WBFlow: Few-shot White Balance for sRGB Images via Reversible Neural Flows

Chunxiao Li, **Xuejing Kang**, **Anlong Ming**[*]
School of Computer Science (National Pilot Software Engineering School), Beijing University of Posts and Telecommunications
{chunxiaol, kangxuejing, mal}@bupt.edu.cn

Abstract

The sRGB white balance methods aim to correct the nonlinear color cast of sRGB images without accessing raw values. Although existing methods have achieved increasingly better results, their generalization to sRGB images from multiple cameras is still under explored. In this paper, we propose the network named WBFlow that not only performs superior white balance for sRGB images but also generalizes well to multiple cameras. Specifically, we take advantage of neural flow to ensure the reversibility of WBFlow, which enables lossless rendering of color cast sRGB images back to pseudo raw features for linear white balancing and thus achieves superior performance. Furthermore, inspired by camera transformation approaches, we have designed a camera transformation (CT) in pseudo raw feature space to generalize WBFlow for different cameras via few shot learning. By utilizing a few sRGB images from an untrained camera, our WBFlow can perform well on this camera by learning the camera specific parameters of CT. Extensive experiments show that WBFlow achieves superior camera generalization and accuracy on three public datasets as well as our rendered multiple camera sRGB dataset. Our code is available at https://github.com/ChunxiaoLe/WBFlow.

1 Introduction

The color cast in sRGB images is caused by improper color temperature settings in the white balance (WB) module and then exacerbated by non-linear color renderings in image signal processor (ISP) [Afifi *et al.*, 2019a]. The sRGB White balance (sRGB-WB) methods aim to correct such color cast without accessing raw values. It is an emerging direction and has an essential impact on some high-level computer vision tasks, such as object recognition and image segmentation [Afifi and Brown, 2019].

To perform accurate sRGB-WB, an ideal way is to reverse color-cast sRGB images to unprocessed raw values for linear correcting and then re-render them to obtain white-balanced sRGB images [Afifi and Brown, 2020a]. However, this process is challenging for sRGB-WB methods because the essential reversibility is difficult to achieve without raw values [Chakrabarti *et al.*, 2014]. In fact, instead of the ideal WB process, recent sRGB-WB methods [Afifi *et al.*, 2019a; Afifi *et al.*, 2019b; Afifi and Brown, 2020b] directly utilize *irreversible* polynomial kernel functions [Hong *et al.*, 2001] to train the mappings between color-cast sRGB images and ground truths (GTs). These trained mappings are not appropriate on untrained cameras due to the inherent color differences among sRGB images from different cameras [Afifi, 2021]. This seriously limits cross-camera generalization in real-world applications. In Figure 1(b), the representative sRGB-WB method, KNN-WB [Afifi *et al.*, 2019a], results in a noticeable greenish cast on the untrained camera.

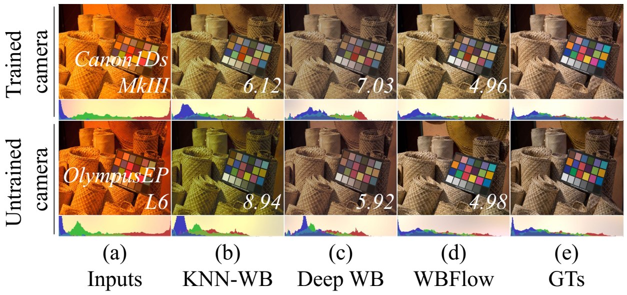


Figure 1: Existing sRGB-WB methods are not effective in generalizing well to multiple cameras: KNN-WB [Afifi *et al.*, 2019a] has a negative WB performance on the untrained camera, as in obvious greenish output in the second row of (b). Although Deep WB [Afifi and Brown, 2020a] suppresses this color bias, it performs worse on the trained camera, as in the output with dark colors in the first row of (c). Comparatively, our WBFlow shows superior performance on both trained and untrained cameras. As shown in (d), it generates the outputs with similar color histograms to GTs in (e).

Few endeavors have been made to address this problem. The only method, Deep WB [Afifi and Brown, 2020a], proposes modeling the ideal WB process end-to-end via U-Net [Ronneberger *et al.*, 2015]. It further generalizes to different cameras by jointly training multi-camera sRGB images. However, due to the lossy structure of U-Net and the vanilla training strategy, Deep WB becomes *irreversible* and suppresses camera specificity. Thus, in Figure 1(c), although

[*]Corresponding Author.

Deep WB improves generalization to the untrained camera, it has a degraded performance on the trained camera.

In this paper, we propose an sRGB-WB network called WBFlow that achieves superior performance on both trained and untrained cameras. Unlike existing *irreversible* sRGB-WB methods, we take advantage of neural flows [Kingma and Dhariwal, 2018] to enable the *reversibility* of WBFlow. Specifically, we design Reversible Non-linear Rendering Transformation (RNRT) that simulates ISP color renderings by additive coupling layers [Kingma and Dhariwal, 2018]. Due to the inherent reversibility of additive coupling layers, RNRT can losslessly render color-cast sRGB images back to their pseudo-raw features via forward propagation. Then, to correct the color cast of pseudo-raw features, we present Reversible Linear Correction Transformation (RLCT) that simulates white balance and color transformation matrices as reversible 1×1 convolutions [Kingma and Dhariwal, 2018]. RLCT helps RNRT lossless back-propagation to generate white-balanced sRGB images by separating and correcting the color information from pseudo-raw features. Further, inspired by the fact that inter-camera transformation can be achieved by learning the mapping between raw values from different cameras [Banić *et al.*, 2017], we generalize WBFlow to multiple cameras via inserting a camera transformation (CT) layer in pseudo-raw space. Specifically, by utilizing a few sRGB images from an untrained camera, we update the weightings of CT to enhance the camera specificity of pseudo-raw features, which cooperates with RNRT and RLCT to generate white-balanced sRGB images. Our contributions are as follows:

(1) We propose WBFlow that generalizes well to multiple cameras via reversible neural flows and few-shot learning.

(2) We propose RNRT and RLCT to simulate nonlinear color renderings and linear color transformations to enable the reversibility of WBFlow.

(3) We propose CT that generalizes WBFlow to untrained cameras by updating its weights via few-shot learning.

(4) Extensive experiments show that WBFlow achieves superior multi-camera generalization on three public datasets and our rendered multi-camera sRGB dataset.

2 Related Works

White Balance Methods for Raw Images. The white balance methods for raw images fall into computational color constancy (CCC) research. Most CCC methods [Gijsenij and Gevers, 2007; Foster, 2011; Hu *et al.*, 2017; Lo *et al.*, 2021; Song *et al.*, 2021; Xu *et al.*, 2021; Tang *et al.*, 2022; Zhang *et al.*, 2022] achieve the goal of correcting the color cast in raw images by estimating illumination colors. They make up the WB module of camera ISPs. However, none of these methods can be applied to sRGB images [Afifi *et al.*, 2019a], since the linear color cast of raw images is broken due to non-linear ISP renderings. Different from CCC methods, our WBFlow focuses on correcting the non-linear color cast of sRGB images.

White Balance Methods for sRGB Images. Recent sRGB-WB methods corrected the color cast in sRGB images by optimizing non-linear mappings (exemplar-based method) [Afifi *et al.*, 2019a; Afifi *et al.*, 2019b; Afifi and Brown, 2020b; Afifi and Brown, 2019; Afifi *et al.*, 2020] or modeling ideal WB process (DNN-based method) [Afifi and Brown, 2020a]. In detail, exemplar-based methods combined trained non-linear mappings to correct inputs. Since these mappings are irreversible, the specificities of trained cameras are mixed and applied to encode white-balanced sRGB images during testing, making them biased towards trained cameras rather than test ones. Inevitably, the performances of exemplar-based methods are inferior on untrained cameras (Figure 1(b)). To solve this problem, the DNN-based method, Deep WB, proposed rendering inputs back to pseudo-raw features by U-Net encoder, then correcting pseudo-raw features and re-rendering them to white-balanced sRGB images by U-Net decoder. It also allowed WB manipulation in sRGB images by setting two additional decoders to render inputs with 2850 Kelvin(K) and 7500K. However, due to lossy pooling layers, Deep WB is irreversible, making the content of pseudo-raw features less complete than actual raw values and further restricting WB accuracy. Further, the joint training strategy suppressed the specificities of different cameras [Huang *et al.*, 2020], and hence together with irreversible structure, limited the improvement of multi-camera generalization (Figure 1(c)). Our WBFlow belongs to DNN-based methods and achieves superior generalization to multiple cameras via reversible neural flows and few-shot learning.

3 White Balance for sRGB Image via Reversible Flows and Few-shot Learning

In this section, we introduce WBFlow that aims to achieve superior generalization on multiple cameras. We start with formulating the ideal WB process for sRGB images. Then, we introduce details of WBFlow and how it generalizes to multiple cameras via few-shot learning.

3.1 Ideal White Balance Process for sRGB Images

An sRGB image $\mathbf{I}_{ct}^{n}$ that is captured by camera n and rendered by color temperature ct can be formed as [Afifi *et al.*, 2019a]:

$$\mathbf{I}_{ct}^{n} = f_n(\mathbf{T}_n \mathbf{W}_{ct} \mathbf{I}_{raw}^{n}), \tag{1}$$

where $\mathbf{I}_{raw}^{n}$ is a raw image from camera n, $\mathbf{W}_{ct}$ is a WB module with color temperature ct, $\mathbf{T}_n$ is a color transformation matrix for camera n, $f_n(\cdot)$ is a non-linear rendering function, including color enhancement, tone manipulation, and gamma encoding *et al.*. Corresponding correct sRGB image $\mathbf{I}_{wb}^{n}$, *i.e.* GT, is obtained by manually setting $\mathbf{W}_{ct}$ according to actual illumination color. Here, we denote this ideal setting as $\mathbf{W}_{ideal}$. Therefore, the relationship between $\mathbf{I}_{wb}^{n}$ and $\mathbf{I}_{ct}^{n}$ can be modeled as:

$$\mathbf{I}_{wb}^{n} = f_n(\mathbf{T}_n \mathbf{W}_{ideal} \mathbf{W}_{ct}^{-1} \mathbf{T}_n^{-1} f_n^{-1}(\mathbf{I}_{ct}^{n})). \tag{2}$$

From Equation 2, the ideal WB process is reversible to render color-cast sRGB image toward its white-balanced version while preserving complete content. Specifically, $\mathbf{I}_{ct}^{n}$ is first reversed back to its color-cast linear values via $f_n^{-1}(\cdot)$; then, $\mathbf{T}_n$ transforms color-cast linear values into raw values and then $\mathbf{W}_{ideal}\mathbf{W}_{ct}^{-1}$ white balance them; finally, $f_n(\cdot)$ and $\mathbf{T}_n$ render white-balanced raw values to obtain $\mathbf{I}_{wb}^{n}$.

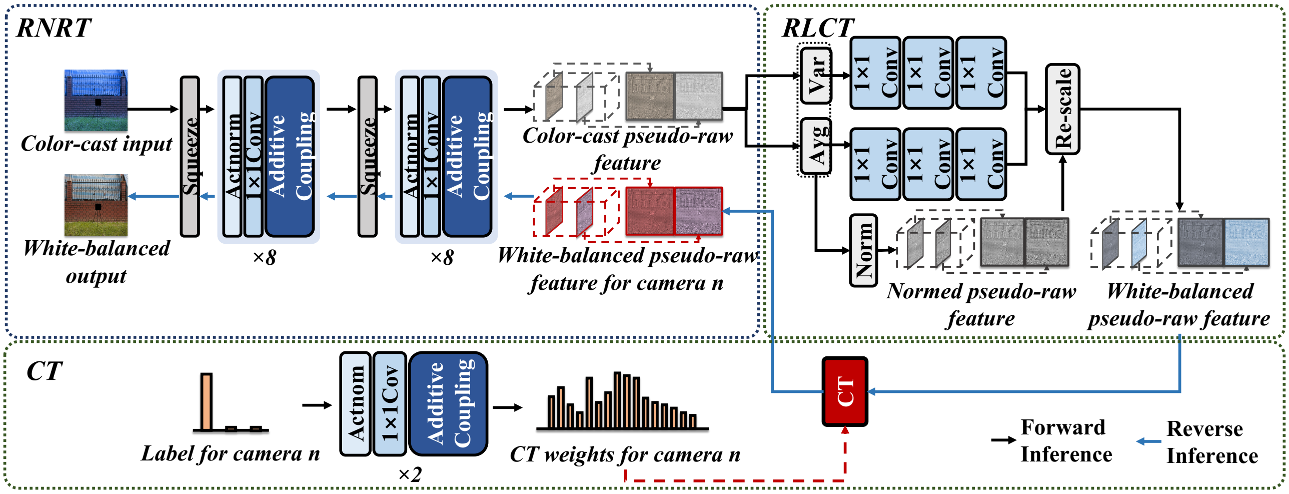


Figure 2: Illustration of our WBFlow. We propose RNRT and RLCT to ensure the reversibility of WBFlow and thus improve WB accuracy. CT is then applied in pseudo-raw space to generalize WBFlow for multiple cameras via few-shot learning.

As discussed in Section 1 and 2, existing sRGB-WB methods [Afifi *et al.*, 2019a; Afifi *et al.*, 2020; Afifi and Brown, 2020b; Afifi *et al.*, 2021; Afifi and Brown, 2020a] are *irreversible* and thus fail to satisfy Equation 2, which limits their generalization to multiple cameras. In this paper, we propose an sRGB-WB network named WBFlow that satisfies the reversibility of Equation 2 and generalizes well to cameras via few-shot learning. Details will be introduced as follows.

3.2 White Balance for sRGB Images via Reversible Transformations

To achieve accurate white balance on multiple cameras, our primary goal is to appropriately model the ideal WB process. According to Equation 2, this process has two types of modules: nonlinear color rendering function ($f_n(\cdot)$) and linear color transformations ($\mathbf{W}_{ct}$, $\mathbf{W}_{ideal}$, $\mathbf{T}_n$). Deep WB [Afifi and Brown, 2020a] models these two kinds of modules via identical irreversible structures, which only improves multi-camera generalization to some extent. To solve this problem, different from Deep WB, we model two kinds of modules by individual reversible functions:

$$\hat{\mathbf{I}}^n_{wb} = \mathcal{F}^{-1}(\mathcal{M}(\mathcal{F}(\mathbf{I}^n_{ct}))), \tag{3}$$

where $\mathcal{F}(\cdot)$ is a reversible non-linear function that losslessly renders color-cast sRGB image $\mathbf{I}^n_{ct}$ back to pseudo-raw features, $\mathcal{M}(\cdot)$ is a reversible linear function that corrects the color cast of pseudo-raw features and preserves the content information, $\mathcal{F}^{-1}(\cdot)$ is the inverse of $\mathcal{F}(\cdot)$ that re-renders white-balanced pseudo-raw features to corresponding sRGB image $\hat{\mathbf{I}}^n_{wb}$. Theoretically, due to $\mathcal{F}(\cdot)$ and $\mathcal{M}(\cdot)$, Equation 3 satisfies the reversibility of Equation 2 to generate accurate white-balanced sRGB images on different cameras. Following are the implementation details.

Reversible Non-linear Rendering Transformation

As discussed, the reversible non-linear function $\mathcal{F}(\cdot)$ aims to obtain pseudo-raw features with lossless content. Deep WB [Afifi and Brown, 2020a] cannot satisfy it because of lossy pooling layers. Here, considering color renderings included in $\mathcal{F}(\cdot)$ are independent [Ramanath *et al.*, 2005], we model them by a sequence of reversible bijective sub-functions: $\mathcal{F} = \mathcal{F}_1 \circ \mathcal{F}_2 \circ \cdots \circ \mathcal{F}_K$. This way, the relationship between color-cast sRGB image $\mathbf{I}^n_{ct}$ and intermediate feature $\mathbf{z}^n_{ct,k}$ is:

$$\mathbf{I}^n_{ct} \overset{\mathcal{F}_1}{\longleftrightarrow} \mathbf{z}^n_{ct,1} \overset{\mathcal{F}_2}{\longleftrightarrow} \mathbf{z}^n_{ct,2} \cdots \overset{\mathcal{F}_K}{\longleftrightarrow} \mathbf{z}^n_{ct,K}, \tag{4}$$

where $\mathbf{z}^n_{ct,K} = \mathbf{z}^n_{ct}$ is pseudo-raw feature.

In practice, we adopt additive coupling layers [Kingma and Dhariwal, 2018] to model $\{\mathcal{F}_k\}$, which are effective in simulating nonlinear reversible transformations [Kingma and Dhariwal, 2018; An *et al.*, 2021]. In our WBFlow, the forward computation of the additive coupling layer is:

$$\begin{aligned} \mathbf{z}^{n,a}_{ct,k}, \mathbf{z}^{n,b}_{ct,k} &= Split(\mathbf{z}^n_{ct,k}), \\ \hat{\mathbf{z}}^{n,a}_{ct,k} &= Conv(\mathbf{z}^{n,a}_{ct,k}) + \mathbf{z}^{n,b}_{ct,k}, \\ \mathbf{z}^n_{ct,k+1} &= Concat(\hat{\mathbf{z}}^{n,a}_{ct,k}, \mathbf{z}^{n,b}_{ct,k}), \end{aligned} \tag{5}$$

where $Split(\cdot)$ splits $\mathbf{z}^n_{ct,k}$ into two parts along channel dimension to obtain $\mathbf{z}^{n,a}_{ct,k}$ and $\mathbf{z}^{n,b}_{ct,k}$. $Conv(\cdot)$ renders colors of $\mathbf{z}^{n,a}_{ct,k}$ with 3×3 convolutions and relu functions. $Concat(\cdot)$ concatenates $\hat{\mathbf{z}}^{n,a}_{ct,k}$ and $\mathbf{z}^{n,b}_{ct,k}$ to form rendered feature $\mathbf{z}^n_{ct,k+1}$. The reverse computation of the additive coupling layer is easily derived from Equation 5. We can superimpose additive coupling layers to establish the reversible mapping between input color-cast sRGB image and pseudo-raw feature as Equation 4. However, as in Equation 5, the additive coupling layer leaves some channels of intermediate feature unchanged due to $Split(\cdot)$. To solve this problem, following [Kingma and Dhariwal, 2018], we use reversible 1×1 convolution that integrates all channels by fixing the channel numbers of input and output as the same. We also adopt Actnorm [Kingma and Dhariwal, 2018] to normalize features, an alternative to BatchNorm that accelerates the training times.

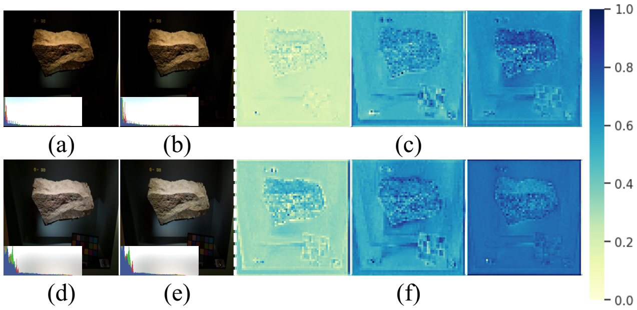


Figure 3: Pseudo-raw features with lossless content in (c) is obtained from color-cast input in (a) via forward RNRT. Restored input in (b) is re-rendered from pseudo-raw features in (c) via reverse RNRT. White-balanced pseudo-raw features in (f) are obtained by RLCT and then re-rendered back to the white-balanced sRGB image in (e) via reverse RNRT.

The proposed RNRT is shown in Figure 2, consisting of additive coupling layers, reversible 1×1 convolutions, and Actnorms. All components of RNRT are reversible, making content lossless during in forward and backward propagations. As shown in Figure 3(c), we can obtain pseudo-raw features with lossless content from color-cast input in Figure 3(a) via forward RNRT. Simultaneously, in Figure 3(b), the input can be restored accurately from pseudo-raw features in Figure 3(c) via reverse RNRT.

Reversible Linear Correction Transformation

In Equation 2 and 3, the reversible linear function $\mathcal{M}(\cdot)$ is linear and reversible due to the WB module and color transformation matrix. To properly implement it while preserving lossless content information, inspired by [Huang and Belongie, 2017], we propose RLCT that separates color and content information from pseudo-raw features and corrects the color information by reversible 1×1 convolutions [Kingma and Dhariwal, 2018]. The formation is:

$$\mathbf{z}_{wb}^{n} = \mathcal{T}_{\sigma}(\sigma_z)(\frac{\mathbf{z}_{ct}^{n} - \mu_z}{\sigma_z}) + \mathcal{T}_{\mu}(\mu_z), \tag{6}$$

where channel mean and variance, μ_z and σ_z, represents the color information of pseudo-raw feature $\mathbf{z}_{ct,k}^{n}$ [Stricker and Orengo, 1995]. They also help to keep content information by normalizing $\mathbf{z}_{ct}^{n}$, *i.e.*, $(\mathbf{z}_{ct}^{n} - \mu_z)/\sigma_z$. For individual color information, we correct it by respectively forwarding μ_z and σ_z into linear mappings $\mathcal{T}_{\sigma}(\cdot)$ and $\mathcal{T}_{\mu}(\cdot)$, both of which consist of three reversible 1×1 convolutions [Kingma and Dhariwal, 2018]. This way, in Figure 3(f), the pseudo-raw features are white-balanced by re-scaling lossless content with learned channel mean $\mathcal{T}_{\sigma}(\sigma_z)$ and variance $\mathcal{T}_{\mu}(\mu_z)$. Further, in Figure 3(e), the white-balanced sRGB image is successfully generated from white-balanced pseudo-raw features in Figure 3(f) via reverse propagation of RNRT. It has an almost identical color histogram as GT in Figure 3(d).

As introduced above, RNRT and RLCT are reversible and thus cooperate to ensure complete reversibility, which satisfies the requirement of Equation 2. Thus, our WBFlow can significantly improve the floor of WB accuracy on multiple cameras. As shown in Figure 4(a), with the same multi-camera training strategy of Deep WB, our WBFlow converges with a much lower l_1 loss on multiple cameras.

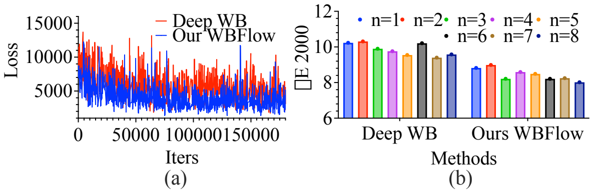


Figure 4: Comparison between Deep WB and WBFlow: (a) multi-camera training loss and (b) generalizations on a new camera (PanasonicGX1) (n is the number of training cameras).

3.3 Camera Generalization via Few-shot Learning

According to Equation 2, the actual raw features should be specific to various cameras. However, existing methods suppressed such specificities due to combined trained mappings [Afifi *et al.*, 2019a; Afifi and Brown, 2020b] or joint multi-camera training strategy [Afifi and Brown, 2020a]. Inevitably, this will result in a limited multi-camera generalization. To enhance the camera specificity of pseudo-raw feature, we introduce few-shot learning into WBFlow, which is based on the fact that inter-camera transformation can be achieved by learning the mapping between raw color values from two cameras [Gao *et al.*, 2017]. In Figure 2, we set the camera transformation named CT after RNRT. For camera n, Equation 6 can be re-written as:

$$\hat{\mathbf{z}}_{wb}^{n} = \mathbf{w}_{cam}^{n} * \left\{ \mathcal{T}_{\sigma}(\sigma_z)(\frac{\mathbf{z}_{ct}^{n} - \mu_z}{\sigma_z}) + \mathcal{T}_{\mu}(\mu_z) \right\}, \tag{7}$$

where $*$ is the group convolution operator, $\mathbf{w}_{cam}^{n}$ is a set of weights learned from the camera labels by two reversible neural flow combinations. This strategy ensures that the learned weights are unique to the different cameras. To adapt WBFlow to camera n, we fix the pre-trained parameter of RNRT, then train the new weights in Equation 7 by minimizing the l_1 loss with a few samples. This way, the specificity of pseudo-raw features to camera n is enhanced, thus improving the generalization effect. In Figure 4(b), the $\Delta E2000$ values of WBFlow on the new camera, PanasonicGX1, are always lower than these of Deep WB, regardless of the number of training cameras. More details of few-shot learning settings will be introduced in Section 4.

4 Experiments

In this section, we compare the effectiveness of WBFlow with state-of-the-art sRGB-WB methods in terms of multi-camera generalization, inference time, and few-shot learning. In addition, ablation experiments are designed to evaluate the effectiveness of each part in WBFlow.

4.1 Datasets

Public Datasets. Following [Afifi and Brown, 2020a], we randomly selected 12000 sRGB images from the first fold of Set1 [Afifi *et al.*, 2019a; Cheng *et al.*, 2014] to train WBFlow.

Methods	$\Delta E2000\downarrow$				Mean Angle Error (MAE) $\downarrow$				Inference Time (s) $\downarrow$
	Mean	Q1	Q2	Q3	Mean (°)	Q1 (°)	Q2 (°)	Q3 (°)	
Set1–Test (trained cameras)									
KNN-WB[Afifi *et al.*, 2019a]	3.58	2.07	3.09	4.55	3.06	1.74	2.54	3.76	**0.78**
MixedWB[Afifi *et al.*, 2021]	4.55	3.00	4.15	5.63	4.07	2.64	3.68	5.16	1.43
Deep WB[Afifi and Brown, 2020a]	3.77	2.16	3.30	4.86	3.12	1.88	2.70	3.84	1.10
Our WBFlow	**3.13**	**1.92**	**2.79**	**3.94**	**2.67**	**1.73**	**2.39**	**3.24**	1.09
Set2 (untrained cameras)									
KNN-WB[Afifi *et al.*, 2019a]	5.60	3.43	4.90	7.06	4.48	2.26	3.64	5.95	**0.82**
MixedWB[Afifi *et al.*, 2021]	6.05	3.45	4.92	7.20	4.92	2.69	4.10	6.37	1.55
Deep WB[Afifi and Brown, 2020a]	4.90	**3.13**	4.35	6.08	3.75	2.02	3.08	4.72	1.09
Our WBFlow	**4.64**	3.16	**4.07**	**5.56**	**3.51**	**1.93**	**2.92**	**4.47**	1.08
Rendered Cube Dataset (untrained cameras)									
KNN-WB[Afifi *et al.*, 2019a]	5.68	3.22	4.61	6.70	4.12	1.96	3.17	5.04	**0.81**
MixedWB[Afifi *et al.*, 2021]	5.03	2.07	3.12	7.19	4.20	1.39	2.18	5.54	1.52
Deep WB[Afifi and Brown, 2020a]	4.59	**2.68**	3.81	5.53	3.45	**1.87**	2.82	4.26	1.08
Our WBFlow	**4.28**	2.71	**3.77**	**5.21**	**3.34**	1.94	**2.82**	**4.11**	1.07

Table 1: Quantitative results of our WBFlow and state-of-the-art sRGB-WB methods on three public datasets. The top results are in bold.

We used the remaining two folds (Set1-Test), Set2 [Afifi *et al.*, 2019a] and rendered cube dataset [Afifi *et al.*, 2019a; Banić *et al.*, 2017] for evaluation.

Rendered Multi-camera sRGB Dataset. We collected a multi-camera sRGB dataset to evaluate the multi-camera generalization effect. Specifically, we selected and compiled 184 groups of raw images from the NUS dataset [Cheng *et al.*, 2014]. In each group, the raw images are consistent in the scenes and different in the cameras: Canon1DsMkIII, Canon600D, FujifilmXM1, NikonD5200, OlympusEPL6, PanasonicGX1, SamsungNX2000, and SonyA57. To obtain color-cast sRGB versions of these images, following [Afifi *et al.*, 2019a], we rendered them by Adobe Camera Raw with five common color temperatures (2850 K, 3800 K, 5500 K, 6500 K, and 7500 K) and camera standard photo finishing. We obtain GTs by manually selecting the correct color temperature from the middle gray patches in the color checker of each raw image. The rest of the operations remain unchanged. Our multi-camera sRGB dataset contains 7360 sRGB images with 184 scenes, five color temperatures, and eight cameras.

4.2 Implementation Details

Loss Function. Following [Afifi and Brown, 2020a], we apply l_1 loss to train WBFlow:

$$\underset{\mathcal{F},\mathcal{M}}{\arg\min}\sum_{ct}\sum_{n}\left\|\mathcal{F}^{-1}(\mathcal{M}(\mathcal{F}(\mathbf{I}^{n}_{ct}))) - \mathbf{I}^{n}_{wb}\right\|_{1} . \qquad (8)$$

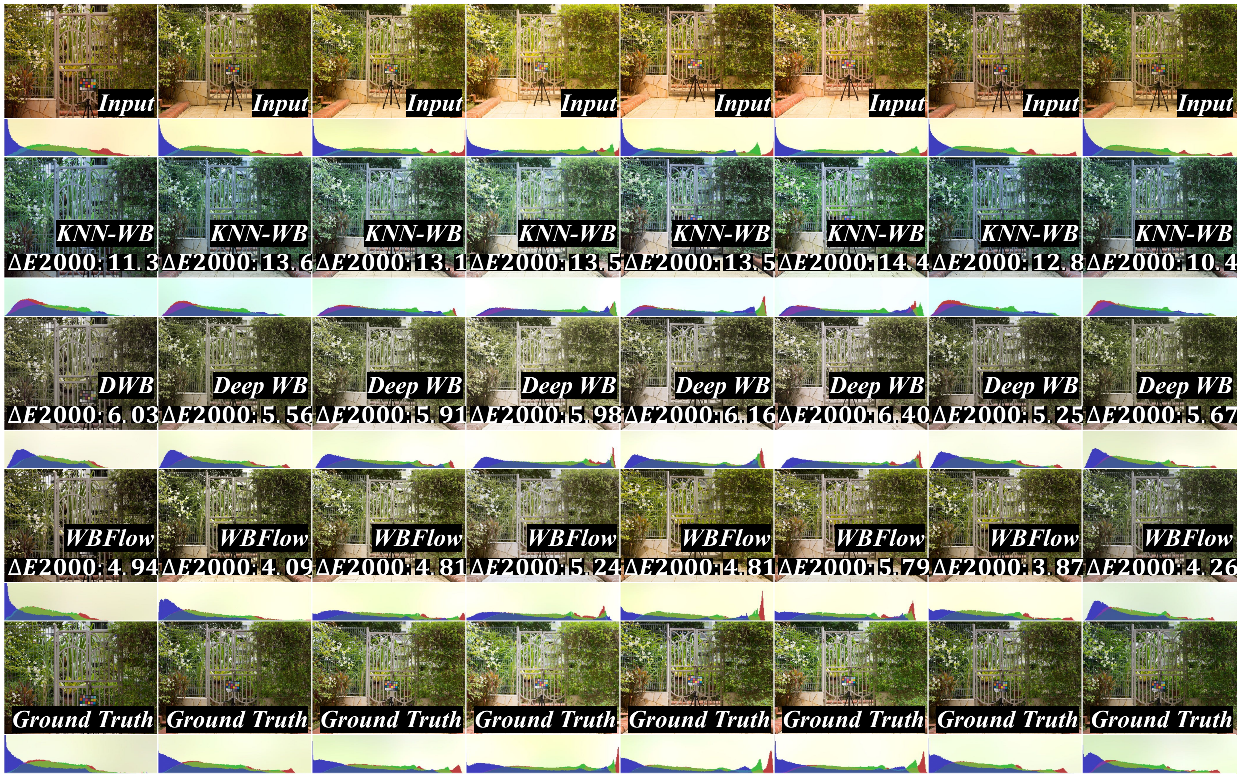
Training and Testing Detail. We implemented WBFlow on Pytorch with CUDA support and used the Adam [Kingma and Ba, 2014] with $\beta_1 = 0.9$ and learning rate 10^{-4} to optimize it. For the experiments with all training images, we trained WBFlow for 340000 iterations with batch size 4. While for few-shot experiments, we trained CT for 15000 iterations. We used color jittering, average blur, geometric rotation, and flipping to augment data. During testing, following [Afifi and Brown, 2020a], we resized all input images to a maximum dimension of 656 pixels and set a color mapping procedure to compute the final white-balanced sRGB images.

Error Metric. We used the same error metrics as existing sRGB-WB methods [Afifi and Brown, 2020a; Afifi *et al.*, 2019a]: mean angle error (MAE) and $\Delta E2000$ [Sharma *et al.*, 2005]. We reported the mean, first quantile (Q1), second quantile (Q2), and third quantile (Q3) for evaluation. A lower error metric denotes a better sRGB-WB performance.

4.3 Multi-camera Generalization Evaluation

Table 1 shows the quantitative results of our WBFlow and state-of-the-art sRGB-WB methods on Set1-Test (trained cameras), Set2 (untrained cameras), and rendered cube dataset (untrained cameras). The results of MixedWB on Set1-Test and Set2 are computed by its code, and others are collected from [Afifi *et al.*, 2021] and [Afifi and Brown, 2020a].

Quantitate Comparison. From Table 1, due to the irreversible and biased mapping, the exemplar-based methods, KNN-WB and MixedWB, generalized badly to untrained cameras. For example, their mean values of $\Delta E2000$ on Set2 and rendered cube dataset are 56.42% and 36.97% higher than those in Set1-Test. Deep WB alleviated this by modeling the ideal WB process by U-Net. However, since U-Net is irreversible, Deep WB performed inferiorly on trained cameras. That is, although Deep WB reduced the performance of KNN-WB in the mean of $\Delta E2000$ about 12.50% and 19.19% in Set2 and rendered cube dataset, its mean of $\Delta E2000$ is 5.31% worse than KNN-WB in Set1-test. In contrast, in Set1-test, our WBFlow outperformed the KNN-WB (ranking second) and the Deep WB (most related method) by about 12.57% and 16.98% in the mean of $\Delta E2000$, respectively. Simultaneously, in Set2 and rendered cube dataset, our WBFlow still outperformed all methods in most metrics, *e.g.*, it greatly improved the accuracy of Deep WB in the mean of $\Delta E2000$ and MAE about 5.31% and 6.40%. Similar results



(a) Canon1DsMkIII (b) Canon600D (c) FujifilmXM1 (d) NikonD5200 (e) OlympusEPL6 (f) PanasonicGX1 (g) SamsungNX200 (h) SonyA57

Figure 5: Qualitative comparison (sRGB images and their color histograms) for multi-camera generalization.

Methods	OlympusEPL6 (△E2000↓)				PanasonicGX1 (△E2000↓)				SamsungNX2000 (△E2000↓)				SonyA57 (△E2000↓)			
	Mean	Q1	Q2	Q3	Mean	Q1	Q2	Q3	Mean	Q1	Q2	Q3	Mean	Q1	Q2	Q3
Deep WB	7.12	5.43	6.94	8.46	6.69	5.50	6.48	7.69	6.87	5.49	6.71	7.76	6.53	4.99	6.28	7.87
WBFlow K=0	6.19	4.39	5.73	7.72	5.72	4.23	5.55	6.65	6.04	4.30	5.30	7.03	5.51	3.82	5.38	6.83
WBFlow K=5	6.13	4.33	5.68	7.75	5.65	4.12	5.24	6.90	6.02	4.23	5.26	7.37	5.50	3.79	5.12	6.76
WBFlow K=10	6.00	4.37	5.37	7.18	5.62	4.10	5.21	6.93	6.00	4.23	5.26	7.34	5.48	3.72	5.10	6.81
WBFlow K=20	5.97	4.25	5.39	7.36	5.61	4.08	5.08	6.92	5.95	4.13	5.16	6.97	5.45	3.70	5.07	6.62

Table 2: Few shot evaluation of WBFlow on rendered multi-camera sRGB dataset (Deep WB:[Afifi and Brown, 2020a]).

appear in Set2. These results verify the effectiveness of our WBFlow in multi-camera generalization.

Inference Time Comparison. From Table 1, our WBFlow achieves superior WB performance on all datasets with almost the same inference time as Deep WB (the most related method). Compared with KNN-WB with the fastest inference time, our WBFlow outperforms it by about 12.57%, 17.14% and 24.65% for the mean of $\Delta E2000$ in three datasets.

Qualitative Comparison. To qualitatively compare the multi-camera generalization, we show the generated white-balanced sRGB images and their color histograms of KNN-WB, Deep WB, and WBFlow on eight cameras in Figure 5. From it, WBFlow achieves the most similar colors as GTs on all cameras, consistent with the quantitate comparison.

4.4 Few Shot Evaluation

To validate few-shot capacity, we compared the performances of WBFlow with the most related method, Deep WB[Afifi and Brown, 2020a], on four untrained cameras. We retrained Deep WB and WBFlow by randomly selecting 5000 sRGB images from the first fold of Set1 and used four untrained cameras in the last 84 groups of our rendered multi-camera sRGB dataset for evaluation: OlympusEPL6, PanasonicGX1, SamsungNX2000, and SonyA57. Training details are followed by [Afifi and Brown, 2020a]. For few-shot experiments, we randomly selected K images (0, 5, 10, 20) for untrained cameras from the first 100 groups of our rendered multi-camera sRGB dataset to train the parameters of CT, which are 5.82% of our WBFlow's. To avoid randomness and disturbance, we repeated experiments 1000 times with randomly selected images and then computed the $\Delta E2000$ average in Table 2.

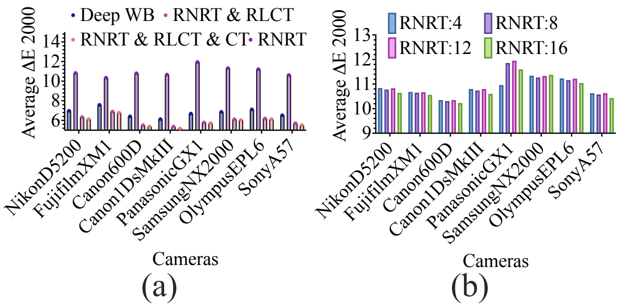


Figure 6: Ablation comparisons of WBFlow for sRGB images from eight cameras (fixed scenes and fixed color temperatures): (a) different variants of WBFlow and (b) different flow numbers of RNRT.

From Table 2, WBFlow significantly outperforms Deep WB even with no shots (K=0) due to the reversible structure. For example, the average $\Delta E2000$ values of WBFlow are smaller than these of Deep WB by about 17.44%, 23.09%, 21.68%, and 23.45% in Q2 on four untrained cameras. This superiority was enhanced when we updated CT with K=5, 10, 20 to enhance the camera specificity. Especially, the few-shot learning effectively improves Q3 of average $\Delta E2000$ on four untrained cameras. This indicates that our WBFlow with few-shot learning is superior in handling untrained cameras.

4.5 Ablation Analysis

To verify the effectiveness of each part in WBFlow, we conducted an ablation analysis on our rendered multi-camera sRGB dataset. Since Deep WB is the most related method, we report the average $\Delta E2000$ of Deep WB and three variants of our WBFlow in Figure 6(a). Training and testing settings are the same as few-shot experiments. Further, we compared the influences of flow numbers for RNRT in Figure 6(b) and the camera specificity verification for CT in Figure 7.

Variants Comparison. From Figure 6(a), the average $\Delta E2000$ values of RNRT on eight cameras are much worse than Deep WB, which shows that only modeling color renderings hardly improves the multi-cameras generalization. This phenomenon is significantly mitigated when we integrate RNRT and RLCT. Specifically, compared with only RNRT, RNRT&RLCT work together to reduce the average $\Delta E2000$ values by about 50% on eight cameras. RNRT&RLCT also considerably outperforms Deep WB by 13% on eight cameras. This is because both RNRT and RLCT are reversible to effectively model the ideal WB process. When we add CT, the multi-camera generalization is further improved, as in the smallest average $\Delta E2000$ of RNRT&RLCT&CT on eight cameras.

Flows Numbers Influence for RNRT. To explore the optimal flow numbers of RNRT, in Figure 6(b), we compared the average $\Delta E2000$ of RNRT with 4, 8, 12, and 16 flows on eight cameras. It can be seen that the complexity of the RNRT becomes greater as the number of streams increases to better simulate ISP nonlinear renderings. The average $\Delta E2000$ of RNRT is minimized at a number of 16, which means that 16 is the optimal flow number.

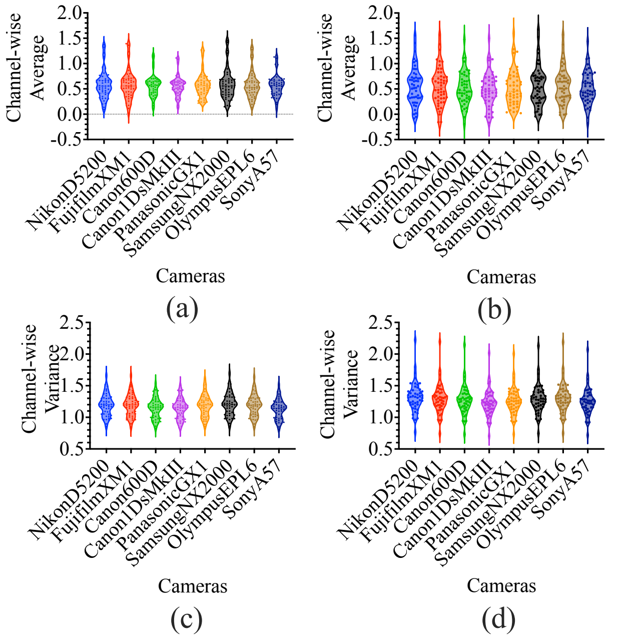


Figure 7: Channel average and variance values distributions of white-balanced pseudo-raw features for sRGB images from eight cameras when WBFlow w/ and w/o CT: (a) and (c) are the distributions w/o CT, (b) and (d) are the distributions w/ CT.

Camera Specificity Verification. Since channel average and variance can represent the color information of pseudo-raw features [Stricker and Orengo, 1995], we compute them for white-balanced pseudo-raw features without and with CT on eight cameras to verify the camera specificity in Figure 7. With CT, the difference between mean/variance for all white balance pseudo-raw feature channels becomes significantly larger on eight cameras. This verifies that CT can significantly enhance the camera specificity of pseudo-raw features, thus improving multi-camera generalization.

5 Conclusion

In this paper, we propose an sRGB-WB network named WBFlow, which performs superior white balance for sRGB images and generalizes well to multiple cameras. Unlike existing irreversible sRGB-WB methods that fail to model the ideal WB process, WBFlow successfully models this process through reversible RNRT and RLCT. Furthermore, we generalize WBFlow to multiple cameras by enhancing the camera characteristic of pseudo-raw features via few-shot learning. Extensive experiments indicate the superiority of our WBFlow in multi-camera generalization and few-shot sRGB-WB tasks. Ablation analysis shows the effectiveness of each part of WBFlow and their optimal combinations.

Acknowledgements

This work was supported by the national key R & D program intergovernmental international science and technology innovation cooperation project (2021YFE0101600).

References

[Afifi and Brown, 2019] Mahmoud Afifi and Michael S Brown. What else can fool deep learning? addressing color constancy errors on deep neural network performance. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 243–252, 2019.

[Afifi and Brown, 2020a] Mahmoud Afifi and Michael S Brown. Deep white-balance editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1397–1406, 2020.

[Afifi and Brown, 2020b] Mahmoud Afifi and Michael S Brown. Interactive white balancing for camera-rendered images. *arXiv preprint arXiv:2009.12632*, 2020.

[Afifi *et al.*, 2019a] Mahmoud Afifi, Brian Price, Scott Cohen, and Michael S Brown. When color constancy goes wrong: Correcting improperly white-balanced images. In *CVPR*, 2019.

[Afifi *et al.*, 2019b] 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.

[Afifi *et al.*, 2020] Mahmoud Afifi, Abdelrahman Abdelhamed, Abdullah Abuolaim, Abhijith Punnappurath, and Michael S Brown. Cie xyz net: Unprocessing images for low-level computer vision tasks. *arXiv preprint arXiv:2006.12709*, 2020.

[Afifi *et al.*, 2021] Mahmoud Afifi, Marcus A Brubaker, and Michael S Brown. Auto white-balance correction for mixed-illuminant scenes. *arXiv preprint arXiv:2109.08750*, 2021.

[Afifi, 2021] Mahmoud Afifi. Image color correction, enhancement, and editing. *arXiv preprint arXiv:2107.13117*, 2021.

[An *et al.*, 2021] Jie An, Siyu Huang, Yibing Song, Dejing Dou, Wei Liu, and Jiebo Luo. Artflow: Unbiased image style transfer via reversible neural flows. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 862–871, 2021.

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

[Chakrabarti *et al.*, 2014] Ayan Chakrabarti, Ying Xiong, Baochen Sun, Trevor Darrell, Daniel Scharstein, Todd Zickler, and Kate Saenko. Modeling radiometric uncertainty for vision with tone-mapped color images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(11):2185–2198, 2014.

[Cheng *et al.*, 2014] 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*, 31(5):1049–1058, 2014.

[Foster, 2011] David H Foster. Color constancy. *Vision research*, 51(7):674–700, 2011.

[Gao *et al.*, 2017] Shao-Bing Gao, Ming Zhang, Chao-Yi Li, and Yong-Jie Li. Improving color constancy by discounting the variation of camera spectral sensitivity. *JOSA A*, 34(8):1448–1462, 2017.

[Gijsenij and Gevers, 2007] Arjan Gijsenij and Theo Gevers. Color constancy using natural image statistics. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007.

[Hong *et al.*, 2001] Guowei Hong, M Ronnier Luo, and Peter A Rhodes. A study of digital camera colorimetric characterization based on polynomial modeling. *Color Research & Application: Endorsed by Inter-Society Color Council, The Colour Group (Great Britain), Canadian Society for Color, Color Science Association of Japan, Dutch Society for the Study of Color, The Swedish Colour Centre Foundation, Colour Society of Australia, Centre Français de la Couleur*, 26(1):76–84, 2001.

[Hu *et al.*, 2017] 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*, pages 4085–4094, 2017.

[Huang and Belongie, 2017] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE international conference on computer vision*, pages 1501–1510, 2017.

[Huang *et al.*, 2020] Xinwei Huang, Bing Li, Shuai Li, Wenjuan Li, Weihua Xiong, Xuanwu Yin, Weiming Hu, and Hong Qin. Multi-cue semi-supervised color constancy with limited training samples. *IEEE Transactions on Image Processing*, 29:7875–7888, 2020.

[Kingma and Ba, 2014] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[Kingma and Dhariwal, 2018] Durk P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. *Advances in neural information processing systems*, 31, 2018.

[Lo *et al.*, 2021] 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. *arXiv preprint arXiv:2106.04989*, 2021.

[Ramanath *et al.*, 2005] Rajeev Ramanath, Wesley E Snyder, Youngjun Yoo, and Mark S Drew. Color image processing pipeline. *IEEE Signal Processing Magazine*, 22(1):34–43, 2005.

[Ronneberger *et al.*, 2015] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[Sharma *et al.*, 2005] Gaurav Sharma, Wencheng Wu, and Edul N Dalal. The CIEDE2000 color-difference formula: Implementation notes, supplementary test data, and mathematical observations. *Color Research & Application*, 30(1):21–30, 2005.

[Song *et al.*, 2021] Zeyu Song, Dongliang Chang, Zhanyu Ma, Xiaoxu Li, and Zheng-Hua Tan. Cc-loss: Channel correlation loss for image classification. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 7601–7608. IEEE, 2021.

[Stricker and Orengo, 1995] Markus Andreas Stricker and Markus Orengo. Similarity of color images. In *Storage and retrieval for image and video databases III*, volume 2420, pages 381–392. SPiE, 1995.

[Tang *et al.*, 2022] Yuxiang Tang, Xuejing Kang, Chunxiao Li, Zhaowen Lin, and Anlong Ming. Transfer learning for color constancy via statistic perspective. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 2361–2369, 2022.

[Xu *et al.*, 2021] Qinwei Xu, Ruipeng Zhang, Ya Zhang, Yanfeng Wang, and Qi Tian. A fourier-based framework for domain generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14383–14392, 2021.

[Zhang *et al.*, 2022] Zhifeng Zhang, Xuejing Kang, and Anlong Ming. Domain adversarial learning for color constancy. In Lud De Raedt, editor, *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, pages 1693–1699. International Joint Conferences on Artificial Intelligence Organization, 7 2022. Main Track.