Modeling the Lighting in Scenes as Style for Auto White-Balance Correction

Furkan Kınlı[1] Doğa Yılmaz[2] Barış Özcan[3] Furkan Kıraç[4]
Vision and Graphics Lab, Özyeğin University, Türkiye
{furkan.kinli[1], furkan.kirac[4]}@ozyegin.edu.tr,
{doga.yilmaz.11481[2], baris.ozcan.10097[3]}@ozu.edu.tr

Abstract

Style may refer to different concepts (e.g. painting style, hairstyle, texture, color, filter, etc.) depending on how the feature space is formed. In this work, we propose a novel idea of interpreting the lighting in the single- and multi-illuminant scenes as the concept of style. To verify this idea, we introduce an enhanced auto white-balance (AWB) method that models the lighting in single- and mixed-illuminant scenes as the style factor. Our AWB method does not require any illumination estimation step, yet contains a network learning to generate the weighting maps of the images with different WB settings. Proposed network utilizes the style information, extracted from the scene by a multi-head style extraction module. AWB correction is completed after blending these weighting maps and the scene. Experiments on single- and mixed-illuminant datasets demonstrate that our proposed method achieves promising correction results when compared to the recent works. This shows that the lighting in the scenes with multiple illuminations can be modeled by the concept of style. Source code and trained models are available on `https://github.com/birdortyedi/lighting-as-style-awb-correction`.

1. Introduction

Perceptual systems are generally intended to separate the content and style factors of the observations [63]. Words spoken in an unfamiliar accent, letters written in a novel hand-writing style, or the objects displayed under different lighting conditions can be considered as some examples of the style factors integrated into the content of audio, text or image, respectively. Earlier studies [44, 24, 43, 63] attack this problem with different computational factor models to provide expressive representations of these factors. The term of *factor* is used to represent the well-characterized representation of the observations [35]. Particularly, separating the content from the style in natural images is a challenging problem. Convolutional Neural Networks (CNNs) have the ability to produce generic feature representations, which can be used for independently processing the content and the style of natural images. Previous studies attempt to process the content and the style separately on texture recognition [23] and synthesis [33, 41, 57], classifying the artistic style [48] and filter style [66, 53], style transfer [32, 36, 46], style removal [53] and generative image synthesis [51, 49, 50]. These studies demonstrate that the style representation can be distilled by forming a particular feature space of images via learning objectives.

The concept of style can be interpreted in different ways. For example, it can represent the age of a person, the haircut type and wearing glasses or not as the compact style of the face images [51]. The affine parameters can be extracted by the mapping network where they form the feature maps as the way that packs the different attributes together in the feature space. To achieve this, the mapping network exploits a random vector or the feature vector extracted by pre-trained networks (*e.g.* VGG [62]). On the other hand, the concept of style may refer to the painting style of an artist [32] or the filters applied to the natural images [53]. This time, the affine parameters stand for the correlation between the features, and they can be directly used for manipulating the painting style of an image or removing the filters applied to an image. Based upon these findings, one may argue that any disruptive or modifying factors for the whole image can be modeled as the style factor.

The image signal processor (ISP) applies consecutive processing operations to the raw-RGB sensor image to obtain the standard RGB (sRGB) output image. Some examples of these operations are noise reduction, white-balance (WB), gamma correction, auto-exposure and tone-mapping. WB is one of the earliest ISP operations applied to the raw-RGB sensor image, which normalizes the effect of different lighting conditions in the captured scene [25]. Auto white-balance (AWB) corrects the captured image by estimated illuminant color of the scene, assuming that the illuminant in the scene is global. This operation makes it possible to perceive a particular color in the scene content as the same when viewed under different illuminations, as similar to the

human visual system [40].

Prior works on AWB correction [61, 45, 10, 12] thoroughly focuses on global illuminant estimation. The recent studies [67, 42, 54] achieve significant improvements on this task specialized on single-illuminant scenes. As a common practice, a diagonal-based correction matrix [38] is applied to the images to perform WB. More recently, it is shown that the diagonal correction matrix can be replaced by different static non-linear [6, 5] or learnable [2] functions. Beyond the single-illuminant scenarios, performing single-illuminant AWB algorithms on the scenes illuminated by multiple light sources leads to produce color tint in the sRGB output image. Instead of directly estimating the illumination in the scene, blending the weighting maps of the scenes illuminated by different WB settings [4] is a recently proposed solution introducing a method for increasing the robustness of AWB on mixed-illuminant scenes.

Contribution: In this work, we propose a novel AWB method that models the lighting in single- and mixed-illuminant scenes as style. Our proposed method contains a network that learns the effect of different lighting conditions on the scene with the help of the style information of the scene. Assuming that multiple illuminations in the scene basically stands for the additional style information injected to the scene, our model normalizes the feature maps of the encoder by style information in adaptive manner. Then, it simply employs these normalized feature maps for the learning process of the pixel-wise weighting maps of the same scene with different WB settings. Our AWB strategy does not require to apply any illuminant estimation algorithm, but learns to blend the scene and the pixel-wise weighting maps, as practiced in [4]. We evaluate our method on well-known single illuminant datasets [9, 18], synthetic mixed-illuminant dataset [4] and night photography rendering set. Moreover, we evaluate the performance of our method when changed the patch size for training and the set of WB settings used to generate the weighting maps.

2. Related Works

This section briefly reviews the previous studies on illuminant estimation, WB correction methods and the prior work on learning the style factor.

2.1. Illuminant Estimation

The main aim of the illuminant estimation in the previous studies is to predict the global scene illumination color. In the literature, this problem has been attacked with different strategies. The prior work can be divided into two main categories as statistical methods and learning-based approaches. Statistical illuminant estimation methods mostly use some statistical hypothesis in order to estimate the scene illuminant color. Although these methods are computationally-efficient, they struggle to predict the correct illumination color for real-world scenarios. These methods can be listed as gray-world hypothesis [17], white-patch hypothesis [16], the shades-of-gray [29], the gray-edges [65, 39], the bright-and-dark colors PCA [22], the bright pixels [47], the gray pixel [60] and the grayness index [59]. Recently proposed learning-based methods can produce more accurate results, due to the usage of the information from real-world examples, which better represent the real-world illumination. Learning-based illuminant estimation methods include gamut-based methods [27, 30, 28, 37], Bayesian methods [15, 14, 34, 42], and neural network-based methods [31, 20, 55]. Advanced neural network-based methods further involve different learning strategies such as patch-wise learning [61, 45], achromatic pixel detection [13], metric learning [67], contrastive learning [54], cross-camera illumination estimation [1] and weighting map blending [4].

2.2. WB Correction

Given the estimated illumination color of the scene, a simple diagonal-based correction matrix [38] is employed for white-balancing the raw images. In real-world scenarios, multiple illuminants may occur in the same scene, hence the AWB modules are prone to misinterpret the intensity and the color of the illuminant in particular parts of the scene. This makes WB correction a challenging problem in post-capture. To overcome this problem in multiple illuminant cases, a few attempts are proposed that replaces the diagonal-based correction matrix with a non-linear correction function [3, 5, 7]. Moreover, the recent deep learning-based strategies [2, 4] take in place to perform WB in multiple illuminant scenarios.

2.3. Learning Style Factor

Separating the content and style factors is a well-known topic that aims to process the content and the style information independently. With the help of this idea, the content of an observation can be expressed in a more compact way [63, 44, 24, 43, 63], the style of an observation can be recognized [23, 48, 66] and manipulated to the desired style [32, 36, 46, 53] or even novel contents can be generated with a particular style [33, 41, 51, 49, 50]. For image domain, the style representation can be captured by designing a dedicated feature space of the images [32]. This feature space is mainly built on top of the correlations between different filter responses extracted by a particular layer of a pre-trained network, generally VGG-16 [62]. Intuitively, the style may refer to an abstract concept. Depending on how the learning objective shapes the space, the feature space can model the artistic painting style of an artwork, the hairstyle of a person, the texture of a clothing item, or the color of a cat as style. The prior work [53] demonstrates that image filters

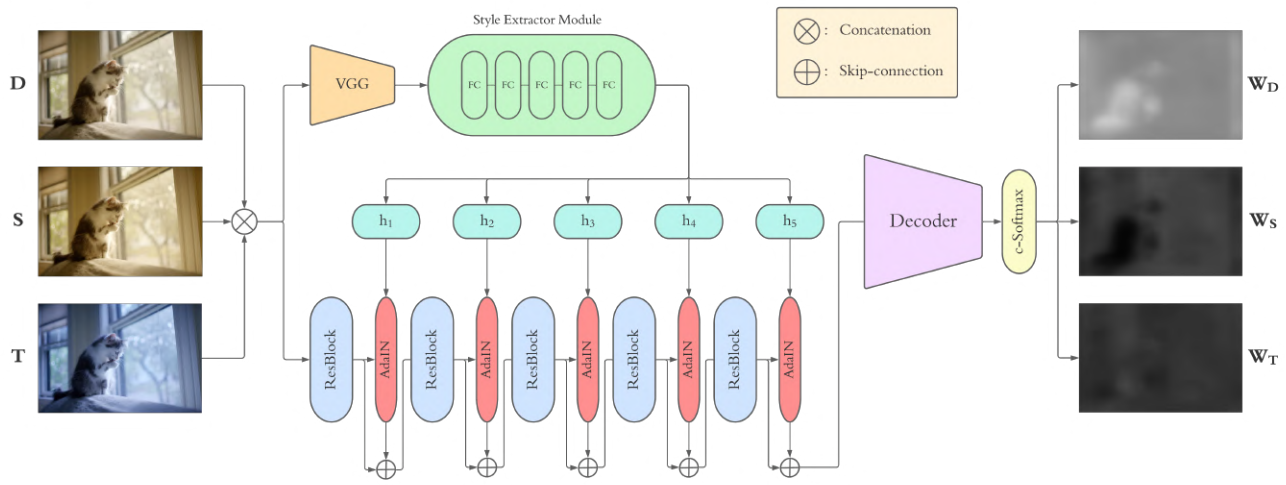


Figure 1: Overall design of proposed learning mechanism for the weighting maps. The latent representations of the images with different WB settings (*i.e.* daylight (**D**), shade (**S**), tungsten (**T**)) are fed into the style extractor module, then the affine parameters are computed, and sent to the corresponding AdaIN layer to discard the external style from the feature maps. The decoder part generates the weighting maps for all WB settings (*i.e.* $\mathbf{W}_D$, $\mathbf{W}_S$, $\mathbf{W}_T$).

that corrupt the original version of the image can be modeled as style. From this point of view, we propose a method that models the lighting in the scenes with single or multiple illuminants as style. Note that this method does not transfer the particular style of an image to the another one, but aims to normalize the additional injected style information, in which the illumination is considered as the style factor.

3. Methodology

Our proposed AWB strategy models the lighting in the scenes with multiple illuminants as style injected to the scene by different light sources. The weighting maps of different WB settings are extracted by using the affine parameters learned by a style extractor module, which is adapted from [53]. Proposed network adaptively normalizes different filter responses in any layer of the encoder by a particular style latent code. To finalize the capture, we follow the methods used in [4] for the inference-time post-processing.

3.1. Modified Camera Image Signal Processor

Following the prior work on modified camera ISP [2, 4], we employ a method for producing the high-resolution image with a fixed WB settings (*i.e.* daylight) and additional small images rendered with a set of pre-defined WB settings. The formula for rendering the small images can be described as follows

$$\hat{I}_{c_i} = M_{c_i}\phi(I_{init}) \tag{1}$$

where I_{init} is the initial high-resolution image rendered with a fixed WB setting (*i.e.* daylight), $\hat{I}_{c_i}$ represents the output image mapped to the target WB setting, M_{c_i} is the matrix that maps the colors of the initial image represented in a higher-dimensional space, and $\phi(\cdot)$ is a polynomial kernel function projecting the colors of the initial image into the higher-dimensional space. $\phi(\cdot)$ is optimized by minimizing the sum-squared error between the colors of target and source images, as in [4]. As distinct from [4], we consider this part as a pre-processing for training, and save the target images before training, instead of computing them on-the-fly.

After extracting the small images, following the method in [4], we employ a learning mechanism for the weighting maps of different scenes with a pre-defined set of WB settings. The details of the learning mechanism are explained in Section 3.2. We use this learned weighting maps for generating the final sRGB output image by linearly combining them with the small images, as shown in the following equation:

$$\tilde{I}_{corr} = \sum_i W_i \odot \tilde{I}_{c_i} \tag{2}$$

where $\tilde{I}_{corr}$ is the corrected small sRGB image, $\odot$ is Hadamard product, W_i represents the weighting map for i^{th} WB setting (*i.e.* c_i), and $\tilde{I}_{c_i}$ denotes the small image rendered with c_i.

3.2. Learning Weighting Maps by Style

Given a set of small images $\tilde{I}_{c_i}$, the learning mechanism learns to estimate $\{W_i\}$. In this work, we adapt a style removal network proposed in [53] as the learning mechanism of the weighting maps. This network consists of an encoder-

Figure 2: Example of predictions for the weighting maps and AWB results by blending these maps. We render the linear raw DNG files for the images in MIT-Adobe FiveK dataset [18] (id: 2808) in different WB settings. The rendered images with given WB settings and the AWB results of traditional camera pipeline are presented in the first row. The predicted weighting maps and the AWB correction results of our method are demonstrated in the second row.

decoder structure that employs adaptive feature normalization strategy to all layers of the encoder part. With the help of this strategy, any internal factor in the images can be modeled as an external style, which needs to be discarded or adjusted into another style. The main component to achieve this is Adaptive Instance Normalization (AdaIN) [46] for each encoder layer, which transfers the feature statistics computed across spatial locations. AdaIN simply aligns the channel-wise mean μ and variance σ of the feature maps of the content image x to the statistics of the style input y, as formulated in Equation 3.

$$\text{AdaIN}(x, y) = \sigma(y) \left(\frac{x - \mu(x)}{\sigma(x)} \right) + \mu(y) \tag{3}$$

To extract the style input for the images, we use a multi-head mapping module that maps the feature representations encoded by a pre-trained VGG network to the style latent space. The style latent code $\mathbf{w}$ is fed into different heads for different encoder levels, and each head h_i is attached to a projection layer p_i (*i.e.* fully-connected), which adapts the affine parameters y_i of each normalization layer in the encoder.

$$\mathbf{w} = M(\mathbf{z}), y_i = p_i(h_i(\mathbf{w})) \tag{4}$$

where $\mathbf{z}$ is the feature representation of the input image x extracted by VGG, and M denotes the style extractor module mapping the input latent space to the style latent space.

In our design, the style extractor module is composed of 5-layer MLP. The encoder contains 5 residual blocks, each of which has specific AdaIN layer to normalize the feature maps with the affine parameters projected by the corresponding head. The network takes the concatenated feature representations of the small images rendered with different WB settings as input, and learns to produce the weighting maps for these WB settings. As suggested in [53], we use skip-connections between encoder layers to preserve the related information while distilling the style. Overall design of proposed learning mechanism is shown in Figure 1.

We optimize this network by minimizing the reconstruction error between the ground truth and corrected patches, as shown in Equation 5.

$$\mathcal{L}_r = ||P_{gt} - \sum_i \hat{W}_i \odot P_{c_i}||_F^2 \tag{5}$$

where P_{gt} and P_{c_i} denote the ground truth patch and input patch rendered with WB setting of c_i, $\hat{W}_i$ is the weighting map of c_i, as the output of the network. We include the smoothing loss [4] to our final objective function.

$$\mathcal{L}_s = \sum_i ||\hat{W}_i * \nabla_x||_F^2 + ||\hat{W}_i * \nabla_y||_F^2 \tag{6}$$

where ∇_x and ∇_y are the horizontal and vertical Sobel filters with 3×3 kernel size. We did not include the perceptual loss since it dramatically increases the computational complexity of training. Our final objective function can be represented as follows:

$$\mathcal{L} = \mathcal{L}_r + \lambda \mathcal{L}_s \tag{7}$$

where λ denotes the regularization coefficient, which is set to 100 in our experiments.

3.3. Post-processing

We have two post-processing operations that can be applied to the learned weighting maps to further improve the quality of the final sRGB image. Following the prior work, we first apply the multi-scale ensembling for the weighting maps. This strategy is mainly based on generating a set

(A) Camera AWB **(B)** Deep WB **(C)** Mixed WB **(D)** Ours

Figure 3: Comparison of the qualitative results of our AWB method and the other methods on the selected samples from MIT-Adobe FiveK dataset [18]. We compare our results with the traditional camera AWB, Deep WB [2] and Mixed WB [4]. Image indices from top to bottom: 2882, 606, 659, 2431, 2550.

of multi-scale weighting maps, bilinear upsampling to the high-resolution, then averaging them for each WB setting. Secondly, we apply edge-aware smoothing (EAS) –with the help of the fast bilateral solver [11]– to the weighting maps with the guidance of high-resolution input image. In our experiments, we pick to apply both operations as the performance noticeably increases, as shown in the prior work.

4. Experiments

4.1. Experimental Details

In our experiments, we have employed RenderedWB dataset [5] as the training set. The dataset contains 65,000 sRGB images captured by different cameras, each of which has a specific pre-defined WB settings. Following the setup in the prior work, we have two sets of pre-defined WB settings, which are $\{\texttt{t},\texttt{f},\texttt{d},\texttt{c},\texttt{s}\}$ and $\{\texttt{t},\texttt{d},\texttt{s}\}$. $\{\texttt{t},\texttt{f},\texttt{d},\texttt{c},\texttt{s}\}$ refers to tungsten (2850K), fluorescent (3800K), daylight (5500K), cloudy (6500K), and shade (7500K), respectively. Each image has a corresponding accurately white-balanced sRGB image as ground-truth. We did not apply any data augmentation technique to the images. For all settings, we have trained each building block of our proposed model from scratch by using Adam optimizer [52] ($\beta_1 = 0.9$, $\beta_2 = 0.999$). The learning rate is set to $1e-4$ and we did not employ any scheduling strategy.

Table 1: Benchmark on single-illuminant Cube+ dataset [9]. Following the prior work [5], we reported the mean, first (**Q1**), second (**Q2**) and third (**Q3**) quantile of mean-squared error (**MSE**), mean angular error (**MAE**) and color difference (Δ**E 2000**) metrics. Different WB settings are denoted as {t, f, d, c, s}, which refers to tungsten, fluorescent, daylight, cloudy, and shade, respectively. p refers to the patch size. The top results are indicated with colored cells as, the best: **green**, the second: yellow, the third: red.

Method	MSE				MAE				ΔE 2000				Size
	Mean	Q1	Q2	Q3	Mean	Q1	Q2	Q3	Mean	Q1	Q2	Q3	
FC4 [45]	371.90	79.15	213.41	467.33	6.49°	3.34°	5.59°	8.59°	10.38	6.60	9.76	13.26	5.89 MB
Quasi-U CC [13]	292.18	15.57	55.41	261.58	6.12°	1.95°	3.88°	8.83°	7.25	2.89	5.21	10.37	622 MB
KNN WB [5]	194.98	27.43	57.08	118.21	4.12°	1.96°	3.17°	5.04°	5.68	3.22	4.61	6.70	21.8 MB
Interactive WB [3]	159.88	21.94	54.76	125.02	4.64°	2.12°	3.64°	5.98°	6.20	3.28	5.17	7.45	**38 KB**
Deep WB [2]	**80.46**	15.43	33.88	74.42	3.45°	1.87°	2.82°	4.26°	4.59	2.68	3.81	5.53	16.7 MB
Mixed WB [4] results													
$p = 64$, WB={t, d, s}	168.38	8.97	19.87	105.22	4.20°	1.39°	2.18°	5.54°	5.03	2.07	3.12	7.19	5.09 MB
$p = 64$, WB={t, f, d, c, s}	161.80	9.01	19.33	90.81	4.05°	1.40°	2.12°	4.88°	4.89	2.16	3.10	6.78	5.10 MB
$p = 128$, WB={t, f, d, c, s}	176.38	16.96	35.91	115.50	4.71°	2.10°	3.09°	5.92°	5.77	3.01	4.27	7.71	5.10 MB
Our results													
$p = 64$, WB={t, d, s}	92.65	**6.52**	**14.23**	**35.01**	**2.47°**	**0.82°**	**1.44°**	**2.49°**	**2.99**	**1.36**	**2.04**	**3.32**	61.0 MB
$p = 64$, WB={t, f, d, c, s}	151.38	29.49	56.35	125.33	4.18°	2.13°	3.03°	4.81°	5.42	3.11	4.42	6.76	61.1 MB
$p = 128$, WB={t, d, s}	88.03	7.92	17.73	45.01	2.61°	0.93°	1.58°	2.85°	3.24	1.50	2.30	3.95	61.2 MB
$p = 128$, WB={t, f, d, c, s}	100.24	10.77	37.74	70.18	3.09°	1.15°	2.61°	3.87°	3.96	1.59	3.55	5.51	61.3 MB

Table 2: Benchmark on mixed-illuminant evaluation set [4]. Highlights and symbols are the same as in Table 1.

Method	MSE				MAE				Δ E 2000			
	Mean	Q1	Q2	Q3	Mean	Q1	Q2	Q3	Mean	Q1	Q2	Q3
Gray Pixel [60]	4959.20	3252.14	4209.12	5858.69	19.67°	11.92°	17.21°	27.05°	25.13	19.07	22.62	27.46
Grayness index [59]	1345.47	727.90	1055.83	1494.81	6.39°	4.72°	5.65°	7.06°	12.84	9.57	12.49	14.60
KNN WB [5]	1226.57	680.65	1062.64	1573.89	5.81°	4.29°	5.76°	6.85°	12.00	9.37	11.56	13.61
Interactive WB [3]	1059.88	616.24	896.90	1265.62	5.86°	4.56°	5.62°	6.62°	11.41	8.92	10.99	12.84
Deep WB [2]	1130.60	621.00	886.32	1274.72	**4.53°**	**3.55°**	4.19°	**5.21°**	10.93	**8.59**	**9.82**	11.96
Mixed WB [4] results												
$p = 64$, WB={t, d, s}	**819.47**	655.88	845.79	1000.82	5.43°	4.27°	4.89°	6.23°	**10.61**	9.42	10.72	**11.81**
$p = 64$, WB={t, f, d, c, s}	938.02	757.49	961.55	1161.52	4.67°	3.71°	**4.14°**	5.35°	12.26	10.80	11.58	12.76
$p = 128$, WB={t, d, s}	830.20	584.77	853.01	**992.56**	5.03°	3.93°	4.78°	5.90°	11.41	9.76	11.39	12.53
$p = 128$, WB={t, f, d, c, s}	1089.69	846.21	1125.59	1279.39	5.64°	4.15°	5.09°	6.50°	13.75	11.45	12.58	15.59
Our results												
$p = 64$, WB={t, d, s}	868.01	649.36	889.00	1026.98	5.73°	4.48°	5.42°	6.34°	12.11	10.42	12.12	13.36
$p = 64$, WB={t, f, d, c, s}	1051.07	760.86	1024.00	1332.50	6.30°	4.43°	6.01°	7.69°	14.43	11.90	13.11	16.15
$p = 128$, WB={t, d, s}	822.77	**576.52**	**840.67**	1025.26	5.11°	3.93°	4.85°	5.51°	11.65	10.63	11.86	13.02
$p = 128$, WB={t, f, d, c, s}	834.28	629.95	842.71	1005.59	5.71°	4.57°	5.54°	6.19°	11.79	9.84	12.19	13.00

We have applied two post-processing operations, which are ensembling of multi-scale weighting maps and edge-aware smoothing. We have used the cropped images with the size of 64 and 128 for training, and the batch size is set to 32. We have conducted our experiments on a single NVIDIA RTX 2080Ti for 200 epochs. Our implementation is built on top of prior works [4, 53], and done in PyTorch [58].

Inference: During inference, we produce low-resolution version (*i.e.* 384×384) of the input images with the predefined WB settings, and concatenate them in order to feed into the proposed model. The model produces the weighting maps as output, to blend the final sRGB output. Before post-processing, we resize the weighting maps to the input resolution.

4.2. Evaluation Sets

To evaluate our method, we have used four different evaluation sets for both scenarios, which are namely Cube+ [8] and MIT-Adobe FiveK [19], and mixed-illuminant evaluation set proposed by [4] and night photography rendering set [26]. The Cube+ contains 1,707 single illumination color-calibrated images taken with Canon EOS 550D camera during various seasons. The MIT-Adobe FiveK dataset contains 5,000 images captured by different DSLR cameras where each image is manually retouched by multiple experts to correct the white-balance of the images. Moreover, we have used mixed-illuminant cases for our evaluation. The mixed-illuminant test set has 150 synthetic images composed of multiple illuminantions, which is rendered from 3D scenes modeled in Autodesk 3Ds Max [64].

(a) $\mathbf{W}_D$ **(b)** $\mathbf{W}_T$ **(c)** $\mathbf{W}_S$ **(d)** AWB

Figure 4: Comparison of the performance of the prior work [4] and our method on mixed-illuminant dataset. **Rows:** (odd) Mixed WB results, (even) Our results. Weighting maps for $\mathbf{W}_D$: Daylight, $\mathbf{W}_T$: Tungsten, $\mathbf{W}_S$: Shade.

4.3. Results

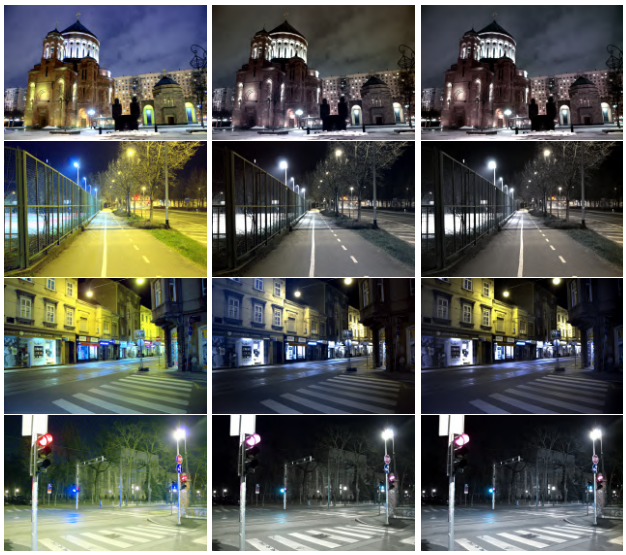
In this work, we propose to model the lighting in single- and mixed-illuminant scenes as style to improve the AWB strategy proposed in [4]. To demonstrate the qualitative results of our strategy, we use a set of images containing multiple illuminant scene from MIT-Adobe FiveK dataset [18]. At this point, we first render the linear raw DNG image files with different WB settings (*e.g.* daylight, tungsten, shade) by using the MATLAB code shared by [6]. Then, we feed these rendered images to proposed network for extracting their weighting maps to blend the final AWB corrected image. Note that using different re-touched versions of these images may produce different results. Moreover, for single-illuminant scenes, we compare the performance of our proposed strategy with the recent studies [45, 13, 5, 3, 2, 4] on Cube+ dataset [9]. Next, we include our results to the benchmark on mixed-illuminant evaluation set [4]. Following the prior work, we reported the mean, first (**Q1**), second (**Q2**) and third (**Q3**) quantile of mean-squared error, mean angular error and the color difference error on both datasets.

Figure 2 demonstrates the examples of predicted weighting maps of different WB settings and the AWB results blended by these maps. Samples are selected from MIT-Adobe FiveK dataset [18] where their indices are 323 and 2808 in a top-down order. The results indicate that style factor can represent the illuminant in a more detail-oriented way, and thus producing more interpretable weighting maps. Instead of roughly representing the region of the light falling to the objects in the captured scene, our method can differentiate the different illuminants on the same object accurately. This leads to improve the performance of the AWB strategy proposed in the prior work [4]. Note that our method does not require any illuminant estimation step. Moreover, in Figure 3, we introduce the comparison of the qualitative results of our AWB method and the recent methods [2, 4] on the selected samples from the same dataset. It shows that our proposed method achieves competitive per-pixel performance on AWB correction in sRGB space when compared to the recent methods.

Models	**MSE**	**MAE**	**ΔE 2000**
Single-illuminant dataset, WB = $\{\texttt{t},\texttt{d},\texttt{s}\}$, $p = 64$			
$ms = 0, eas = 0$	98.55	2.71°	3.32
$ms = 1, eas = 0$	93.78	2.59°	3.15
$ms = 0, eas = 1$	97.20	2.66°	3.28
$ms = 1, eas = 1$	**92.65**	**2.47°**	**2.99**
Mixed-illuminant dataset, WB = $\{\texttt{t},\texttt{d},\texttt{s}\}$, $p = 128$			
$ms = 0, eas = 0$	878.58	5.05°	12.12
$ms = 1, eas = 0$	843.50	**5.04°**	11.70
$ms = 0, eas = 1$	843.64	**5.04°**	11.98
$ms = 1, eas = 1$	**822.77**	5.11°	**11.65**

Table 3: The ablation study on using multi-scale weighting maps and applying edge-aware smoothing to weighting maps. p: patch size, ms: multi-scale weighting maps, eas: edge-aware smoothing.

Table 1 presents the quantitative results of our method and the recent methods, which are evaluated on Cube+ dataset. We have conducted our experiments with different patch sizes (*i.e.* 64 and 128) and different sets of WB settings (*i.e.* $\{\texttt{t},\texttt{d},\texttt{s}\}$ and $\{\texttt{t},\texttt{f},\texttt{d},\texttt{c},\texttt{s}\}$). All of our results outperform the results of the other compared methods over the most parts of all metrics. Particularly, the model trained with the patch size of 64 and on WB settings of $\{\texttt{t},\texttt{d},\texttt{s}\}$ achieves the best performance among the other models with different settings. As also stated in [4], we state that smaller patch sizes lead to better model the illuminant. However, in contrast to [4], increasing the number of WB setting in the set of training WB settings does not provide any advantage on modeling the lighting in single-illuminant scenarios. We think that less number of WB settings to blend for the final output makes easier to build the knowledge on the corre-



(a) Standard ISP **(b)** ISP with [4] **(c)** ISP with ours

Figure 5: Comparison on the night photography rendering results of the standard camera pipeline, the prior work [4] and our method.

lation between pixels by weighting maps. We also believe that using the input with more channels for training may increase the required complexity of the architecture to model the illuminant, and it may not be fair to compare them with the exact same architecture. Lastly, due to the style extraction part of the proposed network, our method has larger memory overhead when compared to the recent methods.

We have conducted the same experiments on synthetic mixed-illuminant evaluation set [4], and the overall results are reported in Table 2. According to these results, it is hard to pick a superior method over all other methods, as depending on the metric. Our method performs better when evaluated on mean-squared error, while achieving competitive results compared to the recent methods for mean angular error and color difference metrics. Moreover, Figure 4 demonstrates the visual comparison of the weighting maps and the blended AWB results of the prior work and our method on this synthetic dataset. As several illumination sources can affect the different parts of a single object at the same time, modeling the lighting as style helps to produce more detailed weighting maps, especially for the parts containing objects. The weighting maps produced by our proposed network give more detail-oriented results when compared to the prior work, even if it falls behind the prior work on some quantitative metrics. Synthetic data could have more sharp edges than the real-world images. When we use this kind of blending strategy with detail-oriented weighting maps and not including such samples to training, it may have caused the color discrepancy on the edges of the final output. This may reduce the quantitative performance of our method on this dataset.

As ablating the effects of two post-processing methods proposed in [4] on the performance of our strategy, we have conducted additional experiments on our best-performed settings for both datasets where the post-processing methods are alternately excluded during inference time. Table 3 presents the results of different combinations of the post-processing methods applied. It verifies that these methods help to improve the quality of the weighting maps, so do the qualitative results. At this point, one may argue that adding total variation regularization term to the final objective function can also achieve a similar improvement, and resolve the need of post-processing. This addition would require re-considering the smoothing loss and the pipeline proposed by [4], which is beyond the scope of this study.

Moreover, night photography rendering [26] is another challenging task containing different scenarios affected by multiple illuminants. Correcting AWB for the images captured at night may not be easily handled by assuming the illuminant in the scene is global. To show the validity of our strategy, we integrate our AWB method into the camera ISP for processing night images. Figure 5 presents the rendered night images by the standard camera pipeline, and its variants that include Mixed WB [4] and our AWB method. Results show that the pipeline including our AWB method produces more natural and realistic images over a wide range of night images. At this point, we assess the results according to how similar the produced colors of the objects in the scene to the human visual perception, not visual plausibility. Note that we include the same operations (*i.e.* denoising [21, 56], gamma correction, tone mapping and auto-contrast) to the pipeline in the same order, except white-balancing strategies.

5. Conclusion

In this work, we have proposed a novel idea of modeling the lighting as style factor for improving the recent AWB correction methods for single- and mixed-illuminant scenes. Our proposed network extracts the additional style information injected by the lighting sources to the scene, and learns to weight the maps of different WB settings to blend them for AWB correction. We have conducted several experiments on the datasets containing mostly single-illuminant scenes, the synthetic mixed-illuminant evaluation set and night photography rendering set. The results indicate the illuminant can be modeled by the style factor, and our method produces promising correction results in both real-world scenarios and the synthetic scenes without requiring illuminant estimation. The next steps for this task could be to design a style extraction module with lower memory overhead without sacrificing the performance.

References

[1] Mahmoud Afifi, Jonathan T Barron, Chloe LeGendre, Yun-Ta Tsai, and Francois Bleibel. Cross-camera convolutional color constancy. In *The IEEE International Conference on Computer Vision (ICCV)*, 2021.

[2] Mahmoud Afifi and Michael S. Brown. Deep white-balance editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[3] Mahmoud Afifi and Michael S Brown. Interactive white balancing for camera-rendered images. In *Color and Imaging Conference*, volume 2020, pages 136–141. Society for Imaging Science and Technology, 2020.

[4] Mahmoud Afifi, Marcus A. Brubaker, and Michael S. Brown. Auto white-balance correction for mixed-illuminant scenes. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 1210–1219, January 2022.

[5] Mahmoud Afifi, Brian Price, Scott Cohen, and Michael S Brown. When color constancy goes wrong: Correcting improperly white-balanced images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1535–1544, 2019.

[6] Mahmoud Afifi, Abhijith Punnappurath, Abdelrahman Abdelhamed, Hakki Can, Abdullah Abuolaim, and Michael Brown. Color temperature tuning: Allowing accurate post-capture white-balance editing. *Color and Imaging Conference*, 2019:1–6, 10 2019.

[7] Mahmoud Afifi, Abhijith Punnappurath, Abdelrahman Abdelhamed, Hakki Can Karaimer, Abdullah Abuolaim, and Michael S Brown. Color temperature tuning: Allowing accurate post-capture white-balance editing. In *Color and Imaging Conference*, volume 2019, pages 1–6. Society for Imaging Science and Technology, 2019.

[8] Nikola Banić, Karlo Koščević, and Sven Lončarić. Unsupervised learning for color constancy. *arXiv preprint arXiv:1712.00436*, 2017.

[9] Nikola Banić, Karlo Koščević, and Sven Lončarić. Unsupervised learning for color constancy, 2019.

[10] Jonathan T. Barron. Convolutional color constancy. *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 379–387, 2015.

[11] Jonathan T. Barron and Ben Poole. The fast bilateral solver. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision – ECCV 2016*, pages 617–632, Cham, 2016. Springer International Publishing.

[12] Jonathan T. Barron and Yun-Ta Tsai. Fast fourier color constancy. In *CVPR*, 2017.

[13] Simone Bianco and Claudio Cusano. Quasi-unsupervised color constancy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[14] David H. Brainard and William T. Freeman. Bayesian method for recovering surface and illuminant properties from photosensor responses. In Bernice E. Rogowitz and Jan P. Allebach, editors, *Human Vision, Visual Processing, and Digital Display V*, volume 2179, pages 364 – 376. International Society for Optics and Photonics, SPIE, 1994.

[15] David H Brainard and William T Freeman. Bayesian color constancy. *JOSA A*, 14(7):1393–1411, 1997.

[16] David H. Brainard and Brian A. Wandell. Analysis of the retinex theory of color vision. *Journal of the Optical Society of America. A, Optics and image science*, 3 10:1651–61, 1986.

[17] Gershon Buchsbaum. A spatial processor model for object colour perception. *Journal of The Franklin Institute-engineering and Applied Mathematics*, 310:1–26, 1980.

[18] Vladimir Bychkovsky, Sylvain Paris, Eric Chan, and Fredo Durand. Learning photographic global tonal adjustment with a database of input/output image pairs. pages 97 – 104, 07 2011.

[19] Vladimir Bychkovsky, Sylvain Paris, Eric Chan, and Frédo Durand. Learning photographic global tonal adjustment with a database of input/output image pairs. In *CVPR 2011*, pages 97–104. IEEE, 2011.

[20] Vlad C. Cardei, Brian Funt, and Kobus Barnard. Estimating the scene illumination chromaticity by using a neural network. *J. Opt. Soc. Am. A*, 19(12):2374–2386, Dec 2002.

[21] Meng Chang, Qi Li, Huajun Feng, and Zhihai Xu. Spatial-adaptive network for single image denoising. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, pages 171–187, Cham, 2020. Springer International Publishing.

[22] Dongliang Cheng, Dilip K. Prasad, and Michael S. Brown. Illuminant estimation for color constancy: why spatial-domain methods work and the role of the color distribution. *J. Opt. Soc. Am. A*, 31(5):1049–1058, May 2014.

[23] Mircea Cimpoi, Subhransu Maji, and Andrea Vedaldi. Deep filter banks for texture recognition and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[24] Peter Dayan, Geoffrey E. Hinton, Radford M. Neal, and Richard S. Zemel. The helmholtz machine. *Neural Comput.*, 7(5):889–904, sep 1995.

[25] Marc Ebner. Color constancy. *Color Constancy*, pages 1–390, 05 2007.

[26] Egor Ershov and the others. Ntire 2022 challenge on night photography rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 1287–1300, June 2022.

[27] Finlayson, Hordley, and Tastl. Gamut constrained illuminant estimation. In *Proceedings Ninth IEEE International Conference on Computer Vision*, pages 792–799 vol.2, 2003.

[28] G. Finlayson and S. Hordley. Improving gamut mapping color constancy. *IEEE Transactions on Image Processing*, 9(10):1774–1783, 2000.

[29] Graham D. Finlayson and Elisabetta Trezzi. Shades of gray and colour constancy. In *Color Imaging Conference*, 2004.

[30] David Alexander Forsyth. A novel algorithm for color constancy. *International Journal of Computer Vision*, 5:5–35, 2004.

[31] Brian Funt, Vlad Cardei, and Kobus Barnard. Learning color constancy. pages 58–60, Dec. 1997.

[32] Leon Gatys, Alexander Ecker, and Matthias Bethge. A neural algorithm of artistic style. *arXiv*, 08 2015.

[33] Leon Gatys, Alexander S Ecker, and Matthias Bethge. Texture synthesis using convolutional neural networks. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015.

[34] Peter Vincent Gehler, Carsten Rother, Andrew Blake, Tom Minka, and Toby Sharp. Bayesian color constancy revisited. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2008.

[35] Zoubin Ghahramani. Factorial learning and the em algorithm. In G. Tesauro, D. Touretzky, and T. Leen, editors, *Advances in Neural Information Processing Systems*, volume 7. MIT Press, 1994.

[36] Golnaz Ghiasi, Honglak Lee, Manjunath Kudlur, Vincent Dumoulin, and Jonathon Shlens. Exploring the structure of a real-time, arbitrary neural artistic stylization network. *ArXiv*, abs/1705.06830, 2017.

[37] Arjan Gijsenij, Theo Gevers, and Joost van de Weijer. Generalized gamut mapping using image derivative structures for color constancy. *International Journal of Computer Vision*, 86(2-3):127–139, 2010.

[38] Arjan Gijsenij, Theo Gevers, and Joost van de Weijer. Computational color constancy: Survey and experiments. *IEEE Transactions on Image Processing*, 20(9):2475–2489, 2011.

[39] Arjan Gijsenij, Theo Gevers, and Joost van de Weijer. Improving color constancy by photometric edge weighting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(5):918–929, 2012.

[40] Alan Gilchrist. *Seeing Black and White*, volume 17, page 430. 06 2006.

[41] David J. Heeger and James R. Bergen. Pyramid-based texture analysis/synthesis. In *Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '95, page 229–238, New York, NY, USA, 1995. Association for Computing Machinery.

[42] Daniel Hernandez-Juarez, Sarah Parisot, Benjamin Busam, Ales Leonardis, Gregory Slabaugh, and Steven McDonagh. A multi-hypothesis approach to color constancy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[43] Geoffrey E. Hinton and Zoubin Ghahramani. Generative models for discovering sparse distributed representations. *Philosophical Transactions: Biological Sciences*, 352(1358):1177–1190, 1997.

[44] Geoffrey E. Hinton and Richard S. Zemel. Autoencoders, minimum description length and helmholtz free energy. NIPS'93, page 3–10. Morgan Kaufmann Publishers Inc., 1993.

[45] Yuanming Hu, Baoyuan Wang, and Stephen Lin. Fc4: Fully convolutional color constancy with confidence-weighted pooling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[46] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. pages 1510–1519, 10 2017.

[47] H.R. Joze, Mark Drew, Graham Finlayson, and Perla Troncoso Rey. The role of bright pixels in illumination estimation. *Final Program and Proceedings - IS and T/SID Color Imaging Conference*, 2012:41–46, 01 2012.

[48] Sergey Karayev, Matthew Trentacoste, Helen Han, Aseem Agarwala, Trevor Darrell, Aaron Hertzmann, and Holger Winnemoeller. Recognizing image style. In *Proceedings of the British Machine Vision Conference*. BMVA Press, 2014.

[49] Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data. In *Proc. NeurIPS*, 2020.

[50] Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Alias-free generative adversarial networks. In *Proc. NeurIPS*, 2021.

[51] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[52] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017.

[53] Furkan Kinli, Baris Ozcan, and Furkan Kirac. Instagram filter removal on fashionable images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 736–745, June 2021.

[54] Yi-Chen Lo, Chia-Che Chang, Hsuan-Chao Chiu, Yu-Hao Huang, Chia-Ping Chen, Yu-Lin Chang, and Kevin Jou. Clcc: Contrastive learning for color constancy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8053–8063, June 2021.

[55] Zhongyu Lou, Theo Gevers, Ninghang Hu, and Marcel P. Lucassen. Color constancy by deep learning. In Xianghua Xie, Mark W. Jones, and Gary K. L. Tam, editors, *Proceedings of the British Machine Vision Conference (BMVC)*, pages 76.1–76.12. BMVA Press, September 2015.

[56] Sami Menteş, Furkan Kınlı, Barış Özcan, and Furkan Kıraç. [re] spatial-adaptive network for single image denoising. In *ML Reproducibility Challenge 2020*, 2021.

[57] Taesung Park, Jun-Yan Zhu, Oliver Wang, Jingwan Lu, Eli Shechtman, Alexei A. Efros, and Richard Zhang. Swapping autoencoder for deep image manipulation. In *Advances in Neural Information Processing Systems*, 2020.

[58] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. 2019.

[59] Yanlin Qian, Joni-Kristian Kämäräinen, Jarno Nikkanen, and Jiri Matas. On finding gray pixels. In *IEEE International Conference of Computer Vision and Pattern Recognition*, 2019.

[60] Yanlin Qian, Said Pertuz, Jarno Nikkanen, J. Kämäräinen, and Jiri Matas. Revisiting gray pixel for statistical illumination estimation. In *VISIGRAPP*, 2019.

[61] Wu Shi, Chen Change Loy, and Xiaoou Tang. Deep specialized network for illuminant estimation. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision – ECCV 2016*, pages 371–387. Springer International Publishing, 2016.

[62] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015.

[63] Joshua B. Tenenbaum and William T. Freeman. Separating style and content with bilinear models. *Neural Computation*, 12(6):1247–1283, 2000.

[64] Sham Tickoo. Autodesk 3ds max 2021: A comprehensive guide. *Cadcim Technologies*, 2020.

[65] Joost van de Weijer, Theo Gevers, and Arjan Gijsenij. Edge-based color constancy. *IEEE Transactions on Image Processing*, 16(9):2207–2214, 2007.

[66] Zhe Wu, Zuxuan Wu, Bharat Singh, and Larry Davis. Recognizing instagram filtered images with feature de-stylization. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(07):12418–12425, Apr. 2020.

[67] Bolei Xu, Jingxin Liu, Xianxu Hou, Bozhi Liu, and Guoping Qiu. End-to-end illuminant estimation based on deep metric learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.