPU implementation (i.e. real-time color constancy).

4.4 Experiment 3: ColorChecker Dataset: Small Dataset

In this section, the proposed deep learning approach is evaluated and compared to state-of-the-art algorithms on the ColorChecker [19] dataset. During training, the parameters of $Net2$ (in section 3.2.2) obtained by training on imagenet is used as the initial parameters of the network. Then, we retrain the parameters on the ColorChecker [19] dataset to obtain the final CNN, denoted by *NetColorChecker*. No data augmentation is used. Further, the ColorChecker dataset is small. The average results are reported in Table 3. We only report the mean and median results for algorithms that did not report their standard deviation in their papers. It can be derived that the proposed algorithm obtains similar results in comparison to [23], but the proposed method outperforms all the other algorithms.

Table 3: Results obtained for the ColorChecker dataset. *NetColorChecker* is trained on the Imagenet dataset and retrained on ColorChecker.

Methods	mean	median	std
Drew et al. (ECCV 2012) [8]	4.1°	2.8°	-
Gijsenij et al. (IJCV 2008)[21]	4.1°	2.5°	-
Chakrabarti et al. (PAMI 2012) [3]	3.7°	3.0°	-
Weijer et al. (ICCV 2007) [31]	3.5°	2.5°	-
Gao et al. (ECCV 2014) [18]	3.4°	2.6°	-
Joze et al. (PAMI 2014) [23]	**3.1°**	**2.3°**	-
NetColorChecker (This paper)	**3.1°**	**2.3°**	3.3°

4.5 Experiment 4: Cross Dataset Validation

In this section, we evaluate the generalization capabilities of the proposed algorithm. To this end, the model is trained on the extended Grayball dataset [6], (i.e. $NetFinal$ in section 4.3), and tested on the ColorChecker dataset [19]. None of the previous papers have reported on the cross (inter) dataset except for Joze et al. [23]. From Table 4, it can be derived that our algorithm significantly outperforms the method of [23]. The error reduction is 35% in terms of median angular error. This is an indication that the proposed CNN approach is able to learn generic feature hierarchies to achieve robust color constancy.

Table 4: Results obtained on the ColorChecker dataset. Our model and the model of Joze et al. [23] are trained on the Grayball dataset and tested on the ColorChecker. Our *NetFinal* is the fine-tuned model on Extended Grayball as shown in section 4.3.

Methods	mean	median	std
Joze et al. (PAMI 2014) [23]	6.5°	5.1°	-
NetFinal (This paper)	**4.7°**	**3.3°**	5.3°

4.6 Experiment 5: Feature Visualization

Different from existing methods which rely on predefined low-level features, the proposed CNN learn multi-scale features to achieve robust color constancy. To illustrate the inclusion of higher-order image structures, in Figure 6, the F7 hidden unit is shown. F7 features are extracted first from each image of the Grayball dataset. Per one dimensional feature, we cluster the features into groups. Then, the average image of the Grayball for each group is shown. As shown in Figure 6, it can be derived that a number of high level image structures are indicated by the hidden unit of F7.

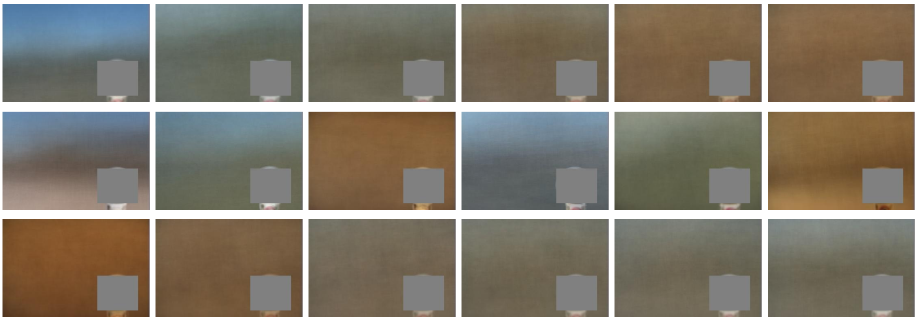


Figure 6: Visualization of hidden unit of F7. Each row visualizes one dimension of F7 shown in figure 1.

4.7 Experiment 6: Training Scheme Validation

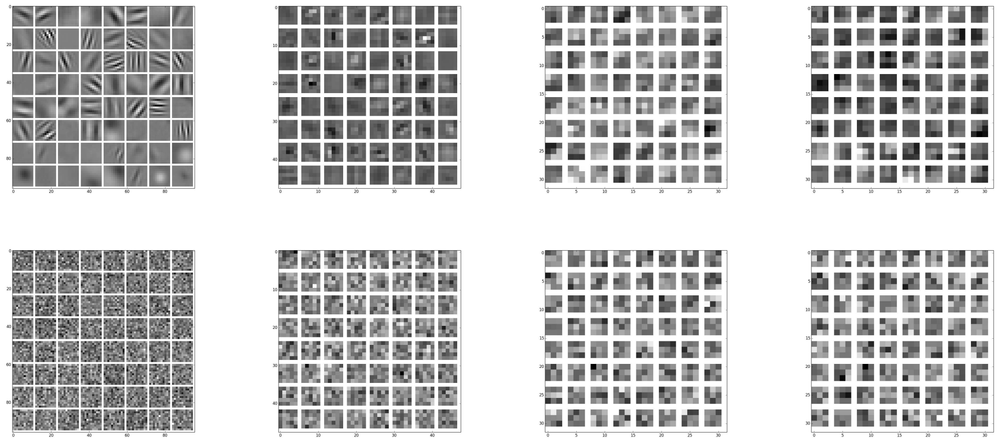
Because of the lack of large scale training datasets, we have proposed a three-step learning strategy to train CNNŠs to estimate the color of the light source. To validate this approach, the influence of the pre-training steps is investigated. Therefore, two experiments are conducted without pre-training. First, instead of using the parameters obtained in the first step as the initial parameters, we directly train the network on the ImageNet dataset with generated light sources. For the second experiment, instead of using the parameters obtained in the second step as the initial parameters, we directly train and evaluate the obtained network on the Grayball dataset.

As shown in 5, the model (directly trained on ImageNet i.e. without pre-training) with light source generated by Gray-shades does not perform as good as the Gray-shades algorithm. This implies that the model (directly trained on ImageNet) is not sufficiently able to learn proper features for color constancy. In Figure 7, we show the parameters learned with pre-training and without pre-training. For the parameters with pre-training, clear semantic,

Table 5: Results on the Grayball dataset without pre-training.

Methods	mean	median	std
Gray-shades [□]	5.4°	4.6°	3.8°
Directly trained on ImageNet	6.7°	5.6°	4.6°
Directly trained on Grayball	8.7	7.1	4.9

hierarchical image patterns are obtained. However, for the parameters without pre-training, the patterns are noisy and less distinctive. In conclusion, our three-step training strategy is a valid approach to compute proper image features (for color constancy) which are intrinsically hierarchical and represent the underlying image structures.



(a) conv1 (b) conv2 (c) conv3 (c) conv4

Figure 7: Visualization of parameters learned with and without pre-training. The first row is parameters learned with pre-training and the second row is parameters without pre-training.

5 Conclusion

Previous color constancy algorithms are limited by their assumptions, hand-crafted features and shallow learning models. Therefore, for the first time, we have introduced deep learning for color constancy enabling deep feature representations.

Our experiments show that the deep learning framework obtains accurate results on real-world datasets. For inter dataset cross validation, it is demonstrated that the proposed algorithm outperforms the state-of-the-art. Our algorithm is very efficient as it processes images with 100 fps.

References

[1] S. Bianco, C. Cusano, and R. Schettini. Color constancy using cnns. *In Computer Vision and Pattern Recognition Workshops (CVPRW), on Deep Vision: Deep Learning in Computer Vision*, 2015.

[2] G. Buchsbaum. A spatial processor model for object colour perception. *Journal of the Franklin Institute*, 1980.

[3] A. Chakrabarti, K. Hirakawa, and T. Zickler. Color constancy with spatio-spectral statistics. *PAMI*, 2012.

[4] 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*, 2014.

[5] A. Choudhury and G. Medioni. Color constancy using denoising methods and cepstral analysis. *Intl, Conf. on Image Prossing (ICIP)*, 2009.

[6] F. Ciurea and B. Funt. A large image database for color constancy research. *Proceedings of the Eleventh Color Imaging Conference*, 2003.

[7] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. *CVPR*, 2009.

[8] M. Drew, H. Vaezi Joze, and G. Finlayson. Specularity, the zetaimage, and information-theoretic illuminant estimation. *ECCV*, 2012.

[9] N. Elfiky, T. Gevers, A. Gijsenij, and J. Gonzalez. Color constancy using 3d scene geometry derived from a single image. *IEEE TIP*, 2014.

[10] G. Finlayson and S. Hordley. Selection for gamut mapping colour constancy. *In Proceedings of British Machine Vision Conference (BMVC)*, 1997.

[11] G.D. Finlayson and E. Trezzi. Shades of gray and colour constancy. *Proceedings of the Color Imaging Conference*, 2004.

[12] G.D. Finlayson, M.S. Drew, and B.V. Funt. Spectral sharpening: Sensor transformations for improved color constancy. *J. Optical Soc. of Am. A*, 11:1553–1563, 1994.

[13] G.D. Finlayson, S.D. Hordley, , and I. Tastl. Gamut constrained illuminant estimation. *IntâĂŹl J. Computer Vision*, 2006.

[14] Graham Finlayson. Corrected-moment illuminant estimation. *ICCV*, 2013.

[15] B. Funt and W. Xiong. Estimating illumination chromaticity via support vector regression. *Color and Imaging Conference*, 2004.

[16] B.V. Funt. Color constancy in digital image. *Intl, Conf. on Image Prossing (ICIP)*, 1999.

[17] B.V. Funt and B.C. Lewis. Diagonal versus affine transformations for color correction. *J. Optical Soc. of Am. A*, 17:2108–2112, 2000.

[18] S. Gao, W. Han, K. Yang, C. Li, and Y. Li. Efficient color constancy with local surface reflectance statistics. *ECCV*, 2014.

[19] P.V. Gehler, C. Rother, A. Blake, T. Sharp, and T. Minka. Bayesian color constancy revisited. *CVPR*, 2008.

[20] A. Gijsenij and T. Gevers. Color constancy using natural image statistics and scene semantics. *IEEE TPAMI*, 2011.

[21] A. Gijsenij, T. Gevers, and J. van de Weijer. Generalized gamut mapping using image derivative structures for color constancy. *IJCV*, 2008.

[22] H. Joze and M. Drew. Exemplar-based colour constancy. *Proceedings of the British Machine Vision Conference (BMVC)*, 2012.

[23] H. Joze and M. Drew. Exemplar-based color constancy and multiple illumination. *IEEE TPAMI*, 2014.

[24] H.R.V. Joze and M.S. Drew. White patch gamut mapping colour constancy. *Intl, Conf. on Image Prossing (ICIP)*, 2012.

[25] A. Krizhevsky, I. Krizhevsky, and G. Hinton. Imagenet classification with deep convolutional neural networks. *NIPS*, 2012.

[26] E.H. Land and J.J. McCann. Lightness and retinex theory. *Journal of the Optical society of America*, 1971.

[27] J. Mukhopadhyay and S.K. Mitra. Color constancy in the compressed domain. *Intl, Conf. on Image Prossing (ICIP)*, 2009.

[28] V. Prinet, D. Lischinski, and M. Werman. Illuminant chromaticity from image sequences. *ICCV*, 2013.

[29] R. Stanikunas, H. Vaitkevicius, and JJ. Kulikowski. Investigation of color constancy with a neural network. *Neural Networks*, 2004.

[30] J. van de Weijer Th. Gevers, H. M. G. Stokman. Color constancy from hyper-spectral data. *Proceedings of the British Machine Vision Conference (BMVC)*, 2000.

[31] J. van de Weijer, C. Schmid, and J. Verbeek. Using high-level visual information for color constancy. *ICCV*, 2007.

[32] J. Van De Weijer, T. Gevers, and A. Gijsenij. Edge-based color constancy. *IEEE Transactions on Image Processing*, 2007.

[33] J. Xie, L. Xu, and E. Chen. Image denoising and inpainting with deep neural networks. *Advances in Neural Information Processing Systems (NIPS)*, 2012.

[34] X. Ying, L. Hou, Y. Hou, J. Kong, and H. Zha. Canonicalized central absolute moment for edge-based color constancy. *Intl, Conf. on Image Prossing (ICIP)*, 2013.