Neural Computing and Applications
https://doi.org/10.1007/s00521-023-08487-z

ORIGINAL ARTICLE

Color constancy for non-uniform illumination estimation with variable number of illuminants

Ilija Domislović[1] · Donik Vršnjak[1] · Marko Subašić[1] · Sven Lončarić[1]

Received: 27 October 2022 / Accepted: 14 March 2023

Abstract

Image white-balancing is an integral part of every camera's processing pipeline. White-balancing is used to remove illumination chromaticity from an image. Most research in this field has been limited to images with a single uniform illuminant. In this paper, we introduce a novel method for illumination estimation for situations where the scene is illuminated by a variable number of different illuminants and where the illumination in the scene can be non-uniform. The proposed method uses a lightweight convolutional neural network that achieves state-of-the-art results. The method performs illumination estimation on a patch-by-patch basis. We use the assumption that only one illuminant affects each patch since they are so small. Unlike other such methods, our method uses features extracted from the entire image to perform patch illumination estimation. The paper also shows how the image features improve method accuracy with a minimal increase in complexity. The proposed method has around 42 k parameters, and it was tested on three different cameras from the Large-Scale Multi-Illuminant dataset.

Keywords Color constancy · Image color analysis · Image processing · Illumination estimation · Non-uniform illumination

1 Introduction

The Human Visual system is fascinating, and one of its most interesting features is chromatic adaptation. This feature allows us to perceive an object's color as relatively constant. This means the Human Visual System adapts to the scene illumination chromaticity so that we perceive the color of the object as though it is illuminated by a canonical illuminant, which is usually a perfectly white light. Humans will perceive an object's color as constant, meaning we will perceive a banana as yellow regardless of whether it is seen at dusk, dawn, or high noon. This process is called color constancy, and it is subjective. It ensures a relatively constant perception of an object's color under a diverse set of illumination conditions.

Cameras are unable to perform color constancy as this is an ill-posed problem, and many methods have been developed that try to emulate this feature of the Human Visual System. The process of chromatic adaptation of digital images is also known as white-balancing. The job of a white-balancing algorithm is to remove the effect the scene illumination chromaticity has on a digital image. What this means is the algorithm transforms the image to make it look as though the scene is illuminated by a canonical illuminant.

White-balancing is an image processing step present in all modern digital cameras. A variety of methods have been created to solve the problem, ranging from simple methods that use image statistics [1, 2] to methods that use neural networks [3]. Many white-balancing algorithms can be divided into two steps. In the first step, the scene illumination is estimated, and in the second, the estimated illumination is used to chromatically adapt the image. The first step is the main focus of most research in this area since it

✉ Ilija Domislović
ilija.domislovic@fer.hr

Donik Vršnjak
donik.vrsnak@fer.hr

Marko Subašić
marko.subasic@fer.hr

Sven Lončarić
sven.loncaric@fer.hr

[1] Faculty of Electrical Engineering and Computing, University of Zagreb, Unska 3, 10000 Zagreb, Croatia

Published online: 29 March 2023

is the harder, more complex step. More detail is given in Sect. 2.

The process of white-balancing is also important for computer vision tasks since one of the most important features of objects is their color. Tasks such as object detection, object classification, and object tracking benefit from a constant object color. One such example is face recognition, where the effect of illumination needs to be removed to improve accuracy [4].

The problem with most illumination estimation methods is that they use the assumption that an image is only affected by one uniform illuminant. This assumption is sometimes not true, and such methods fail to provide satisfactory results on multi-illuminant images. A simple example is an indoor room with windows and a lightbulb. The lightbulb is one illuminant, and the sun coming from the windows is the other illuminant.

There are some methods that perform multi-illuminant estimation. Many of them [5, 6] divide the image into patches and perform illumination estimation for each patch independently. Such methods have low complexity, but they also have low accuracy. The lower accuracy is the result of the fact that patches are small, and sometimes they do not contain enough information for proper illuminant estimation. Authors in [7] have used segmentation methods such as U-Net [8]. In such cases, the entire image is used, and the illuminant correction is performed on a pixel-by-pixel basis. They have higher accuracy and higher complexity. In this paper, we propose a lightweight convolutional neural network (CNN) that performs illumination estimation for images that contain a variable number of illuminants that do not need to be uniform. We also perform experiments that show the model achieves state-of-the-art results.

The paper is divided into sections as follows. Section 2 gives a formal problem definition. Section 3 presents some currently existing approaches. Section 4 presents the proposed approach and the used training setup. Section 5 gives an overview of the used dataset. Section 6 shows how we evaluated the proposed method. The performed ablation study is also explained in Sect. 6. In Sect. 7, the obtained results are shown. The proposed method is compared to existing methods. Finally, in Sect. 8 the conclusion and future directions are presented.

2 Color constancy overview

Scene illumination has a substantial effect on the color of an object. The human visual system has the ability to ignore the illumination's chromatic effect so that we can perceive the object's color as relatively constant. Several different methods have been developed for computers to emulate this. One way to achieve this is to divide the process into two steps: the scene illumination estimation step and the removal of illumination chromaticity step. A large number of methods [3, 9] use this approach. Another approach for white-balancing is to directly perform color correction without first creating an illumination map. Some of these methods are based on Retinex theory [1, 10, 11], and some use segmentation models to predict the color-corrected image[7].

2.1 Problem formulation

We represent an image as an array of 3-element vectors called pixels. The three elements of a pixel are the red, green, and blue intensities $f = (f_r, f_g, f_b)$. A common image formation model is the Lambertian model [12].

$$f_c = \int_{\omega} I(\lambda) p_c(\lambda) S(\mathbf{x}, \lambda)\, \mathrm{d}\lambda \tag{1}$$

Pixels RGB intensities depend on illumination color $I(\lambda)$, the camera sensitivity function $p(\lambda) = (p_r(\lambda), p_g(\lambda), p_b(\lambda))$ of the three channels, and surface reflectance $S(\mathbf{x}, \lambda)$ where $\mathbf{x}$ represents the spatial coordinates and λ represents the light wavelength.

A common way to perform chromatic adaptation is to use the von Kriss model [13]. It uses a diagonal matrix to perform color correction separately for each color channel. The model uses the assumption that the camera sensitivity functions of the three channels are independent. This is not always true, but experiments have shown that this gives sufficient results [13].

$$I^{\mathrm{c}} = \Lambda^{\mathrm{u,c}} * I^{\mathrm{u}} \tag{2}$$

I^{c} is the image taken under the canonical illuminant, $\Lambda^{\mathrm{u,c}}$ represents the von Kriss diagonal matrix, $*$ is matrix multiplication, and I^{u} is the image taken under an unknown illuminant. The diagonal matrix can be expressed as:

$$\Lambda^{\mathrm{u,c}} = \begin{bmatrix} \frac{L_{\mathrm{r}}^{\mathrm{c}}}{L_{\mathrm{r}}^{\mathrm{u}}} & 0 & 0 \\ 0 & \frac{L_{\mathrm{g}}^{\mathrm{c}}}{L_{\mathrm{g}}^{\mathrm{u}}} & 0 \\ 0 & 0 & \frac{L_{\mathrm{b}}^{\mathrm{c}}}{L_{\mathrm{b}}^{\mathrm{u}}} \end{bmatrix} \tag{3}$$

where $L_{\mathrm{r}}^{\mathrm{u}}$, $L_{\mathrm{g}}^{\mathrm{u}}$, $L_{\mathrm{b}}^{\mathrm{u}}$ are the red, green, and blue values of the unknown illuminant and $L_{\mathrm{r}}^{\mathrm{c}}$, $L_{\mathrm{g}}^{\mathrm{c}}$, $L_{\mathrm{b}}^{\mathrm{c}}$ are the red, green, and blue values of the canonical illuminant. The canonical illuminant is the white light or a L^{c} value of $(1, 1, 1)^T$.

The problem of illumination estimation is the fact that it is an ill-posed problem. This fact can be explained in an image with a few surfaces. As an example, we can use an image that only contains a purple wall. Without additional information, we cannot tell whether the wall is purple and the scene illumination is white light, whether the wall is white and the illuminant is purple, or if another wall color/illumination color combination has created this particular image of a purple wall.

3 Related work

There are a lot of different methods that are used for white-balancing. The oldest methods use low-level statistical features extracted from an image. These methods use assumptions to simplify the problem of illumination estimation.

One such method is the Grey-World algorithm [2]. Grey-World uses the assumption that the average reflectance of a scene is achromatic. Deviation from the achromatic average is caused by the chromaticity of the illumination. To calculate the illuminant, the method takes the average of each color channel.

The White-patch algorithm [1] is another popular statistical method. It uses the assumption that the maximum response of each channel is caused by perfect reflectance. To calculate the image illuminant, the maximum value of each channel is used.

As shown in [14], these two methods are just special instances of a color constancy framework. This framework was further expanded in [15] which resulted in the Grey-Edge framework. For this method, the assumption is that the average edge difference in a scene is achromatic. This framework can be represented using the following formula.

$$\left(\int \left|\frac{\partial^n f_{c,\sigma}(\mathbf{x})}{\partial \mathbf{x}^n}\right|^p d\mathbf{x}\right)^{\frac{1}{p}} = kL_c^{n,p,\sigma} \tag{4}$$

In the formula, $\mathbf{x}$ represents the spatial coordinates, n the order of the derivative, p is the Minkowski norm, c one of the three color channels R, G, B, k is a scalar for illumination vector normalization, $|\cdot|$ is the Frobenius norm and σ is a scale factor used for the convolution of the image with a Gaussian filter $f_{C,\sigma} = f_C \bigotimes G_\sigma$.

These methods are used for illumination estimation when an image has a single uniform illuminant. They can be adapted for multi-illuminant estimation by dividing the image into small patches and performing illumination estimation on each patch [16]. Since the patches are small, the assumption that each patch is illuminated by a single uniform illuminant is used.

Recently, researchers have shown that learning-based methods produce good illumination estimation results. In [3], they present a convolutional neural network that uses an attention mechanism to perform single-illuminant estimation. In [17], an efficient lightweight convolutional neural network with less than 22k parameters is presented. Both of these methods can only perform single-illuminant estimation, and they will not give satisfactory results in multi-illuminant situations.

There are a couple of learning-based methods that perform multi-illuminant estimation. Both [5] and [6] are methods for single and multiple illuminant estimation. They perform illumination estimation by dividing an image into many small patches. The illuminant is estimated for each patch separately. The illuminants are then combined into a single illumination mask that is used to correct the image.

The major difference between the methods is that [6] uses two neural networks. One neural network outputs multiple illumination estimations for a single patch and the other predicts which estimation is the best. These methods are lightweight, but the problem with the patch-based approach is that since the patches are so small that sometimes the patches do not contain enough information for accurate estimation.

A way to avoid this problem is to use the entire image. The U-Net model [8] was used for this purpose [7]. Here, the illumination is not directly estimated, instead the model outputs the white-balanced image. The problem here is that this model is significantly more complex than the patch-based methods.

In this paper, we present a method that combines the simplicity of patch-based models with the usage of the entire image to perform illumination estimation.

4 Proposed method

Our proposed method is a patch-based illumination estimation model. It divides the images into many small patches, and the illumination is estimated for each patch. Since the patches are small, we use the assumption that each patch has a single uniform illuminant. What differentiates our model from [5] and [6] is that we use features extracted from the entire image to perform illumination estimation for each patch.

To estimate the illumination of a single patch, we use a two-stage convolutional neural network. In the first stage, we encode the image using several different conventional layers. In the second stage, we encode the patch using a single convolutional layer. After that, we combine the encoded image and encoded patch. The combination is achieved by adding the two tensors. The combined tensor is fed to several different convolutional layers, whose output is the illumination of the patch.

This paper is the continuation of research done in [17]. The major contributions of this paper compared to [17] are that the method in this paper can perform illumination estimation on an image with non-uniform illumination and that it can perform illumination estimation regardless of how many illuminants affect the image.

4.1 Model architecture

The proposed model uses two modified instances of One-Net [17]. The first instance is used as the image encoder. The input to the encoder is an image of size 256 × 256 pixels. In the encoder, we removed the last convolutional layer and changed the size of the max-pooling kernel from (8,8) to (4,4). The size of the input patch is 16 × 16 pixels. The second instance of One-Net is used as the patch encoder and for patch illumination estimation. The first layer of the second One-Net is used as the patch encoder. The encoded patch is combined with the encoded image by adding the two tensors. The rest of the layers are the same as One-Net, except for the max pooling layers whose kernel was changed from (8,8) to (2,2). The created model is lightweight and has around 42k parameters. Just like One-Net, all convolutional layers have a kernel size of (1,1). A detailed schematic of the model architecture can be seen in Fig. 1.

4.2 Training setup

To develop the method, Python [18] and Tensorflow 2.9 [19] were used. For training, the loss function from [20] is used.

$$\text{Loss} = \left\| \frac{\text{ill}_{\text{pred}} - \text{ill}_{\text{gt}}}{\text{ill}_{\text{gt}}} \right\|_2 \tag{5}$$

ill_{pred} is the estimated illumination and ill_{gt} is the illumination ground truth. For the optimizer, AdamW [21] with weight decay 5×10^{-5} was used. The model was trained for 400 epochs on the Nvidia RTX A6000 GPU and AMD Ryzen 3960X CPU. To change the learning rate of the optimizer, a cyclical learning rate [22] was used. The maximum learning rate was 1×10^{-3}, and the minimum learning rate was 1×10^{-7}. A half-cycle period of 200 epochs was used, as it provided the best results.

To train the model, each image was resized to 256 × 256 pixels. That image was then divided into 256 patches of size 16 × 16 pixels. Completely black patches were ignored during training and testing. Black image regions contain calibration objects that were used to extract the illumination. Image standardization was performed on the entire image and each patch separately before being fed into the model. The ground-truth for each patch is calculated by taking the average value of the illumination map for that patch. A batch size of 500 was used during training. Random image rotation and random image flipping were used during model training.

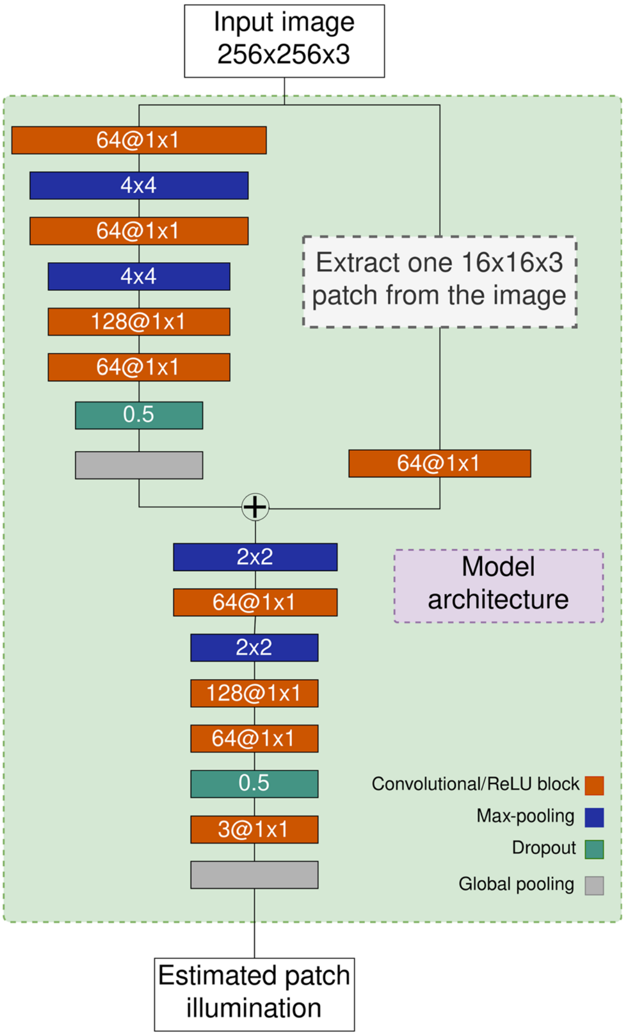

Fig. 1 Visual representation of the model architecture and the parameters used for each layer

5 Dataset

To evaluate our method and compare other methods, we used the LSMI [7] dataset. LSMI is a large-scale dataset that contains 7486 different images taken from 2762 different scenes. To create the dataset, three different cameras were used: Samsung Galaxy Note 20 Ultra, Sony α9, and Nikon D810. The dataset contains images of a variety of different indoor scenes. The images in the dataset contain one to three different illuminants. It contains 3051 two illuminant scenes and 346 three illuminant scenes.

The authors [7] provide a per-pixel illumination mask for each image. To create the mask for two illuminants, the

Table 1 Comparison of results obtained using the different model variants

Model variant				Mean	Std.	Median	Trimean	Best 25%	Worst 25%	95 Percentile
Drop	Conv	Pool	Max*							
–	–	–	(4,4)	1.79	1.91	1.36	1.45	0.39	4.29	5.58
✓	–	–	(4,4)	1.99	1.98	1.43	1.53	0.46	4.50	5.69
–	✓	–	(4,4)	2.74	4.59	1.46	1.64	0.48	7.39	9.10
–	–	✓	(8,8)	2.32	4.34	1.17	1.36	0.37	6.34	7.26
–	–	✓	(4,4)	1.61	1.92	0.95	1.10	0.29	4.03	5.33
✓	✓	–	(4,4)	1.99	2.04	1.29	1.45	0.49	4.70	6.29
–	✓	✓	(8,8)	3.03	4.90	1.69	1.85	0.57	7.95	9.35
–	✓	✓	(4,4)	3.14	4.52	1.77	1.98	0.58	8.20	10.58
✓	✓	✓	(8,8)	2.55	3.40	1.52	1.69	0.50	6.43	8.04
✓	✓	✓	(4,4)	2.09	2.33	1.41	1.52	0.47	4.95	6.32
✓	–	✓	(8,8)	1.94	3.53	1.04	1.16	0.29	5.26	6.08
✓	–	✓	(4,4)	1.57	1.84	0.93	1.06	0.27	3.96	5.00

*Max refers to the Max-pooling kernel size in the feature extractor

Table 2 Comparison of results obtained using all data and only a 25% subset on the galaxy subset

Model	Mean	Std	Median	Trimean	Best 25%	Worst 25%	95 Percentile
Using all patches	2.03	2.15	1.36	1.36	0.48	4.74	5.93
Using 25% of patches	1.57	1.84	0.93	1.06	0.27	3.96	5.00

authors took three images of the scene, two images that are affected by one of the illuminants and an image that is affected by both of the illuminants. The two illuminant scene is simply the sum of the two scenes that are affected by one of the two illuminants. A similar process was used for the three illuminant images, with the difference being the number of images that were taken.

We used this dataset since it is one of the largest datasets for color constancy, with a large diversity of scenes and different illuminations. It also contains non-uniform illumination, which is not common in other datasets. There are other multi-illuminant datasets, but they contain a small number of images. One such dataset is Multiple Light Sources [23]. It contains 68 images, of which 59 were taken in laboratory environments and 9 were real-world images. Laboratory environment images are taken in a dark room with controlled lighting. Such images cannot properly emulate the diverse set of illumination situations that can happen in the real world. The Multiple-Illuminant Multi-Object [24] is another multi-illuminant dataset. It contains 20 real-world images. This is not enough to properly train and test a neural network.

6 Evaluation

Following the LSMI paper [7], we perform an experiment for each camera separately using the train/validation/test split provided by the authors. The original authors used a 70/20/10 train/validation/test. The galaxy subset contains 2500 images, the Nikon subset contains 1988 images, and the Sony subset contains 2998 images. We also divide the

Table 3 Comparison of results obtained only using the information from the patch and results obtained by using information from the patch and the entire image

Model	Mean	Std	Median	Trimean	Best 25%	Worst 25%	95 Percentile
Patch (Galaxy)	3.68	2.71	2.16	2.41	0.70	9.37	11.07
Image-Patch (Galaxy)	1.57	1.84	0.93	1.06	0.27	3.96	5.00
Patch (Nikon)	3.59	3.34	2.03	2.30	0.60	9.34	10.94
Image-Patch (Nikon)	1.53	2.35	0.85	0.97	0.24	4.02	4.76
Patch (Sony)	4.17	5.98	2.42	2.64	0.64	10.91	14.28
Image-Patch (Sony)	1.76	2.83	0.93	1.10	0.25	4.75	5.69

The presented measures were calculated using the angular error of each patch in the dataset, excluding patches that are completely black

Table 4 Comparison of results obtained on the Galaxy phone camera

Method	Single		Multi		Mixed	
	Mean	Median	Mean	Median	Mean	Median
Pix2Pix [27]	6.53	2.17	4.28	2.63	5.66	2.44
Gijsenij et al. [16]	7.49	6.04	12.38	9.57	10.09	7.43
Bianco et al. [5]	4.15	3.30	5.56	4.33	4.89	3.83
HDRNet [26]	2.85	2.20	3.13	2.70	3.06	2.54
U-Net [8]	2.95	1.86	2.35	2.00	2.63	1.91
Proposed method	**1.19**	**0.75**	**2.16**	**1.53**	**1.57**	**0.93**

The proposed method results are bolded

Table 5 Comparison of results obtained on the Nikon camera

Method	Single		Multi		Mixed	
	Mean	Median	Mean	Median	Mean	Median
Pix2Pix [27]	6.10	2.27	4.18	2.76	5.41	2.49
Bianco et al. [5]	3.18	2.61	4.65	4.19	3.93	3.48
HDRNet [26]	2.76	2.43	3.20	3.01	2.99	2.61
U-Net [8]	1.51	1.14	2.36	1.84	1.95	1.45
Proposed method	**1.27**	**0.67**	**1.99**	**1.43**	**1.53**	**0.85**

The proposed method results are bolded

Table 6 Comparison of results obtained on the Sony camera

Method	Single		Multi		Mixed	
	Mean	Median	Mean	Median	Mean	Median
Pix2Pix [27]	4.08	1.72	4.37	3.26	4.20	2.20
Bianco et al. [5]	3.25	2.62	4.38	3.93	3.86	3.19
HDRNet [26]	2.70	2.37	3.65	3.33	3.21	2.89
U-Net [8]	2.83	2.44	3.04	2.78	2.94	2.66
Proposed method	**1.45**	**0.60**	**2.23**	**1.65**	**1.76**	**0.93**

The proposed method results are bolded

Table 7 Comparison of the worst patch illuminant estimation error between the dataset subsets

Dataset subset	Mean	Std	Median	Trimean	Best 25%	75 Percentile	95 Percentile
Galaxy	3.44	3.79	1.85	2.39	0.54	4.99	10.27
Nikon	3.28	3.87	1.71	2.19	0.51	4.55	10.80
Sony	4.06	4.07	2.34	2.97	0.47	6.42	12.00

test set into three subsets for proper comparison with methods in [7]. The three subsets are single illuminant images, multi-illuminant images, and mixed illuminant images.

In LSMI paper [7], they use two types of methods: patch-based methods and methods that use the entire image. The authors show that the results from patch-based methods are significantly worse than the results from the methods that use the entire image. In this paper, we will compare our method to both types of methods even though our method is patch-based. For proper comparison, we used the same image and patch sizes as the authors in LSMI [7]. The used image size is 256×256 pixels, and the used patch size is 16×16 pixels.

To compare different methods, we used the angular distance between the ground-truth illumination vector and the predicted illumination vector. This is a commonly used metric for method comparison.

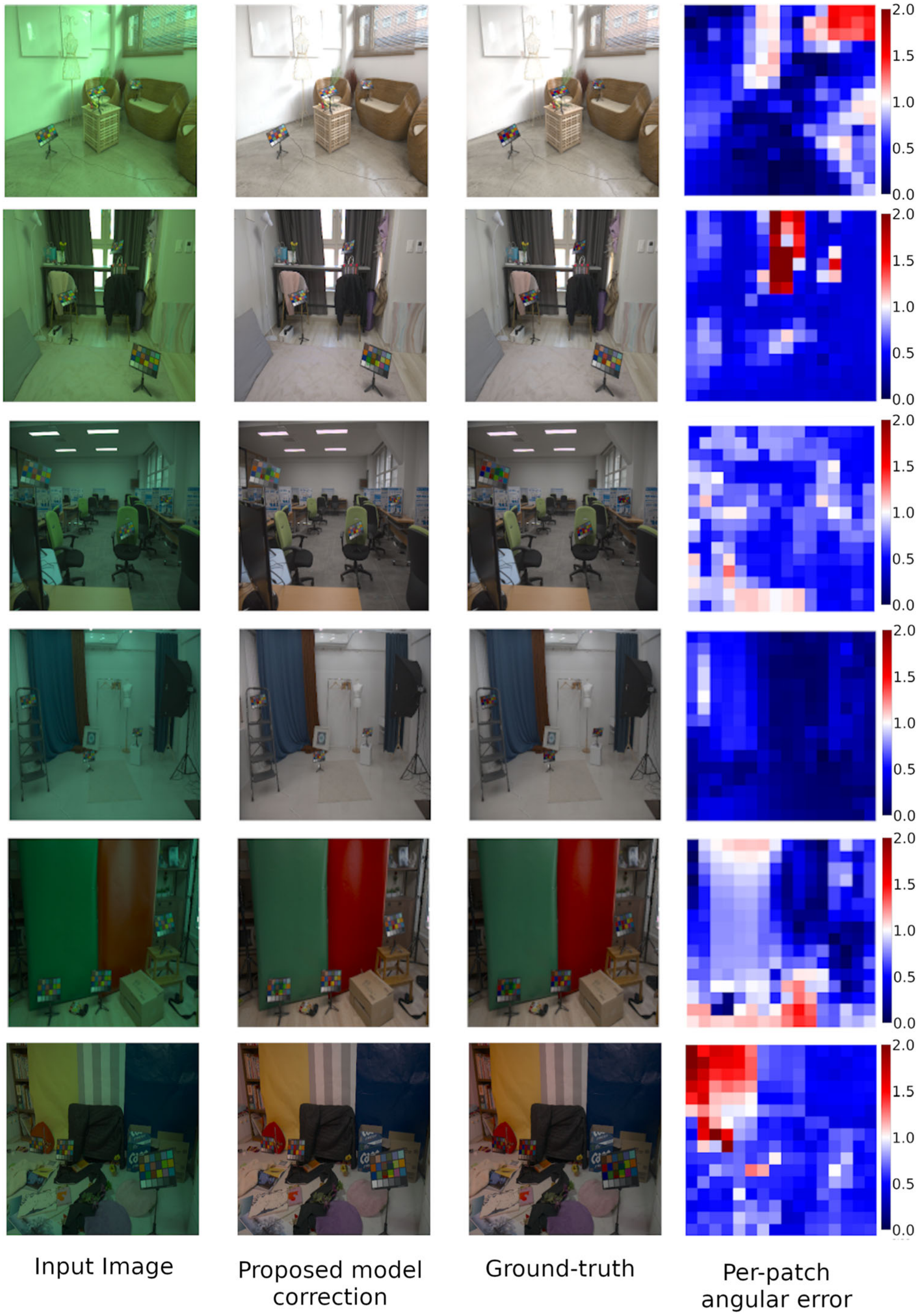

Fig. 2 Visual comparison of model-corrected image and ground-truth corrected image. The angular error for each patch is also shown. The max angular error for each image is around 2°. All images have been tone-mapped for better visualization

$$\text{Angular error} = \cos^{-1}\left(\frac{\mathbf{L}\cdot\hat{\mathbf{L}}}{\|\mathbf{L}\|_2\|\hat{\mathbf{L}}\|_2}\right) \tag{6}$$

$\mathbf{L}$ and $\hat{\mathbf{L}}$ are the ground-truth and predicted illuminations, $\cdot$ is the vector dot product, and $\|\cdot\|_2$ is the Euclidean norm.

The mean, median, best 25% mean, worst 25% mean, and 95 percentile angular error measures are used in the result tables.

6.1 Ablation study

Since we modified the original One-Net [17], we performed an ablation study to see how the method performs when some of the layers of the feature extractor are removed. The layers that were considered were the last three layers of the original One-Net architecture, as they are used to transform the output of the model to an RGB vector that represents the image illuminant prediction. All

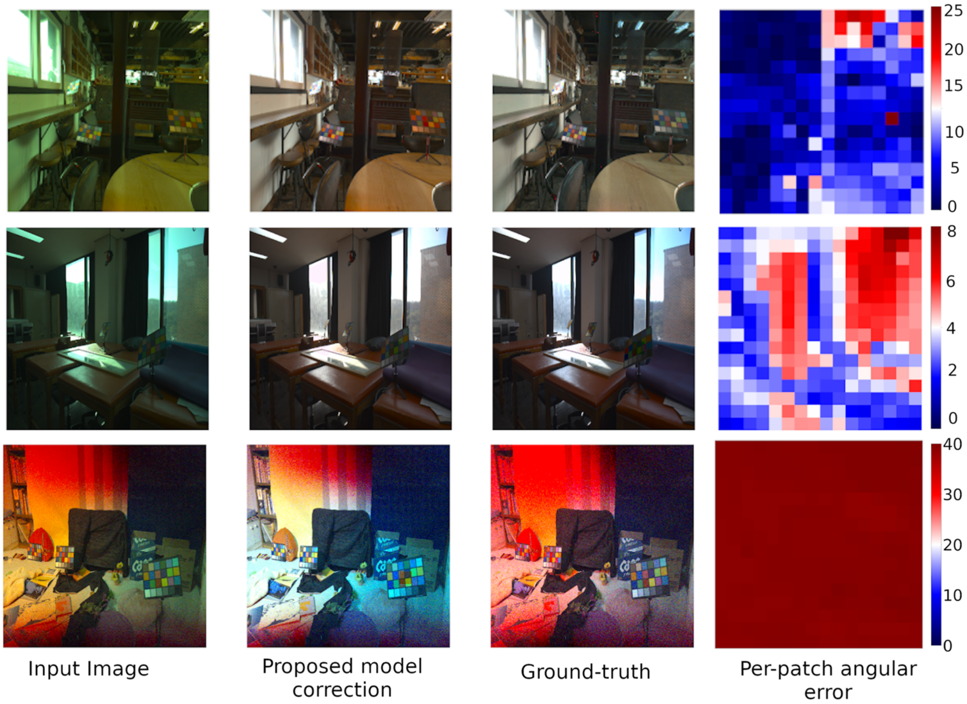

Fig. 3 Showcase of situations where the model gives sub-par results. The angular error for each patch is also shown. The max angular error is different for each image. All images have been tone-mapped for better visualization

other layers from One-Net remained the same. The final three layers are a Dropout layer (Drop), a Convolutional layer (Conv), and a Global-Pooling layer (Pool), in that order. All possible combinations of the final three layers were tested and are presented in Table 1.

For the variants that use the final Convolutional layer, the output of the feature extractor is added to the patch before it is processed by the first Convolutional layer of the estimator. This was done so that the output of the feature extractor can be used for patch illumination estimation. The ablation study was performed on the Galaxy images dataset subset using the same train/validation/test procedure explained in Sect. 5.

Table 1 shows that the proposed model that uses the Drop-Pool layers achieves the best results. During the development of the method, we reduced the Max-pooling kernel from (8,8) to (4,4) so that encoders that do not use the Global-pooling layer can be added to the second One-Net instance. Since the proposed variant uses Global-pooling, we also tested variants with a Max-Pooling kernel of (8,8). Table 1 also shows that using a (4,4) Max-pooling kernel produces the best results and that using kernels of larger size reduces the model accuracy.

During training, only a subset of patches was used. For each epoch, 25% of patches are randomly selected from each image. This was done to decrease training time and to fix overfitting. Since the model is trained end to end, each image is used several hundred times per epoch. Table 2 shows the effect of using only a subset of data for each epoch. It shows that using only 25% of patches improves the results for all angular error measures. The ablation study was performed on the Galaxy images dataset subset using the same train/validation/test procedure explained in Sect. 5.

To test the effectiveness of the image encoder, we performed two experiments. One where we use the image encoder and one where we do not use the image encoder and only information from the patch is used for illumination estimation. The version that does not use an image encoder was trained on patches only. Table 3 shows the angular error of the two methods for each of the three cameras.

In Table 3, we can see that the addition of image features significantly improves the results for all the angular error measures. This is most prominent on the Sony camera, where the angular error of the method that uses both image and patch information is less than half of the angular error of the method that only uses patch information.

The Worst 25% and 95 Percentile show the true benefit of using image information because here we can see the most significant absolute improvement since the Trimean and Median angular error measures fall under the acceptable Human Visual System angular error according to [25].

7 Results

In this section, we compare our results with the results of other methods. We also provide some examples of how our method performs.

Following the LSMI [7] authors, we perform experiments for each camera separately, and we use the train/validation/test split provided with the dataset. A separate model was trained for each camera. Each model has seen images from only one of the cameras. We also divide the test set into three subsets. In the first subset, only images with one illuminant are present. In the second subset, only images with two or three illuminants are present. The final subset contains images with one to three illuminants present.

Tables 4, 5, and 6 compare our method to existing methods. Gijsenij et al. [16] and Bianco et al. [5] are patch-based methods. HDRNet [26], U-Net [8], and Pix2Pix [27] are image-based methods.

Since color constancy can be seen as an image-to-image translation problem when there is non-uniform illumination, we also tested several GAN models. We tested CycleGAN [28], Pix2Pix [27], Pix2PixHD [29], and CUT [30]. Surprisingly, all but Pix2Pix [27] achieved results worse than the worst results in Tables 4, 5, 6. This seems like an interesting research topic, but it goes out of the scope of this paper.

Table 4 shows how our method compares to other methods when looking at an image from the Galaxy phone camera. Our method has less than half the angular error of the patch-based methods. Our method also outperforms all methods when looking at single illuminant images. When looking at the multi-illuminant situation, we can see the result is more similar when compared to the image-based methods, but our method still outperforms all other methods. We also performed statistical analysis to compare our model to the best-performing model from the literature. Using the one-tailed Z-test, we observed a p-value of less than 0.001, concluding that our results are significantly different from the results of U-Net [8].

Table 5 shows how our method compares when looking at Nikon camera images. Here again, our method is significantly better than the patch-based method [5] and image-based methods HDRNet [26], U-Net [8], and Pix2-Pix [27]. For Nikon images, our results are more similar to the result achieved by U-Net, when compared to the other cameras. Also, the result achieved by our method has a smaller accuracy deviation between cameras than U-Net.

Finally, Table 6 compares our method to other methods on Sony camera images. For this dataset subset, our methods outperform all other methods in all angular error measures.

Table 7 shows the angular error measures compiled by only using the worst patch estimation in each image. It shows that the most difficult subset is the Sony images, which have the worst results in almost all angular error measures. We can also compare the results with Table 3 where we can see that the results are comparable with the results obtained by only using patch information for estimation. We can also see that there is a huge jump between the 75 Percentile and 95 Percentile, showing us there are some outlier samples where the model performs poorly. We further explore qualitative results in subsection 7.1.

7.1 Qualitative results

In this subsection, we look into how our method performs on a visual level. We show how images look after we correct them. We also explore in what kind of situations our model fails, how a corrected image looks in such situations, and why the model fails in those situations.

Firstly, we show in Fig. 2 some random images from the dataset. We show a couple of one-illuminant, two-illuminant, and three-illuminant images. We can see from the angular error that there are differences between the ground-truth and the predicted illumination, but these differences are not noticeable. This is consistent with research from [25] that states the human eye cannot distinguish the colors when the angle between the ground truth and prediction is less than 2°. With these examples, we can visualize that our patch-based method can perform accurate illumination estimation even when the patch itself contains only one color, as is the case in many of the showcase images that contain a single color wall.

In Fig. 3, we show three examples where the model cannot properly estimate the illuminant. The first image shows us that the model has difficulty estimating strong artificial illumination. The angular error map confirms this. The upper right area of the image has the worst estimation, and this image region contains artificial lights. The image region that contains a glossy table that reflects the artificial light also has erroneous illumination estimation.

The second situation where the model does not accurately estimate the illumination is in regions that contain highly saturated pixels. In this example, the blue color of the sky gets lost. This is shown in the angular error map. Half of the image illumination is properly estimated, while the other half is not. The area with erroneous results contains highly saturated pixels created by the strong natural outdoor lighting.

The final image shows a situation where the model cannot predict the illumination of any region correctly. This image is illuminated by low-intensity light. The top right part of the image is black, and the rest of the image is full of noisy data. These factors cause the model to perform poorly.

8 Conclusion

In this paper, we propose a lightweight convolutional neural network that only has around 42 000 parameters. We tested the model on the LSMI dataset. We show that our model achieves significantly better results than other patch-based methods. The model also outperforms the per-pixel estimation models in all angular error measures. We show that the addition of global image information significantly improves the accuracy of a patch-based model without significantly increasing the network complexity. We also present qualitative results that show how our model performs and analyzes some situations where the model fails. The analysis shows that the model can predict the patch illumination even when the patch only contains a single one-color surface. In future work, we want to extend the model for cross-camera estimation. We want to develop a method that is camera invariant and can perform accurate illumination estimation on an image from cameras not seen during training.

Data availability The dataset analyzed during the current study is available upon request, https://github.com/DY112/LSMI-dataset [7].

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Land EH (1977) The retinex theory of color vision. Sci Am 237(6):108–129. https://doi.org/10.1038/scientificamerican1277-108
2. Buchsbaum G (1980) A spatial processor model for object colour perception. J Frankl Inst 310(1):1–26. https://doi.org/10.1016/0016-0032(80)90058-7
3. Hu Y, Wang B, Lin S (2017) Fc4: fully convolutional color constancy with confidence-weighted pooling. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 4085–4094 . https://doi.org/10.1109/CVPR.2017.43
4. Wang K, Chen Z, Wu QMJ, Liu C (2019) Face recognition using AMVP and WSRC under variable illumination and pose. Neural Comput Appl 31(8):3805–3818. https://doi.org/10.1007/s00521-017-3316-x
5. Bianco S, Cusano C, Schettini R (2017) Single and multiple illuminant estimation using convolutional neural networks. IEEE Trans Image Process 26(9):4347–4362. https://doi.org/10.1109/TIP.2017.2713044
6. Shi W, Loy CC, Tang X (2016) Deep specialized network for illuminant estimation. In: European conference on computer vision. Springer, pp 371–387 . https://doi.org/10.1007/978-3-319-46493-0_23
7. Kim D, Kim J, Nam S, Lee D, Lee Y, Kang N, Lee H-E, Yoo B, Han J-J, Kim SJ (2021) Large scale multi-illuminant (lsmi) dataset for developing white balance algorithm under mixed illumination. In: 2021 IEEE/CVF international conference on computer vision (ICCV), pp 2390–2399. https://doi.org/10.1109/ICCV48922.2021.00241
8. Ronneberger O, Fischer P, Brox T (2015) U-net: convolutional networks for biomedical image segmentation. In: Navab N, Hornegger J, Wells WM, Frangi AF (eds) Medical image computing and computer-assisted intervention - MICCAI 2015. Springer, Cham, pp 234–241. https://doi.org/10.1007/978-3-319-24574-4_28
9. Xiao J, Gu S, Zhang L (2020) Multi-domain learning for accurate and few-shot color constancy. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 3258–3267. https://doi.org/10.1109/CVPR42600.2020.00332
10. Rizzi A, Bonanomi C (2017) Milano Retinex family. J Electron Imaging 26(3):1–7. https://doi.org/10.1117/1.JEI.26.3.031207
11. Banić N, Lončarić S (2013) Light random sprays retinex: exploiting the noisy illumination estimation. IEEE Signal Process Lett 20(12):1240–1243. https://doi.org/10.1109/LSP.2013.2285960
12. Gijsenij A, Gevers T, Van De Weijer J (2011) Computational color constancy: survey and experiments. IEEE Trans Image Process 20(9):2475–2489. https://doi.org/10.1109/TIP.2011.2118224
13. von Kries J (1905) Influence of adaptation on the effects produced by luminous stimuli. handbuch der Physiologie des Menschen 3:109–282. https://doi.org/10.1016/0016-0032(80)90058-7
14. Finlayson GD, Trezzi E (2004) Shades of gray and colour constancy. In: Color and Imaging Conference. Society for Imaging Science and Technology, vol 2004, pp 37–41. https://ueaeprints.uea.ac.uk/id/eprint/23682
15. Van De Weijer J, Gevers T, Gijsenij A (2007) Edge-based color constancy. IEEE Trans Image Process 16(9):2207–2214. https://doi.org/10.1109/TIP.2007.901808
16. Gijsenij A, Gevers T, van de Weijer J (2012) Improving color constancy by photometric edge weighting. IEEE Trans Pattern Anal Mach Intell 34(5):918–929. https://doi.org/10.1109/TPAMI.2011.197
17. Domislović I, Vršnak D, Subašić M, Lončarić S (2022) One-net: convolutional color constancy simplified. Pattern Recognit Lett 159:31–37. https://doi.org/10.1016/j.patrec.2022.04.035
18. Van Rossum G, Drake FL (2009) Python 3 reference manual. CreateSpace, Scotts Valley, CA
19. Abadi M, Agarwal A, Barham P, Brevdo E, Chen Z, Citro C, Corrado GS, Davis A, Dean J, Devin M, Ghemawat S, Goodfellow I, Harp A, Irving G, Isard M, Jia Y, Jozefowicz R, Kaiser L, Kudlur M, Levenberg J, Mané D, Monga R, Moore S, Murray D, Olah C, Schuster M, Shlens J, Steiner B, Sutskever I, Talwar K, Tucker P, Vanhoucke V, Vasudevan V, Viégas F, Vinyals O, Warden P, Wattenberg M, Wicke M, Yu Y, Zheng X (2015) TensorFlow: large-scale machine learning on heterogeneous systems. Software available from tensorflow.org. https://www.tensorflow.org/
20. Li Z, Ma Z (2021) Robust white balance estimation using joint attention and angular loss optimization. In: Thirteenth international conference on machine vision. International Society for Optics and Photonics, vol 11605, p 116051. https://doi.org/10.1117/12.2586930
21. Loshchilov I, Hutter F (2017) Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101. https://doi.org/10.48550/ARXIV.1711.05101
22. Smith LN (2017) Cyclical learning rates for training neural networks. In: 2017 IEEE winter conference on applications of computer vision (WACV). IEEE, pp 464–472. https://doi.org/10.1109/WACV.2017.58

23. Gijsenij A, Lu R, Gevers T (2011) Color constancy for multiple light sources. IEEE Trans Image Process 21(2):697–707. https://doi.org/10.1109/TIP.2011.2165219
24. Beigpour S, Riess C, Van De Weijer J, Angelopoulou E (2013) Multi-illuminant estimation with conditional random fields. IEEE Trans Image Process 23(1):83–96. https://doi.org/10.1109/CVPR42600.2020.00332
25. Hordley SD (2006) Scene illuminant estimation: past, present, and future. Color Res Appl: Endorsed by Inter-Society Color Council, The Colour Group (Great Britain), Canadian Society for Color, Color Science Association of Japan, Dutch Society for the Study of Color, The Swedish Colour Centre Foundation, Colour Society of Australia, Centre Français de la Couleur 31(4):303–314 . https://doi.org/10.1002/col.20226
26. Li J, Fang P (2019) Hdrnet: single-image-based hdr reconstruction using channel attention cnn. In: Proceedings of the 2019 4th international conference on multimedia systems and signal processing. ICMSSP 2019. Association for Computing Machinery, New York, NY, USA, pp 119–124. https://doi.org/10.1145/3330393.3330426
27. Isola P, Zhu J-Y, Zhou T, Efros AA (2017) Image-to-image translation with conditional adversarial networks. In: 2017 IEEE conference on computer vision and pattern recognition (CVPR), pp 5967–5976. https://doi.org/10.1109/CVPR.2017.632
28. Zhu J-Y, Park T, Isola P, Efros AA (2017) Unpaired image-to-image translation using cycle-consistent adversarial networks. In: 2017 IEEE international conference on computer vision (ICCV), pp 2242–2251. https://doi.org/10.1109/ICCV.2017.244
29. Wang T-C, Liu M-Y, Zhu J-Y, Tao A, Kautz J, Catanzaro B (2018) High-resolution image synthesis and semantic manipulation with conditional gans. In: Proceedings of the IEEE conference on computer vision and pattern recognition
30. Park T, Efros AA, Zhang R, Zhu J-Y (2020) Contrastive learning for conditional image synthesis. In: ECCV

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.