

Revisiting Image Fusion for Multi-Illuminant White-Balance Correction

David Serrano-Lozano^{1,2} Aditya Arora^{3,4,5} Luis Herranz⁶ Konstantinos G. Derpanis^{3,4}
Michael S. Brown³ Javier Vazquez-Corral^{1,2}

¹Computer Vision Center ²Universitat Autònoma de Barcelona ³York University

⁴Vector Institute ⁵TU Darmstadt ⁶Universidad Autónoma de Madrid

{dserrano, javier.vazquez}@cvc.uab.cat luis.herranz@uam.es

{adityac8, kosta, mbrown}@yorku.ca

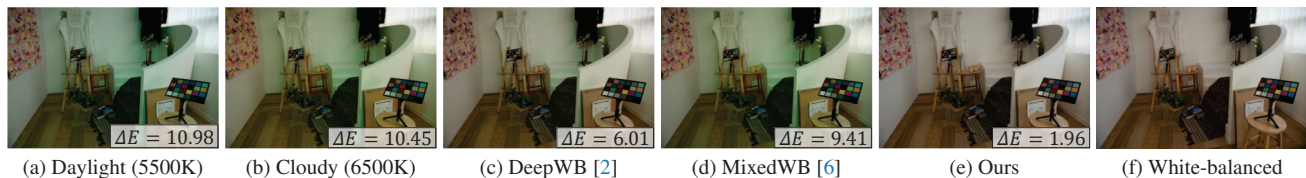


Figure 1. Camera presets (a-b) fail to correct color casts. DeepWB [2] (c) struggles under multiple illuminants. MixedWB [6] (d) linearly blends different WB settings but also fails on complex cases (e.g., the white wall). Our transformer-based method (e) effectively blends five WB presets, achieving better correction. ΔE_{2000} values (lower is better) are shown in the bottom-right corner of each image.

Abstract

White balance (WB) correction under multiple illuminants is a persistent challenge in computer vision. Recent methods explored fusion-based approaches, where a neural network linearly blends multiple sRGB versions of an input image, each processed with predefined WB presets. However, we demonstrate that these methods are suboptimal. Additionally, existing fusion-based methods rely on sRGB WB datasets that lack proper multi-illuminant training data. To address these challenges, we introduce two key contributions. First, we propose an efficient transformer-based model that effectively captures spatial dependencies across sRGB WB presets, substantially improving upon linear fusion techniques. Second, we introduce a large-scale multi-illuminant dataset comprising over 16,000 sRGB images rendered with five different WB settings, along with WB-corrected images. Our method achieves up to 100% improvement over existing techniques on our new multi-illuminant image fusion dataset.

<https://revisitingmiwb.github.io>

1. Introduction

Auto-white-balance (AWB) correction is a crucial step applied onboard cameras to remove the color cast caused by the light conditions in the imaged scene. Proper white-balance correction is needed to produce visually pleasing images and to assist downstream computer vision tasks [4].

Conventional AWB techniques [8, 12, 21, 26, 48] assume that a *single* light source illuminates a scene. Typical AWB methods operate on the RAW sensor image to first estimate the color cast in the image in the form of a sensor-specific R, G, B vector. Subsequently, the illumination’s color cast is removed by scaling each color channel of the RAW image such that the estimated illumination becomes achromatic (i.e., $R=G=B$). The white-balanced RAW image is then rendered through the camera’s image signal processor (ISP) to render the final image in a standard RGB (sRGB) space.

While the single-illuminant assumption works reasonably well for many scenes, it fails on common scenes with multiple illuminants, such as indoor scenes with illumination from indoor and outdoor lighting (e.g., via a window). Prior work examined ways to correct such multiple illuminations [43, 49, 50]. These methods typically follow a two-step process that first estimates the constituent scene illuminants, followed by spatially varying WB correction in the RAW sensor image based on the estimates. The need for both estimation and correction makes this a challenging problem. Another bottleneck for multi-illuminant learning methods is the difficulty in preparing per-pixel annotated RAW image datasets for training and testing.

Afifi et al. [6] introduced MixedWB, a multi-illuminant WB correction approach that leverages pre-defined WB values for common scene illuminations (i.e., the camera settings for *manual WB mode* rather than AWB). The sensor’s R, G, B response to several common scene illuminations can

be accessed directly via camera API calls or image metadata. Given a RAW input image, MixedWB first renders it to sRGB using five WB settings — *tungsten*, *fluorescent*, *daylight*, *cloudy*, and *shade*. A neural network then predicts per-pixel blending weights to linearly merge these five images into a final sRGB output. This formulation transforms the inherently ill-posed problem of per-pixel illuminant estimation into a more straightforward task: learning a combination of a fixed set of images.

MixedWB, however, has two main limitations. First, while linear blending is effective in the RAW linear color space, MixedWB applies it in the non-linear sRGB space. In this paper, we show that this approach is suboptimal because ground-truth correct pixels in sRGB often fall outside the convex hull defined by the preset pixels. To address this limitation, we propose a transformer-based [46] fusion strategy that learns a more flexible and accurate WB blending function.

Second, MixedWB is trained on the RenderedWB dataset [4], which contains images rendered with distinct sRGB WB presets but lacks true multi-illuminant scenes, limiting its effectiveness for real-world WB correction. The only available true multi-illuminant sRGB dataset is generated from 3D-rendered scenes [6], introducing a domain shift from real-world images. To address this limitation, we introduce a new multi-illuminant dataset containing 16,284 sRGB images, each rendered with five WB presets and accompanied by a corresponding ground-truth WB-corrected image.

Figure 1 illustrates the benefits of our approach using a multi-illuminant scene rendered with the two most suitable WB presets (*Daylight* and *Cloudy*). We compare the results of an sRGB WB method (DeepWB [2]), MixedWB [6], our transformer-based approach, and the ground-truth white-balanced image. The results show that all WB methods struggle to remove the greenish color cast, particularly noticeable on the white wall, except for our transformer-based method, which achieves the most accurate correction.

Contribution. We revisit fusion-based multi-illuminant WB correction and introduce two key contributions. First, we propose a transformer-based method that performs WB in multi-illuminant scenes more accurately and efficiently by blending five distinct WB presets. Second, we present a new multi-illuminant white balance dataset in the sRGB domain, created by repurposing the RAW multi-illuminant dataset from Kim et al. [29]. This dataset comprises 2,714 scenes, each rendered into five WB sRGB images with a corresponding ground-truth white-balanced image. Our method significantly outperforms previous approaches across multiple datasets, including our new multi-illuminant dataset, Cube+ [7], and the Synthetic test set [6].

2. Related work

Our related work is organized into three areas: (i) conventional WB correction, (ii) WB methods for processed sRGB images, and (iii) multi-illuminant WB correction.

Conventional white-balance correction. Traditional white balance correction techniques aim to estimate the sensor’s response to illumination from a single RAW image. Early statistical methods [12–14, 16, 17, 22, 33, 40, 45] computed simple RAW image statistics as features for illumination estimation. These early approaches were later extended to employ more complex statistical inferences, such as color-by-correlation [19], Bayesian color constancy [11, 20], and voting (e.g., [41, 47, 48]).

Deep learning methods now comprise the state of the art for illumination estimation (e.g., [3, 10, 24, 37–39, 42, 51, 55]). Since illumination estimation is performed on the sensor-specific RAW image format, these methods require large, annotated datasets of images for each targeted sensor.

One limitation of conventional methods is their reliance on the assumption of a single dominant light source in the scene. When faced with scenes containing multiple light sources, such as indoor environments with artificial and natural lighting, traditional methods may fail to achieve accurate color reproduction.

White-balance methods for rendered sRGB images. The camera’s ISP hardware generally applies white balance as an early processing step, followed by multiple non-linear adjustments to produce the final sRGB output. This makes post-capture WB correction challenging. Recent works explore methods for performing WB correction directly on sRGB images [3–5]. A notable advancement in this area is the application of deep neural networks for WB correction and adjustment in camera-rendered sRGB images [2]. SWBNet [36] introduced a learning-based architecture to improve the consistency across different color temperatures leveraging a contrastive loss, while WBFlow [35] proposed a DNN-based method that leverages neural flow. These methods demonstrate improved color accuracy. Still, they often depend on post-capture correction techniques or user interaction, adding complexity and additional steps to the image processing workflow.

Multiple-illuminant methods. Various techniques target multiple-illuminant white-balance correction. Previous methods often rely on pixel-wise illumination estimation strategies [9, 18, 23, 25, 27, 28], many of which assume that the number of light sources in the scene is known. This strong assumption limits their applicability in real-world scenarios. More recent approaches [30, 52] leverage deep learning to estimate pixel-wise illuminants in multi-illuminant scenes directly from the RAW images.

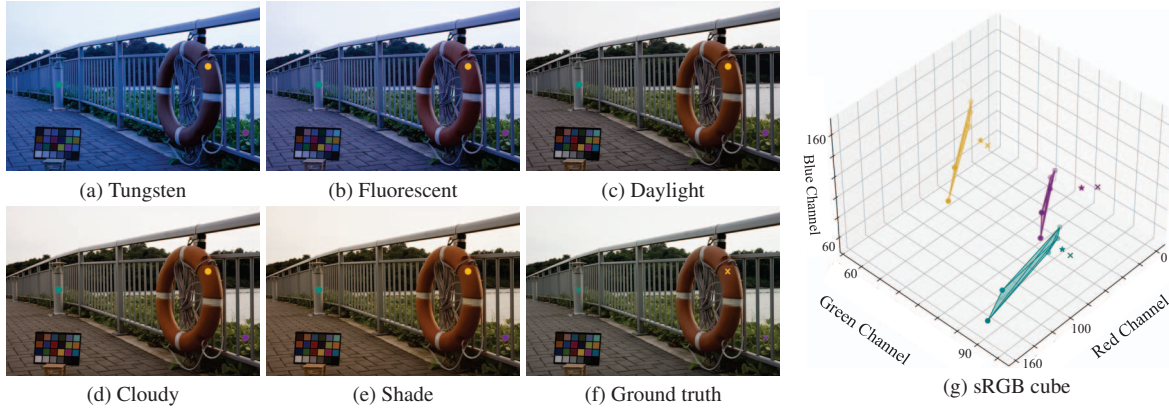


Figure 2. Example from the RenderedWB [4] dataset. (a)-(e) show the same scene processed with five distinct WB presets, while (f) presents the white-balanced image obtained using the color checker. Three sample points, marked with teal, yellow, and purple dots for the WB presets and crosses for the ground truth image, are selected across all images. (g) visualizes the pixel values in the sRGB space, along with the polytope formed by the WB presets. In here, we also show the results of our method as stars. Note that each axis has a different scale to ease the visualization.

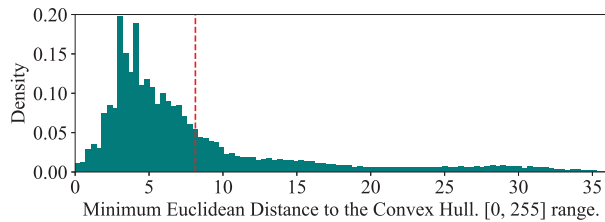


Figure 3. Histogram of the minimum distances to the convex hull for the ground truth pixels that lie outside the convex hull in the RenderedWB dataset [4]. The pixels that lie outside the convex hull are 99.76% of the pixels. The red dashed line represents the mean, 8.13 units.

Most closely related to our work are methods that merge multiple sRGB images. MixedWB [6] address the multi-illuminant WB problem as a constrained fusion problem in the sRGB domain. In particular, they pose the multi-illuminant white balance problem by obtaining a linear combination of a fixed input WB presets. In their approach, a deep neural network generates a per-pixel weighting map for the different settings. Similarly, StyleWB [32] use the features of a VGG-16 [44] architecture for improved performance. In this paper, we demonstrate that simply linearly combining distinct WB presets is insufficient for complex cases, as a result both MixedWB and StyleWB struggle to obtain an accurate white-balanced image.

The success of learning-based methods depends largely on the availability and quality of training and evaluation data. MixedWB and StyleWB focus on multi-illuminant sRGB WB correction but mainly rely on the RenderedWB [4] dataset for training and Cube+ [7] for testing, both of which contain only single-illuminant scenes. The Synthetic test set by Afifi et al. [6] is the only sRGB dataset for multi-illuminant scenes but lacks training samples and has a large domain gap from real images. Kim et al. [29]

created the most comprehensive multi-illumination dataset by capturing RAW images of the same scene under different light sources. We repurpose this dataset to support post-WB fusion in the sRGB domain, which is our target application.

This paper introduces an efficient transformer-based architecture that fuses five WB sRGB presets non-linearly, without requiring explicit per-pixel weight maps. Additionally, we introduce a new dataset for training and evaluating multi-illuminant white balance correction in camera-rendered sRGB images. Empirically, we demonstrate that our transformer-based fusion method outperforms previous state-of-the-art approaches in WB correction by effectively blending distinct WB settings.

3. Motivation

MixedWB [6] and StyleWB [32] hypothesize that multi-illuminant white balance can be achieved by linearly combining a set of sRGB images rendered with different preset illuminants. This assumption implies that the white-balanced pixel value should lie within the convex hull formed by the pixel values of the preset images in sRGB space. Formally, for each pixel i , the white balance pixel, T_i is computed by:

$$T_i = \sum_p w_i^p P_i^p, \quad (1)$$

where P represents the number of WB-presets values, i and p denote the pixel coordinates and WB preset, respectively, and w_i is a per-pixel blending factor defining a convex combination of the preset values:

$$\sum_p w_i^p = 1, \quad w_i^p \geq 0, \quad \forall i, p. \quad (2)$$

While the assumption of linear blending holds in the RAW linear domain, it does not necessarily extend to the

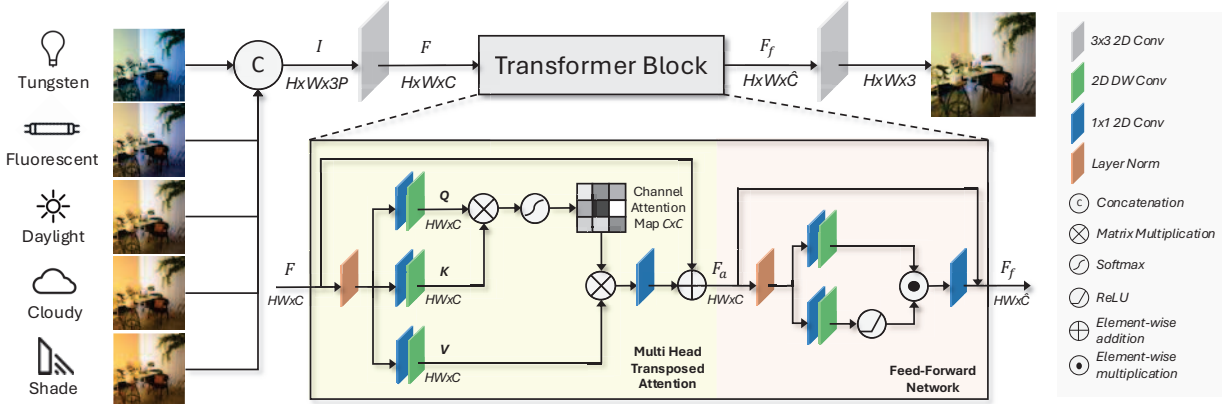


Figure 4. Overview of our multi-illuminant WB correction architecture. The input RAW image is rendered using the five predefined WB settings. The sRGB images are concatenated and processed through a 3×3 convolution. The projection is used to generate the Query, Value, and Key for the transposed attention mechanism. The final features are transformed using a Feed-Forward Network and decoded using a 3×3 convolution. The output is a white-balance corrected sRGB image, fused from the five input settings.

non-linear sRGB space where methods like MixedWB and StyleWB operate. Figure 2 illustrates this limitation by plotting three example pixels from the RenderedWB dataset [4], where WB preset values are shown as dots and ground-truth values as crosses. The ground-truth pixels lie outside the convex hull formed by the presets, indicating that linear blending cannot accurately recover the correct colors in sRGB space. More broadly, in the RenderedWB dataset, only 0.24% of pixels lie within the convex hull, and for those outside, the average minimum distance is 8.13 units. Figure 3 shows the histogram of these distances, highlighting their overall distribution.

To overcome this, we propose an efficient and more accurate fusion strategy using an end-to-end transformer block. Our method, represented by stars in Figure 2 (g), produces values outside the convex hull, significantly closer to the ground truth.

4. Transformer-based WB Fusion

Our method fuses different WB settings in a non-linear end-to-end manner. More specifically, we leverage a transformer block, as the attention mechanism inherently captures long-range spatial dependencies, making it well-suited for this task [34, 54]. While vision transformers [15] traditionally divide images into patches treated as tokens, this approach is prohibitive for high-resolution images. To improve efficiency, we adopt the channel-wise attention mechanism [53] in the feature space, enabling the model to effectively aggregate information across different WB settings while maintaining computational feasibility.

First, our approach renders the RAW sensor image using five predefined WB settings — namely *tungsten*, *fluorescent*, *daylight*, *cloudy*, and *shade*. These sensor responses, calibrated at the factory, can be accessed by setting the cam-

era in manual mode and examining its metadata. The images are then rendered to sRGB using their respective WB presets. Notably, the five input images we use are readily available in most modern ISPs.

Working in sRGB with a set of pre-defined settings offers significant advantages. It transforms an extremely ill-posed problem — obtaining the correct illuminant estimate for each pixel location — into a more manageable one — merging among a narrow finite set of images. Additionally, operating in sRGB rather than RAW facilitates generalization across different camera sensors whose distinct characteristics, such as sensitivities, affect RAW image formation.

Figure 4 shows a schematic diagram of our method. Initially, we concatenate the WB presets to form a composite image, $I \in \mathbb{R}^{H \times W \times 3P}$, where $H \times W$ represents the spatial dimension of the images and P denotes the number of WB presets. Given the combined WB presets, I , we first apply a 3×3 convolution to extract low-level features and convert the number of channels to C . We first use a Multi-Head Transposed Attention block [53] in which we generate the *query* (Q), *key* (K), and *value* (V) projections from the convolutional features. In contrast to spatial attention, the transposed attention map A is computed by swapping Q and K as:

$$A = \text{Softmax}(K^T Q). \quad (3)$$

After the Multi-Head Transposed Attention module, the output undergoes processing through a Feed-Forward Network [15], which transforms the features by operating on each location independently yet identically by formulating it as the element-wise product of two parallel paths. Finally, the output features are convolved with a 3×3 kernel to reduce the number of channels to three and produce the final sRGB white-balanced corrected image. In our experiments, we use five presets ($P=5$) and $C=15$.

To summarize, our multi-illuminant WB method leverages distinct WB presets and a transformer block by fusing these presets in a nonlinear manner, allowing the model to identify regions where each setting most effectively achieves accurate white balance. Unlike MixedWB [6] and StyleWB [32], our approach does not require the inefficient per-pixel weight estimation to form the white-balanced image, handling the process in an end-to-end manner. Notably, our method has only 7.9K trainable parameters, enabling fast inference with minimal memory requirements due to the transposed attention mechanism.

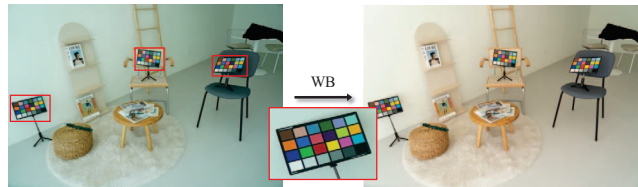
5. Multi-illuminant sRGB Dataset

MixedWB [6], StyleWB [32] and, our method rely on sRGB WB presets to estimate the white-balance corrected image. However, previous state-of-the-art approaches have primarily been both trained and tested on single-illuminant datasets, such as RenderedWB [4] and Cube+ [7], while evaluation on multi-illuminant scenes has been limited to a small synthetic dataset [6] containing only 30 scenes. To bridge this gap, we introduce a new dataset designed to support fusion-based multi-illuminant WB methods. Our dataset includes (i) five sRGB-rendered input images, each corresponding to a different WB preset, and (ii) the sRGB ground truth image with accurate WB.

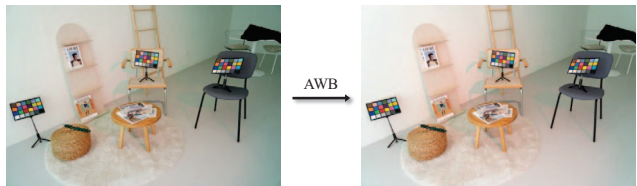
Recently, Kim et al. [29] introduced the large-scale multi-illuminant (LSMI) dataset. This dataset contains RAW images of scenes with multiple cameras. For each scene, the illumination was carefully controlled. In particular, the scene was captured first under a single illumination. Afterward, additional light sources were added to the scene. This step-by-step lighting allowed a per-pixel computation in the RAW image of the contribution of each light source. The public LSMI dataset was intended for RAW multi-illumination color constancy algorithms. This dataset serves as the basis for ours.

In particular, we repurpose the Sony and Nikon images of the LSMI dataset [29]. The total number of scenes is 955 for Nikon and 1,317 for Sony. Each scene contains either two or three illuminants. Given the DNG images of the LSMI dataset, we compute the five sRGB rendered images using Adobe Camera RAW [1] using the five standard camera WB presets: *daylight*, *cloudy*, *tungsten*, *shade*, *fluorescent*. We omit the Samsung Galaxy camera from the LSMI dataset due to incompatibility with Adobe Camera RAW. The images have resolutions of 7360×4912 and 6000×4000 for the Nikon and Sony sets, respectively.

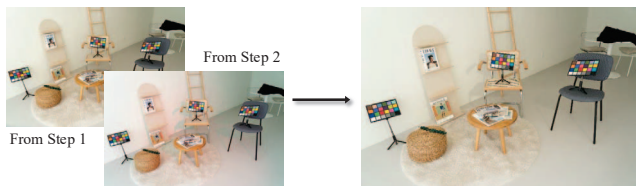
Figure 5 illustrates our procedure for generating ground truth white-balanced images in scenes with multiple illuminants. This step is required because the ground truth of the original LSMI dataset does not consider the differences in brightness caused by having multiple illuminant configurations for images from the same scene. To obtain the ground



(a) **Step 1.** WB correction on the single illuminant image using the Macbeth color checker and render to sRGB (GT before brightness adjustment).



(b) **Step 2.** AWB on the multiple illuminant image and render to sRGB.



(c) **Step 3.** Adjust the per-pixel brightness for the GT image.

Figure 5. Procedure for generating a ground truth (GT) sRGB image for scenes with multiple illuminants. We begin with the dataset from Kim et al. [29], focused on unprocessed RAW images. All scenes are initially captured under a single illuminant, and additional illuminants are introduced individually. (a) We first compute the WB-corrected image for the single-illuminant scene using the Macbeth color checker. (b) Next, we apply AWB to the multiple-illuminant images. (c) Finally, we adjust the per-pixel brightness of the image obtained in Step 1 to generate the ground truth image. We do so by making the pixel brightness of this image match the pixel brightness from the image obtained in Step 2.

truth, we first apply WB correction using the color chart on the single illuminant image, as shown in step 1 of Figure 5. However, this initial ground truth image often has less brightness than an image of the same scene with additional light sources. To correct the brightness discrepancies, we render the multi-illuminant image to sRGB using a standard AWB procedure as shown in step 2 of Figure 5. While AWB does not produce a correct WB image under mixed lighting, it provides a reference for per-pixel brightness normalization of the single-illuminant ground truth image, as shown in step 3 of Figure 5. This brightness adjustment assumes a Lambertian reflectance model, a fair approximation for plausible white-balanced images that maintain spatial consistency with multi-illuminant images. Our dataset provides a valuable benchmark for training and evaluating fusion-based multi-illuminant WB methods, as it introduces real-world variations absent in the synthetic test set [6]. The final dataset includes 16,284 sRGB images from the Nikon and Sony sets. We provide examples from our dataset in the Supplementary Material.

Table 1. Multi-illuminant evaluation. Quantitative evaluation on our dataset presented in Section 5. We compare our method against WBFlow [35], SWBNet [36], DeepWB [2], MixedWB [6], and StyleWB [32]. All the methods are retrained on our dataset. The dashed line divides the conventional sRGB WB methods and the fusion-based methods.

Split	Method	ΔE_{2000}			MSE			MAE		
		Mean	Median	Trimean	Mean	Median	Trimean	Mean	Median	Trimean
Sony	WBFlow [35]	9.64	9.04	9.28	233.88	156.01	185.74	8.42	7.57	7.73
	SWBNet [36]	8.72	8.29	8.42	210.88	142.87	167.42	7.65	6.91	7.02
	DeepWB [2]	<u>7.53</u>	<u>7.14</u>	<u>7.18</u>	<u>195.01</u>	<u>138.39</u>	<u>159.08</u>	<u>6.25</u>	<u>6.08</u>	<u>6.14</u>
	StyleWB [32]	9.46	8.80	9.00	230.76	187.40	193.20	7.89	7.27	7.54
	MixedWB [6]	8.90	8.21	8.27	211.88	153.97	165.03	6.21	5.74	5.82
	Ours	4.71	4.52	4.53	90.27	51.38	54.90	3.75	3.46	3.47
Nikon	WBFlow [35]	9.07	8.57	8.83	206.18	181.57	185.03	6.75	6.25	6.42
	SWBNet [36]	8.23	7.83	7.97	187.61	164.91	169.30	6.10	5.69	5.78
	DeepWB [2]	<u>6.66</u>	<u>6.40</u>	<u>6.47</u>	<u>117.60</u>	<u>94.14</u>	<u>101.70</u>	<u>4.66</u>	<u>4.23</u>	<u>4.44</u>
	StyleWB [32]	9.94	9.57	9.51	247.56	169.28	196.75	8.02	7.26	7.61
	MixedWB [6]	9.85	8.63	9.16	257.65	153.91	176.01	7.02	5.86	6.16
	Ours	3.84	3.55	3.58	48.91	31.59	35.88	3.13	2.83	2.82
Both	WBFlow [35]	9.29	8.79	9.08	223.86	177.04	186.37	7.97	7.72	7.83
	SWBNet [36]	8.43	8.01	8.19	204.81	160.22	169.58	7.18	7.03	7.09
	DeepWB [2]	<u>6.83</u>	<u>6.52</u>	<u>6.58</u>	<u>139.32</u>	<u>124.87</u>	<u>129.05</u>	<u>5.17</u>	<u>4.98</u>	<u>5.03</u>
	StyleWB [32]	9.67	8.96	9.13	234.32	170.38	190.59	7.92	7.41	7.59
	MixedWB [6]	9.31	8.45	8.65	219.51	148.71	162.40	6.36	5.60	5.72
	Ours	4.55	4.45	4.37	75.60	46.88	49.08	3.61	3.37	3.33

Table 2. Cross-camera generalization. Quantitative evaluation on cross-camera generalization of methods with unseen cameras. We present the results for all the metrics when training the models with one camera and testing it on the other split.

Train/Test	Method	ΔE_{2000}			MSE			MAE		
		Mean	Median	Trimean	Mean	Median	Trimean	Mean	Median	Trimean
Sony/Nikon	WBFlow [35]	10.09	9.51	9.86	252.77	186.54	205.31	8.76	7.83	8.63
	SWBNet [36]	9.58	8.91	9.41	238.69	175.65	196.47	8.19	7.41	8.11
	DeepWB [2]	<u>8.91</u>	<u>8.34</u>	<u>8.59</u>	<u>225.82</u>	<u>162.19</u>	<u>185.17</u>	<u>7.44</u>	<u>7.03</u>	<u>7.37</u>
	StyleWB [32]	9.55	8.92	8.95	254.30	217.78	211.05	7.19	6.52	6.77
	MixedWB [6]	9.17	8.60	<u>8.53</u>	246.42	207.80	203.91	<u>6.89</u>	<u>6.27</u>	<u>6.45</u>
	Ours	5.80	5.55	5.50	99.43	67.39	74.35	4.49	3.85	4.05
Nikon/Sony	WBFlow [35]	11.02	10.34	11.00	272.49	205.75	225.02	9.81	8.42	9.60
	SWBNet [36]	10.54	9.64	10.15	266.62	192.16	213.76	9.06	8.30	8.70
	DeepWB [2]	<u>9.44</u>	<u>8.84</u>	<u>9.13</u>	<u>239.47</u>	<u>175.06</u>	<u>197.42</u>	<u>7.99</u>	<u>7.45</u>	<u>7.86</u>
	StyleWB [32]	10.38	10.08	10.23	300.06	246.03	262.13	8.09	7.63	7.92
	MixedWB [6]	9.96	9.70	9.76	289.91	234.54	254.00	<u>7.74</u>	<u>7.34</u>	<u>7.54</u>
	Ours	6.41	6.08	6.16	168.89	116.89	116.44	4.91	4.42	4.46

6. Experiments

We evaluate the performance of our model using our dataset described in Section 5. The images from the Sony and Nikon cameras are organized into training (65%), validation (15%), and testing (20%) splits, ensuring that all images from the same scene are in the same split. Specifically, the Sony split includes 1,092 training images, 252 validation images, and 337 testing images, while the Nikon split comprises 735 training images, 189 validation images, and 207 testing images. Furthermore, we evaluate the generalization with a cross-camera experiment and a cross-dataset evaluation on the Synthetic Multi-Illuminant test set [6].

We also evaluate the performance of our method on single-illuminant scenes, trained on RenderedWB [4] and evaluated on Cube+ [7].

We evaluate the accuracy of white-balance correction using three well-established metrics: ΔE_{2000} , Mean Squared Error (MSE), and Mean Angular Error (MAE). For all metrics, we compute the mean, median, and trimean values. We highlight **best** and **second-best** values for each metric and separate the conventional sRGB methods of the fusion-based method with a dashed line. We compare the performance of our architecture against several recent sRGB white-balance methods, including WBFlow [35], SWB-

Table 3. Cross-dataset generalization. Quantitative evaluation on Synthetic test set [6]. All the methods are retrained on the combined Sony and Nikon splits of our dataset. μ denotes the mean, $\tilde{\mu}$ the median, and TM the trimean.

Method	ΔE_{2000}			MSE			MAE		
	μ	$\tilde{\mu}$	TM	μ	$\tilde{\mu}$	TM	μ	$\tilde{\mu}$	TM
WBFlow	11.86	10.72	10.82	728.96	570.52	634.04	6.08	6.00	5.69
SWBNet	11.54	10.44	10.52	697.31	545.91	555.61	5.92	5.85	5.51
DeepWB	10.93	9.82	10.04	630.60	486.32	517.09	5.53	5.55	5.19
StyleWB	9.90	9.21	9.27	411.88	353.97	365.03	5.21	4.74	4.82
MixedWB	10.43	10.35	10.48	386.39	357.92	363.92	4.75	4.56	4.69
Ours	7.53	6.63	6.79	260.47	238.47	238.31	4.16	3.17	3.31

Net [36], and DeepWB [2], as well as two fusion-based approaches: MixedWB [6] and StyleWB [32]. We follow the official codes and papers to train the comparison methods on our dataset. Specifically, at inference time, we use the *Daylight* preset as the input image for the non-fusion-based methods, as it is the setting with the best ΔE_{2000} . For MixedWB and StyleWB we use the combination of WB presets with better performance. To fit all the models fairly we resize the Nikon images to 614×920 and the Sony images to 500×700 . Note that all the color charts were masked during training to avoid learning trivial solutions.

6.1. Implementation details

We train our method with the Adam optimizer [31] with a cosine learning scheduler, starting with an initial learning rate of $1e-3$ and gradually decreasing to $1e-5$. We employ an L2 loss and select the model that achieves the lowest mean ΔE_{2000} on the validation set.

6.2. Results

Multi-illuminant evaluation. Table 1 summarizes multi-illuminant performance on our dataset. The last set of rows combines Sony and Nikon images (*Both*). Our method substantially outperforms all other approaches across all dataset splits and evaluation metrics. Notably, due to the difficulty of the scenes, standard sRGB WB methods achieve better performance than linear fusion-based methods. Among existing works, DeepWB achieves the best performance. However, our method demonstrates a substantial improvement of 2.28 ΔE_{2000} on the combined Sony and Nikon splits compared to DeepWB.

Cross-camera generalization. Our goal is to develop a WB correction method for multi-illuminant scenes that generalizes well to unseen cameras. That said, we evaluate the models trained on Sony images with Nikon images and vice versa. The results of this experiment are shown in Table 2. Our model consistently outperforms all other methods across all metrics. Notably, even when training on different splits, our approach obtains higher performance.

Table 4. Single illuminant evaluation. Quantitative evaluation on the Cube+ dataset [7]. Results are averaged over all images.

Method	ΔE	MSE	MAE	Param	T(ms)
WBFlow [35]	4.28	87.51	3.15	7.8M	927
SWBNet [36]	4.28	74.35	3.15	2.1M	693
DeepWB [2]	4.59	80.46	3.45	4.4M	459
StyleWB* [32]	4.88	95.82	3.61	15.5M	1893
MixedWB [6]	5.03	168.38	4.20	1.3M	980
Ours	4.19	68.33	2.98	7.9K	179

Cross-dataset generalization. In Table 3, we assess the generalization of the methods trained on our dataset on the Synthetic Multi-Illuminant test dataset [6]. Despite the domain shift between the real images and the synthetic images in this test set, our method outperforms the previous state of the art, as shown in Table 3.

Single illuminant evaluation. Our architecture is designed for multi-illuminant white balance correction. However, we also retrain it using the same approach as MixedWB to demonstrate its superiority over the previous state of the art in image fusion for single illuminant white balance. The results on the Cube+ dataset are presented in Table 4, with models trained on a subset of the RenderedWB dataset [4]. In this experiment, we use the publicly available models except StyleWB which is retrained to match the training sets of the other methods. Our method consistently outperforms both standard sRGB WB methods and fusion-based approaches across all mean metrics.

Efficiency. Table 4 presents the average runtime for processing 50 images with a resolution of 500×700 (Nikon images) on an NVIDIA 3090 GPU. Our method has the fewest number of parameters and the fastest inference time due to the use of transposed attention. Specifically, it is nearly five times faster than DeepWB and has an order of magnitude fewer parameters compared to the other methods.

Qualitative results. Finally, we present qualitative results in Figure 6, comparing our method to DeepWB, MixedWB, and StyleWB. The first row shows a Sony image from our dataset, the next two rows show Nikon images, and the last row presents a rendered image from the synthetic test set [6]. From left to right, we display the WB preset with the lowest ΔE_{2000} , DeepWB, MixedWB, StyleWB, our model, and the corresponding ground truth. In the first two examples, our method achieves superior results by effectively removing the color cast caused by different illuminants, visible on the floor and wooden table, respectively. In the third example, MixedWB and StyleWB perform worse than the Fluorescent WB preset, as evident on the whiteboard, illustrating that estimating a per-pixel blending map can be challenging in some cases. Lastly, the last row shows that despite the domain shift between the training data and

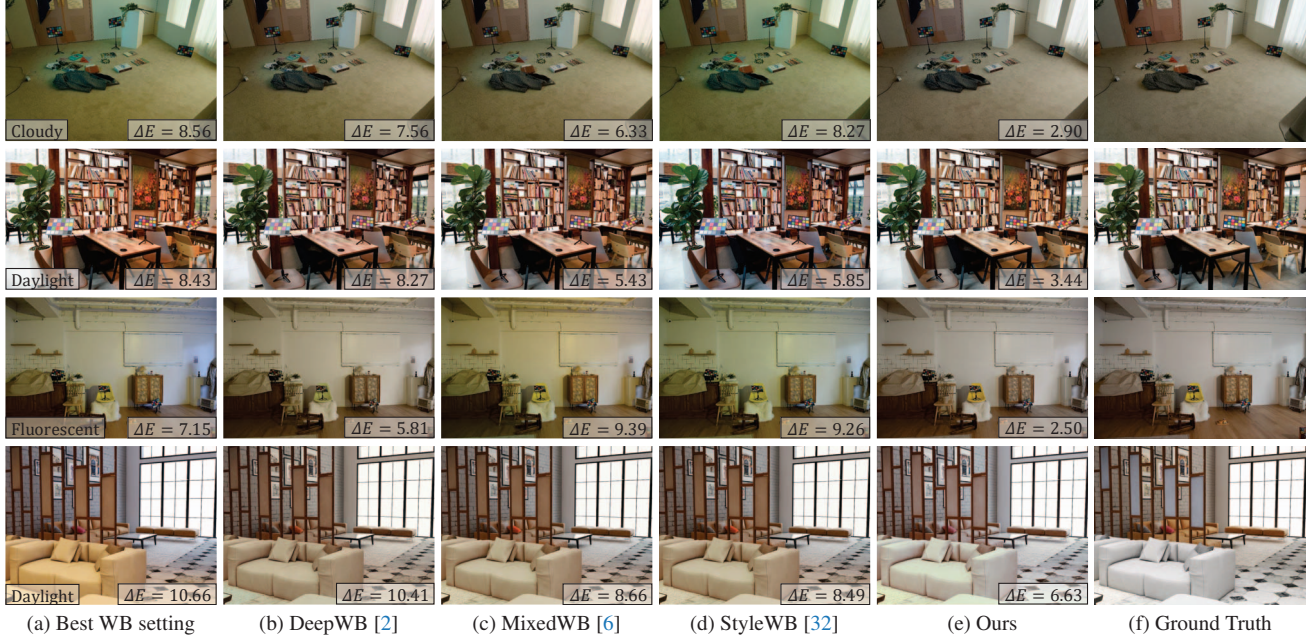


Figure 6. Quantitative results on our dataset and the synthetic test set [6]. Top to bottom: an image of the Sony split, two Nikon images, and the synthetic dataset. Left to right: the results of the WB preset with lowest ΔE_{2000} (a), DeepWB [2] (b), MixedWB [6] (c), StyleWB [32] (d), our transformer-based method (e) and the ground truth (f). In the first column (a), we show the name of the WB preset in the bottom-left corner. The ΔE_{2000} value for each image for each image is shown in the bottom-right corner, lower values indicate higher performance.

Table 5. Study on the number of presets. Models are trained and evaluated on our dataset’s combined Sony and Nikon splits. μ denotes the mean, $\tilde{\mu}$ the median, and TM the trimean.

Method	ΔE_{2000}			MAE		
	μ	$\tilde{\mu}$	TM	μ	$\tilde{\mu}$	TM
Daylight	5.47	5.23	5.19	4.27	3.88	4.02
DST	5.18	4.94	4.96	4.06	3.74	3.76
All presets	4.55	4.45	4.37	3.61	3.37	3.33

the synthetic images from Afifi et al. [6], our method produces results closest to the ground truth, as observed on the white sofa.

6.3. Ablation Study

Standard sRGB WB vs fusion-based. As seen in Tables 1 and 2, in complex cases such as those in our dataset, linear fusion-based methods struggle to produce accurate WB images and are outperformed by regular sRGB WB methods. To further evaluate the effectiveness of our non-linear fusion strategy, we conduct additional experiments to compare it against regular sRGB WB. Specifically, we modify our transformer block to use only the *Daylight* preset as input and, separately, to use three input presets (DST) by adjusting the first convolutional layer. Table 5 reports the ΔE_{2000} and MAE values for these two cases, as well as for our proposed approach using all five presets in the Sony and Nikon splits of our dataset (Both in Table 1). Even when using only the *Daylight* preset, our model outperforms previ-

ous state-of-the-art methods. However, we observe that performance improves as more presets are incorporated. This suggests that the model benefits from additional presets, as they provide more information on how different illuminants interact under varying color temperatures.

7. Conclusion

In this work, we introduced a transformer-based approach for WB correction in multi-illuminant scenes by efficiently fusing WB presets. Our method outperforms previous linear fusion-based approaches and achieves robust generalization in unseen cameras and datasets without the need to compute per-pixel blending maps. By leveraging transposed attention, our approach demonstrated superior adaptability, and efficiency compared to the state of the art. Additionally, we introduced an sRGB dataset tailored for multi-illuminant WB correction. We anticipate that our dataset will support further advancement in multi-illuminant white-balance research.

This work was partially supported by grants PID2021-128178OB-I00, PID2024-162555OB-I00 funded by MCIN/AEI/10.13039/501100011033 and by ERDF/EU, and by the Generalitat de Catalunya — Departament de Recerca i Universitats with reference 2021SGR01499 and CERCA Program. DSL also acknowledges the FPI grant from the Spanish Ministry of Science and Innovation (PRE2022-101525). LH was also supported by the Ramon y Cajal grant RYC2019-027020-I. This work was also partially supported by the grant Càtedra ENIA UAB-Cruïlla (TSI-100929-2023-2) from the Ministry of Economic Affairs and Digital Transition of Spain. This work was funded in part by the CFREF (VISTA) program, an NSERC Discovery Grants, and the Canada Research Chair program.

References

- [1] Adobe Systems. Adobe Camera Raw. Computer software. 5
- [2] Mahmoud Afifi and Michael S Brown. Deep white-balance editing. In *CVPR*, 2020. 1, 2, 6, 7, 8
- [3] Mahmoud Afifi and Michael S Brown. Interactive white balancing for camera-rendered images. In *CIC*, 2020. 2
- [4] Mahmoud Afifi, Brian Price, Scott Cohen, and Michael S Brown. When color constancy goes wrong: Correcting improperly white-balanced images. In *CVPR*, 2019. 1, 2, 3, 4, 5, 6, 7
- [5] 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 *CIC*, 2019. 2
- [6] Mahmoud Afifi, Marcus A Brubaker, and Michael S Brown. Auto white-balance correction for mixed-illuminant scenes. In *WACV*, 2022. 1, 2, 3, 5, 6, 7, 8
- [7] Nikola Banić, Karlo Koščević, and Sven Lončarić. Unsupervised learning for color constancy. *arXiv preprint arXiv:1712.00436*, 2017. 2, 3, 5, 6, 7
- [8] Jonathan T. Barron and Yun-Ta Tsai. Fast Fourier color constancy. In *CVPR*, 2017. 1
- [9] Shida Beigpour, Christian Riess, Joost Van De Weijer, and Elli Angelopoulou. Multi-illuminant estimation with conditional random fields. *TIP*, 23(1):83–96, 2013. 2
- [10] Simone Bianco and Claudio Cusano. Quasi-unsupervised color constancy. In *CVPR*, 2019. 2
- [11] David H Brainard and William T Freeman. Bayesian color constancy. *JOSA A*, 1997. 2
- [12] George Buchsbaum. A spatial processor model for object color perception. *J. Frank. Inst.*, 1980. 1, 2
- [13] Ayan Chakrabarti, Keigo Hirakawa, and Todd Zickler. Color constancy beyond bags of pixels. In *CVPR*, 2008.
- [14] 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. *JOSA A*, 2014. 2
- [15] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*, 2021. 4
- [16] Graham D Finlayson. Corrected-moment illuminant estimation. In *ICCV*, 2013. 2
- [17] Graham D Finlayson and Elisabetta Trezzi. Shades of gray and colour constancy. In *CIC*, 2004. 2
- [18] Graham D Finlayson, Brian V Funt, and Kobus Barnard. Color constancy under varying illumination. In *ICCV*, 1995. 2
- [19] Graham D Finlayson, Paul M Hubel, and Steven Hordley. Color by correlation. In *CIC*, 1997. 2
- [20] Peter Vincent Gehler, Carsten Rother, Andrew Blake, Tom Minka, and Toby Sharp. Bayesian color constancy revisited. In *CVPR*, 2008. 2
- [21] Arjan Gijsenij, Theo Gevers, and Joost Van De Weijer. Computational color constancy: Survey and experiments. *TIP*, 20(9):2475–2489, 2011. 1
- [22] Arjan Gijsenij, Theo Gevers, and Joost Van De Weijer. Improving color constancy by photometric edge weighting. *IEEE TPAMI*, 34(5):918–929, 2011. 2
- [23] Arjan Gijsenij, Rui Lu, and Theo Gevers. Color constancy for multiple light sources. *TIP*, 2012. 2
- [24] Daniel Hernandez-Juarez, Sarah Parisot, Benjamin Busam, Ales Leonardis, Gregory Slabaugh, and Steven McDonagh. A multi-hypothesis approach to color constancy. In *CVPR*, 2020. 2
- [25] Eugene Hsu, Tom Mertens, Sylvain Paris, Shai Avidan, and Frédo Durand. Light mixture estimation for spatially varying white balance. In *ACM SIGGRAPH*, 2008. 2
- [26] Yuanming Hu, Baoyuan Wang, and Stephen Lin. FC4: Fully convolutional color constancy with confidence-weighted pooling. In *CVPR*, 2017. 1
- [27] Md Akmol Hussain and Akbar Sheikh Akbari. Color constancy algorithm for mixed-illuminant scene images. *IEEE Access*, 2018. 2
- [28] Hamid Reza Vaezi Joze and Mark S Drew. Exemplar-based color constancy and multiple illumination. *PAMI*, 36(5):860–873, 2013. 2
- [29] Dongyoung Kim, Jinwoo Kim, Seonghyeon Nam, Dongwoo Lee, Yeonkyung Lee, Nahyup Kang, Hyong-Euk Lee, Byung-In Yoo, Jae-Joon Han, and Seon Joo Kim. Large scale multi-illuminant (LSMI) dataset for developing white balance algorithm under mixed illumination. In *ICCV*, 2021. 2, 3, 5
- [30] Dongyoung Kim, Jinwoo Kim, Junsang Yu, and Seon Joo Kim. Attentive illumination decomposition model for multi-illuminant white balancing. In *CVPR*, 2024. 2
- [31] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 7
- [32] Furkan Kınlı, Doğa Yılmaz, Barış Özcan, and Furkan Kırac. Modeling the lighting in scenes as style for auto white-balance correction. In *WACV*, 2023. 3, 5, 6, 7, 8
- [33] Edwin H Land and John J McCann. Lightness and retinex theory. *JOSA A*, 1971. 2
- [34] Chongyi Li, Chunle Guo, Shangchen Zhou, Qiming Ai, Ruicheng Feng, and Chen Change Loy. Flexicurve: Flexible piecewise curves estimation for photo retouching. In *CVPR*, 2023. 4
- [35] Chunxiao Li, Xuejing Kang, and Anlong Ming. Wbflow: Few-shot white balance for srgb images via reversible neural flows. In *International Joint Conference on Artificial Intelligence*, 2023. 2, 6, 7
- [36] Chunxiao Li, Xuejing Kang, Zhifeng Zhang, and Anlong Ming. Swbnet: A stable white balance network for sRGB images. In *AAAI*, 2023. 2, 6, 7
- [37] 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 *CVPR*, 2021. 2

- [38] Zhongyu Lou, Theo Gevers, Ninghang Hu, and Marcel P Lucassen. Color constancy by deep learning. In *BMVC*, 2015.
- [39] Seoung Wug Oh and Seon Joo Kim. Approaching the computational color constancy as a classification problem through deep learning. *PR*, 61:405–416, 2017. [2](#)
- [40] Yanlin Qian, Joni-Kristian Kamarainen, Jarno Nikkanen, and Jiri Matas. On finding gray pixels. In *CVPR*, 2019. [2](#)
- [41] Guillermo Sapiro. Color and illuminant voting. *PAMI*, 1999. [2](#)
- [42] Wu Shi, Chen Change Loy, and Xiaoou Tang. Deep specialized network for illuminant estimation. In *ECCV*, 2016. [2](#)
- [43] Oleksii Sidorov. Conditional GANs for multi-illuminant color constancy: Revolution or yet another approach? In *CVPRW*, 2019. [1](#)
- [44] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. [3](#)
- [45] Joost Van De Weijer, Theo Gevers, and Arjan Gijsenij. Edge-based color constancy. *TIP*, 16(9):2207–2214, 2007. [2](#)
- [46] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. [2](#)
- [47] Javier Vazquez, C Alejandro Párraga, Maria Vanrell, and Ramon Baldrich. Color constancy algorithms: Psychophysical evaluation on a new dataset. *Journal of Imaging Science and Technology*, 1(3):1, 2009. [2](#)
- [48] Javier Vazquez-Corral, Maria Vanrell, Ramon Baldrich, and Francesc Tous. Color constancy by category correlation. *TIP*, 21(4):1997–2007, 2011. [1](#), [2](#)
- [49] Donik Vršnak, Ilija Domislović, Marko Subašić, and Sven Lončarić. Autoencoder-based training for multi-illuminant color constancy. *JOSA A*, 2022. [1](#)
- [50] Donik Vršnak, Ilija Domislović, Marko Subašić, and Sven Lončarić. Illuminant segmentation for multi-illuminant scenes using latent illumination encoding. *Signal Process. Image Commun.*, 2022. [1](#)
- [51] Bolei Xu, Jingxin Liu, Xianxu Hou, Bozhi Liu, and Guoping Qiu. End-to-end illuminant estimation based on deep metric learning. In *CVPR*, 2020. [2](#)
- [52] Shuwei Yue and Minchen Wei. Robust pixel-wise illuminant estimation algorithm for images with a low bit-depth. *Optics Express*, 32(15):26708–26718, 2024. [2](#)
- [53] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang. Restormer: Efficient transformer for high-resolution image restoration. In *CVPR*, 2022. [4](#)
- [54] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks. In *ICML*, 2019. [4](#)
- [55] Simone Zini, Marco Buzzelli, Simone Bianco, and Raimondo Schettini. Cocoa: combining color constancy algorithms for images and videos. *IEEE Transactions on Computational Imaging*, 8:795–807, 2022. [2](#)