SWBNet: A Stable White Balance Network for sRGB Images

Chunxiao Li, Xuejing Kang, Zhifeng Zhang, Anlong Ming*

School of Computer Science (National Pilot Software Engineering School), Beijing University of Posts and Telecommunications
{chunxiaol,kangxuejing,zhangzhifeng,mal}@bupt.edu.cn

Abstract

The white balance methods for sRGB images (sRGB-WB) aim to directly remove their color temperature shifts. Despite achieving promising white balance (WB) performance, the existing methods suffer from WB instability, *i.e.*, their results are inconsistent for images with different color temperatures. We propose a stable white balance network (SWBNet) to alleviate this problem. It learns the color temperature-insensitive features to generate white-balanced images, resulting in consistent WB results. Specifically, the color temperature-insensitive features are learned by implicitly suppressing low-frequency information sensitive to color temperatures. Then, a color temperature contrastive loss is introduced to facilitate the most information shared among features of the same scene and different color temperatures. This way, features from the same scene are more insensitive to color temperatures regardless of the inputs. We also present a color temperature sensitivity-oriented transformer that globally perceives multiple color temperature shifts within an image and corrects them by different weights. It helps to improve the accuracy of stabilized SWBNet, especially for multi-illumination sRGB images. Experiments indicate that our SWBNet achieves stable and remarkable WB performance.

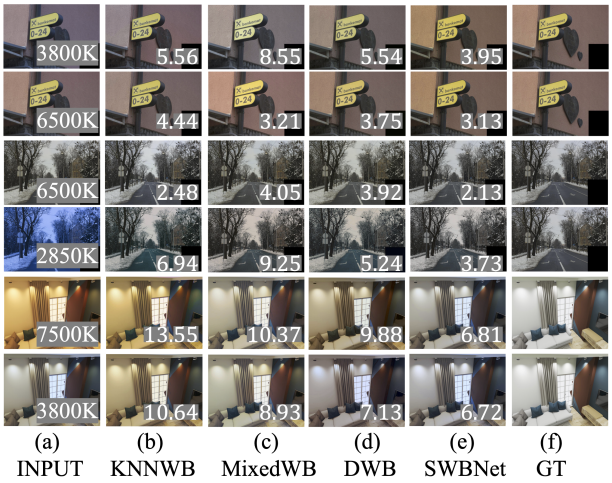


Figure 1: WB instability visualization. The closer and lower $\Delta E2000$ values for the inputs with different color temperatures and same scene mean better WB stability and accuracy.

Introduction

White balance for sRGB images (sRGB-WB) aims at removing the color temperature shifts caused by the improper or personalized white balance (WB) settings in image signal processing (ISP) (Afifi et al. 2019a). It is a burgeoning research direction and has a great impact on some computer vision tasks, such as semantic segmentation and image classification (Afifi and Brown 2019).

Since 2019, several sRGB-WB methods have been proposed, which can be divided into exemplar-based methods and deep neural network (DNN)-based methods. Exemplar-based methods, such as KNN-WB (Afifi et al. 2019a) and MixedWB (Afifi, Brubaker, and Brown 2022), compute nonlinear mappings for different color temperatures and then correct the input images according to them. DNN-based method, *i.e.*, DWB (Afifi and Brown 2020a), builds a unified mapping for different color temperatures with the DNN architecture trained in an end-to-end manner. Although achieving promising WB accuracy, both categories of methods have inconsistent results for the images with different color temperatures. That is, they suffer from WB instability, which limits the performances when applied to downstream tasks, such as TV broadcasting and image editing (Vazquez-Corral and Bertalmío 2014). Fig.1 shows some examples. Three representative sRGB-WB methods (KNN-WB, MixedWB, DWB) are used to correct the sRGB images rendered with four typical color temperatures. Ideally, regardless of the inputs, an effective sRGB-WB method should generate white-balanced images with consistent neutral color temperature. However, for the single-illumination inputs in Fig.1(a) (first four rows), the white-balanced images generated by three methods have inconsistent color temperatures even if the scenes are fixed. This is reflected in the visualizations and varying $\Delta E2000$ values (Sharma, Wu, and Dalal 2005) in Fig.1(b)-(d). Worse yet, for the multi-illumination inputs in Fig.1(a) (last two rows), WB instability is severe, and WB accuracy is greatly degraded.

In this paper, we propose a stable white balance network

*Corresponding Author.

called SWBNet to mitigate WB instability. Our SWBNet learns the color temperature-insensitive features to generate white-balanced sRGB images, thus effectively reducing their color temperature inconsistency. Specifically, we propose a color temperature-insensitive feature (CTIF) extractor based on low-frequency information's color temperature-sensitive property. By suppressing this information, the CTIF extractor can extract the color temperature-insensitive features. Then, these features are encouraged to become more insensitive by the color temperature contrastive (CT-contrastive) loss. It explicitly maximizes the shared information among features with the same scene and different color temperatures. We also propose a color temperature sensitivity-oriented (CTS-oriented) transformer to further improve WB accuracy, especially for multi-illumination sRGB images. Benefiting from the global receptive field, the CTS-oriented transformer can perceive different color temperature shifts within an image and thus corrects them differently. In summary, the contributions are:

(1) We propose a CTIF extractor that gathers color temperature-sensitive information from extracted features and then adaptively suppresses it, thus implicitly learning color temperature-insensitive features.

(2) We propose a CT-contrastive loss to enhance the insensitivity by promoting features with the same scenes and different color temperatures to share the most information.

(3) We present a CTS-oriented transformer to further improve WB accuracy by correcting multiple color temperature shifts differently.

(4) Our SWBNet achieves stable and state-of-the-art WB performance on commonly used datasets.

Related Work

In the WB module of ISP, the classical WB methods (Raw-WB) for raw images are used to remove their color shift caused by the scene illumination (Gijsenij and Gevers 2007; Foster 2011; Hu, Wang, and Lin 2017; Li, Fu, and Heidrich 2021; Kim et al. 2021; Zhao et al. 2022). Most digital cameras offer an option to adjust these methods according to the desired color temperature. However, once the WB is personalized or incorrectly set, the final sRGB images still have color temperature shifts. Further, due to non-linear renderings in ISP, the shifts within an sRGB image is inconsistent (Afifi et al. 2019a). Raw-WB methods can not correct them, which motivated the development of sRGB-WB methods.

In 2019, (Afifi et al. 2019a) proposed the first exemplar-based sRGB-WB method called KNN-WB. It modeled the nonlinear mapping as a polynomial kernel function (Hong, Luo, and Rhodes 2001) by extracting the high-order color matrix. Such a color temperature-sensitive feature makes KNN-WB affected by color temperatures. Followed by KNN-WB, some exemplar-based methods were also proposed. For example, CTT(Afifi et al. 2019b) embedded mappings computed by KNN-WB in the JPEG annotations. When the color temperature shift appeared in an sRGB image, CTT fetched these mappings to correct it. Interactive-WB(Afifi and Brown 2020b) allowed users to select a reference point in an sRGB image and then use KNN-WB to correct it or reset its color temperature. Recently, MixedWB

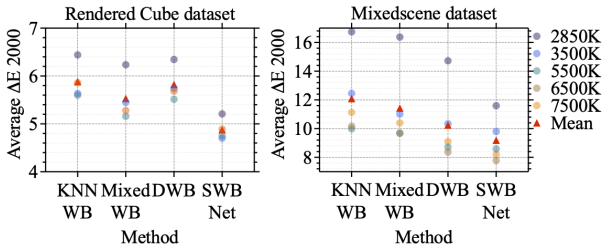


Figure 2: The average $\Delta E2000$ values of KNN-WB, MixedWB, DWB and our SWBNet in Rendered Cube and Mixedscene datasets (different color temperatures and fixed scenes). The closer and lower values refer to better WB stability and accuracy. The values are in Table.2.

(Afifi, Brubaker, and Brown 2022) applied the mappings in KNN-WB to generate images with preset color temperatures. Then, it trained a Gridnet to compute the weighting maps for the preset color temperatures to correct the multi-illumination sRGB images. Although achieving the goal of correcting sRGB images, the above methods suffered from WB instability because they are affected by color temperatures (Fig.1(b),(c)). In addition, they generalized poorly to unseen color temperatures due to the limited exemplars.

With the development of deep learning, the DNN-based method, DWB (Afifi and Brown 2020a), was proposed to build a unified mapping by the U-net (Ronneberger, Fischer, and Brox 2015). It also allowed users to edit color temperatures in the sRGB images. With l_1-norm constraint, DWB lacked consideration of color temperature influences. Inevitably, the input color temperatures affected the generated white-balanced images, resulting in WB instability.

As mentioned above, existing sRGB-WB methods suffer from WB instability due to the influences of color temperatures. In this paper, we attempt to reduce these influences by learning color temperature-insensitive features. We also improve the accuracy of the stabilized model by correcting multiple color temperature shift regions differently.

Preliminaries and Problem Analysis

Preliminaries

Rendered with color temperature ct_i, the sRGB image $\mathbf{I}_{ct_i}$ is formulated as (Karaimer and Brown 2016):

$$\mathbf{I}_{ct_i} = f_{XYZ\to sRGB}(\mathbf{T}\mathbf{D}_{ct_i}\mathbf{I}_{raw}), \tag{1}$$

where $\mathbf{I}_{raw}$ is the raw version of $\mathbf{I}_{ct_i}$, $\mathbf{D}_{ct_i}$ is the diagonal WB matrix with color temperature ct_i, $\mathbf{T}$ is the linear transformation matrix that maps the rendered image from the raw-RGB space to CIE-XYZ space, and $f_{XYZ\to sRGB}(\cdot)$ is the nonlinear function that compounds various rendering operations (Chakrabarti et al. 2014; Kim et al. 2012). From Eq.1, the intuitive way to correct $\mathbf{I}_{ct_i}$ is calibrating $\mathbf{D}_{ct_i}$ in raw-RGB space. However, it won't work because of the nonlinear renderings in ISP (Afifi et al. 2019a).

Nowadays, the sRGB-WB methods are proposed to directly generate the white-balanced sRGB image from input

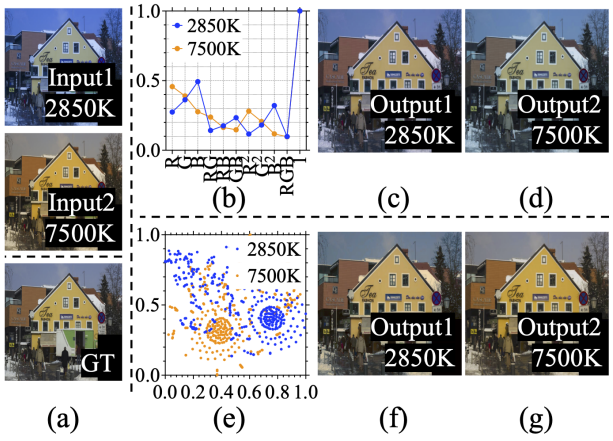


Figure 3: (a) Inputs and Ground truth. (b),(c),(d) Feature distributions and outputs of KNN-WB. (e),(f),(g) Feature distributions and outputs of DWB.

sRGB image (Afifi et al. 2019a; Afifi and Brown 2020a):

$$\hat{\mathbf{I}}_{wb_i} = G(F(\mathbf{I}_{ct_i})), \tag{2}$$

where $F(\cdot)$ is the feature extractor that extracts the feature $\mathbf{X}_{ct_i}$ from the input $\mathbf{I}_{ct_i}$, $G(\cdot)$ is the generator that generates the white-balanced sRGB image $\hat{\mathbf{I}}_{wb_i}$ from $\mathbf{X}_{ct_i}$.

WB Instability

Existing sRGB-WB methods suffer from WB instability. In Fig.1, representative methods (KNN-WB, MixedWB, DWB) generate white-balanced sRGB images with inconsistent color temperatures even if the input scenes are fixed.

To discuss this quantitatively, Fig.2 presents the average $\Delta E2000$ values of representative methods for the sRGB images with different color temperatures (fixed scenes) in Rendered Cube (Afifi et al. 2019a) and Mixedscene datasets (Afifi, Brubaker, and Brown 2022). Ideally, an effective method should obtain close and low average $\Delta E2000$ values regardless of the input color temperatures. However, in fact, these values vary considerably at different color temperatures, as in Fig.2. Further, comparing (a) and (b) in Fig.2, the average $\Delta E2000$ values are larger in the Mixedscene dataset. Such phenomena indicate that the existing methods suffer from WB instability and perform worse for the multi-illumination sRGB images.

Reasons for WB Instability

Exemplar-based method, *e.g.* KNN-WB, models Eq.2 as the polynomial kernel function by minimizing the color difference between the exemplar and Ground truth (GT):

$$\arg\min_{\hat{\mathbf{G}}_{ct_i}} \left\| \mathbf{I}^{e}_{wb_i} - \hat{\mathbf{G}}_{ct_i} F_e(\mathbf{I}^{e}_{ct_i}) \right\|_F, \tag{3}$$

where $\mathbf{I}^{e}_{ct_i}$ is the exemplar image, $\mathbf{I}^{e}_{wb_i}$ is its GT, the fixed $F_e(\cdot)$ extracts the high-order color matrix $\mathbf{X}^{color}_{ct_i}$, the learned mapping $\hat{\mathbf{G}}_{ct_i}$ accepts it to generate white-balanced images. Thus, $\hat{\mathbf{G}}_{ct_i}$ is affected by color temperature ct_i due to the color temperature sensitivity of $\mathbf{X}^{color}_{ct_i}$. WB instability will occur if using these mappings for white balancing.

DNN-based method, *i.e.* DWB, models Eq.2 as U-net by minimizing the pixel differences between the images with different color temperatures and their GTs:

$$\arg\min_{\theta_g,\theta_f} \sum_{ct_i} \sum_{s_j} \left\| \mathbf{I}^{s_j}_{wb_i} - G_{\theta_g}(F_{\theta_f}(\mathbf{I}^{s_j}_{ct_i})) \right\|_1, \tag{4}$$

where θ_f and θ_g are the parameters of encoder $F_{\theta_f}(\cdot)$ and decoder $G_{\theta_g}(\cdot)$, s_j is the j^{th} scene. Eq.4 aims to reconstruct image content perfectly and lacks consideration of different color temperature influences. Such constraint makes the trained $G_{\theta_g}(F_{\theta_f}(\cdot))$ affected by color temperatures. As a result, it would have inconsistent WB accuracy when correcting different color temperature inputs.

Next, we verify the above analysis. In Fig.3(a), the input sRGB images differ only in color temperatures and have the same GT. Affected by color temperatures, the existing sRGB-WB methods (*e.g.* KNN-WB, DWB) learn the color temperature-sensitive features for white-balanced sRGB image generation. As in Fig.3(b) and (e), there are clear non-overlaps among these features. Inevitably, in Fig.3(c), (d), (f), and (e), the generated white-balanced sRGB images have inconsistent color temperatures. Experiments in Rendered Cube and Mixedscene datasets accord with the above phenomena (Supplementary materials). Thus, we can conclude that existing sRGB-WB methods are affected by color temperatures and thus suffer from WB instability.

Stable White Balance Network

In this section, we introduce the SWBNet, which aims to alleviate WB instability. The whole framework is shown in Fig.4. We first introduce the CTIF extractor and CT-contrastive loss (Fig.4A). They collaborate to promote the SWBNet to learn the color temperature-insensitive features so as to alleviate WB instability. Then, we introduce the CTS-oriented transformer (Fig.4B). It helps to improve the performance of stabilized SWBNet by correcting multiple color temperature shifts within an sRGB image differently.

Color Temperature-insensitive Feature Extractor

We propose the CTIF extractor to learn the color temperature-insensitive features. As shown in Fig.4A, it consists of a color temperature-sensitive frequency separation (CTSFS) block and a color temperature sensitivity reduction (CTSR) block in each downsample (DS) module. The CTSFS block gathers color temperature-sensitive information from features. Then, the CTSR block suppresses this information, which makes features insensitive to color temperature. In the following, we introduce two blocks in detail.

Color Temperature-sensitive Frequency Separation Block Inspired by (El Helou, Zhou, and Süsstrunk 2020; Xu et al. 2020), as in Fig.5, we observe that the coefficients sensitive to color temperatures are concentrated in the lowest continuous indexes ($t \leq 4$) in the discrete cosine transform (DCT) domain. Based on this, we calculate the

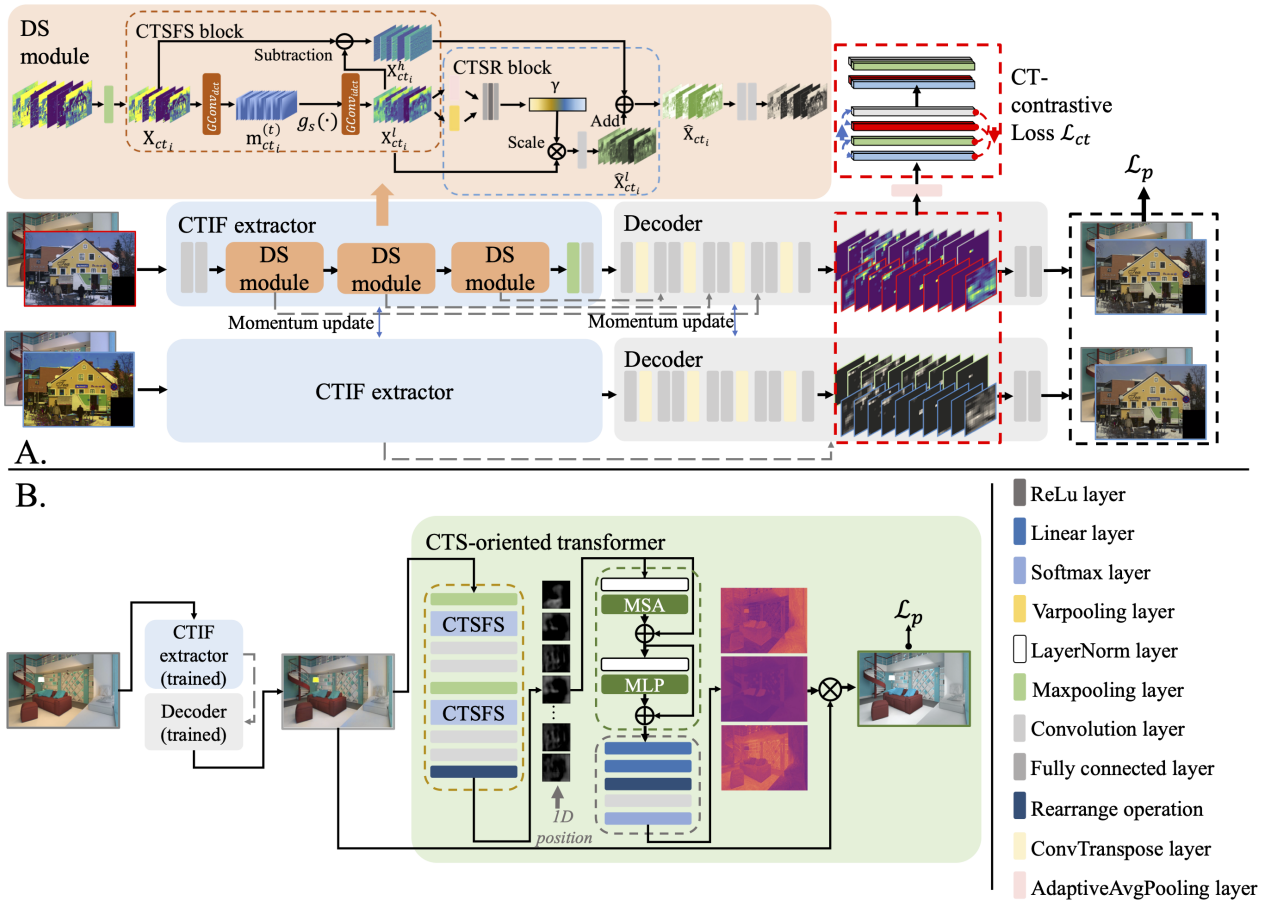


Figure 4: The framework of our SWBNet: A. The CTIF extractor and CT-contrastive loss work together to learn the color temperature-insensitive features for achieving stable WB performance. B. The CTS-oriented transformer corrects multiple color temperature shifts differently to improve WB accuracy, especially for the multi-illumination sRGB images.

color temperature-sensitive DCT coefficients for the c^{th} channel of the deep feature $\mathbf{X}_{ct_i}$ as:

$$\mathbf{m}_{ct_i}^{(c,t)} = \mathcal{T}_t(\mathbf{X}_{ct_i}^{(c)}) = \mathbf{w}_{t_1,t_2} * \mathbf{X}_{ct_i}^{(c)}, \quad (5)$$

where $t_1, t_2 = zig^{-1}(t), t \in [0, T]$ is the index range (we set $T = 4$ according to Fig.5), $\mathbf{w}_{t_1,t_2}$ is the DCT basis in index t, $zig^{-1}(\cdot)$ is the inverse zig-zag function (Guo, Mousavi, and Monga 2019), $*$ is the convolution operation. The color temperature-sensitive information $\mathbf{X}_{ct_i}^{(c,l)}$ can be computed by inverse DCT (IDCT):

$$\mathbf{X}_{ct_i}^{l,(c)} = \mathcal{T}_t^{-1}(\{\mathbf{m}_{ct_i}^{(c,t)}\}) = \sum_0^T \mathbf{w}_{t_1,t_2} * g_s(\mathbf{m}_{ct_i}^{(c,t)}), \quad (6)$$

where $g_s(\cdot)$ is the S zero-padding function (Guo, Mousavi, and Monga 2019). In Fig.5, the remaining high-frequency information is insensitive to color temperatures, which can be calculated as $\mathbf{X}_{ct_i}^{h,(c)} = \mathbf{X}_{ct_i}^{(c)} - \mathbf{X}_{ct_i}^{l,(c)}$.

The attributions and structure of the CTSFS block are in Table.1 and Fig.4A (orange block). Due to the linear property of DCT and IDCT, we make the CTSFS block participate in the training of the CTIF extractor by modeling Eq.5 and Eq.6 as group convolutions. This way, when applying the CTSFS block to the features with different color temperatures (Fig.6(a)), there is a clear difference in the gathered color temperature-sensitive parts (Fig.6(b)), while the color temperature-insensitive parts are similar (Fig.6(c)). Further, such ability provides a centralized object for color temperature sensitivity reduction in the CTSR block.

Color Temperature Sensitivity Reduction Block Since the color temperature sensitivity is concentrated in the information gathered by our CTSFS bock, the whole feature will be color temperature-insensitive if suppressing this portion. Our CTSR block achieves the goal by implicitly learning the color temperature shifts from its intensity description:

$$\begin{aligned}\hat{\mathbf{X}}_{ct_i}^{l} &= Conv(\mathbf{X}_{ct_i}^{l} \cdot \gamma) \\ &= Conv(\mathbf{X}_{ct_i}^{l} \cdot (Fc(Avg(\mathbf{X}_{ct_i}^{l})) + Fc(Var(\mathbf{X}_{ct_i}^{l})))),\end{aligned} \quad (7)$$

where $\cdot$ is the dot product, $Conv(\cdot)$ is the convolution layer. Based on the fact that the color temperatures mainly affect feature intensities (Foster 2011) (*e.g.* Fig.6(b)), the comprehensive descriptions of these influences can be obtained

Layer	$GConv_{dct}(Eq.5)$	$GConv_{idct}(Eq.6)$
Groups	1	T
Stride	8×8	1×1
DCT kernel size	$1 \times 1 \times 8 \times 8$	$T \times 1 \times 8 \times 8$
DCT kernel range	$[0, T]$	$[0, T]$

Table 1: The attributions of our CTSFS block.

by computing the average and variance in each channel by the adaptive average pooling $Avg(\cdot)$ and variance computation $Var(\cdot)$. Then, we apply the fully connected layers $Fc(\cdot)$ to learn the color temperature shift parameter γ from these descriptions. Color temperature-insensitive information $\hat{\mathbf{X}}^l_{ct_i}$ can be obtained by multiplying $\mathbf{X}^l_{ct_i}$ and γ. This effectively suppresses the color temperature sensitivity (Fig.6(d)). Thus, the deep feature $\hat{\mathbf{X}}_{ct_i}$ that reconstructed by $\hat{\mathbf{X}}^l_{ct_i}$ and $\mathbf{X}^h_{ct_i}$ is color temperature-insensitive (Fig.6(f)).

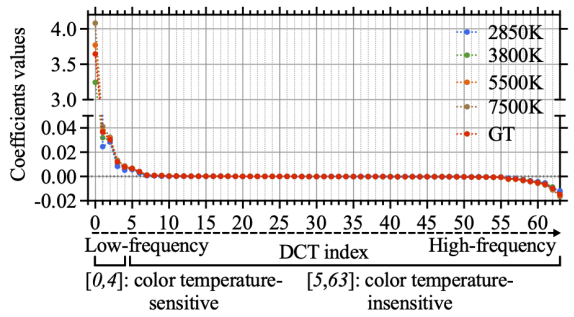


Figure 5: DCT coefficient values of the sRGB images for different color temperatures in Rendered Cube and Mixed-scene datasets: they differ in the continuous low-frequency indexes but tend to be the same in the high-frequency ones.

Color Temperature Contrastive Loss

Since features from the same scene have identical GT, they become increasingly insensitive to color temperatures as sharing more information. This can further promote the model to learn color temperature-insensitive features. Inspired by (He et al. 2020), we introduce an explicit constraint called CT-contrastive loss to facilitate information sharing:

$$\mathcal{L}_{ct} = -log\frac{exp(\left\| Avg(\mathbf{X}_{ct_i}), Avg(\mathbf{X}^{p+}_{ct_j}) \right\|_2)/\tau}{\sum_q exp(\left\| Avg(\mathbf{X}_{ct_i}), Avg(\mathbf{X}^{q-}_{ct_t}) \right\|_2)/\tau + \epsilon}, \quad (8)$$

where $Avg(\cdot)$ is the adaptive average pooling that extracts the global color temperature influence representation in each channel, $\|\cdot\|_2$ is the l_2 distance computation, τ is a temperature hyper-parameter ($\tau = 0.7$) (Peng et al. 2019), ϵ is a constant. From Eq.8, given a target feature $\mathbf{X}_{ct_i}$, reducing the distance between it and the positive feature $\mathbf{X}^{p+}_{ct_i}$ (differ from $\mathbf{X}_{ct_i}$ only in color temperatures) encourages them to share the same information. Simultaneously, increasing the distance between $\mathbf{X}_{ct_i}$ and the negative feature $\mathbf{X}^{q-}_{ct_i}$ (differ from $\mathbf{X}_{ct_i}$ in scenes) preserves as much shared information as possible. Consequently, with our CT-contrastive loss,

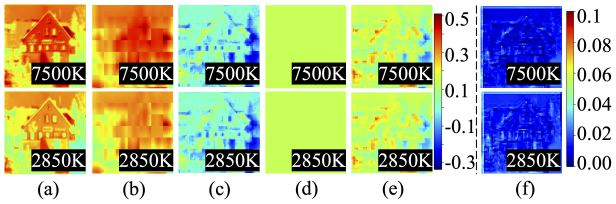


Figure 6: (a) Feature channels (inputs are in Fig.3(a)). (b)-(c) Color temperature-sensitive and insensitive information extracted by the CTSFS block. (d) Color temperature-insensitive information optimized by the CTSR block. (e) Reconstructed color temperature-insensitive feature channels by the CTSFS and CTSR block. (f) Color temperature-insensitive feature channels learned by CT-contrastive loss.

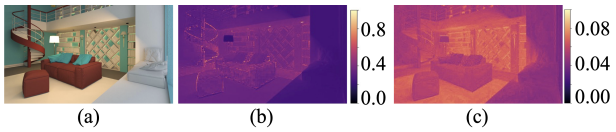


Figure 7: (a) Multi-illumination input. (b) Heatmap of multiple color temperature shifts. (c) Mask formed by our CTS-oriented transformer.

the features from the same scene (Fig.6(a)) become color temperature-insensitive (Fig.6(f)).

In Fig.4A, we use a momentum-updated student model to provide the positive features (He et al. 2020). Also, since the multi-level features extracted by the CTIF extractor are critical to the image generation (Ronneberger, Fischer, and Brox 2015), we apply the CT-contrastive loss on them after the skip connections. This way, the CTIF extractor and CT-contrastive loss work together to learn color temperature-insensitive features via implicit color temperature sensitivity suppression and explicit information-sharing promotion.

Color Temperature Sensitivity-oriented Transformer

Due to the nonlinear renderings in ISP (Eq.1), the color temperature shifts within an sRGB image are inconsistent. Existing methods hardly capture these shifts globally because of the convolution's limited receptive field (Afifi and Brown 2020a; Afifi, Brubaker, and Brown 2022) or the mapping's limited color temperature representation (Afifi et al. 2019a). This limits WB accuracy except for instability, especially for the multi-illumination sRGB images (Fig.1, Fig.2(b)). The transformer can effectively capture the multiple color temperature shifts based on its global receptive field (Xie et al. 2021; Hermann, Chen, and Kornblith 2020). Benefiting from it, we propose a CTS-oriented transformer, which adopts the CTSFS block to extract color temperature-sensitive features for patch embedding. This provides dense and direct clues for exploring the global color temperature shifts.

As shown in Fig.4B, there are three parts in the framework: CTIF extractor, decoder, and CTS-oriented transformer. The input is first passed to the first two for pre-

Methods	Rendered Cube dataset ($\Delta E2000$)					Mixedscene dataset ($\Delta E2000$)					
	$2850K$	$3800K$	$5500K$	$7500K$	Mean	$2850K$	$3800K$	$5500K$	$6500K$	$7500K$	Mean
KNN-WB	6.44	5.64	5.60	5.86	5.88	16.84	12.56	10.10	10.17	11.13	12.10
DWB	6.35	5.74	5.52	5.68	5.82	16.38	11.02	9.69	9.68	10.40	11.43
MixedWB	6.24	5.45	5.16	5.28	5.53	14.73	10.35	8.71	8.37	9.10	10.25
SWBNet	**5.21**	**4.70**	**4.75**	**4.89**	**4.88**	**12.60**	**9.40**	**8.00**	**7.73**	**8.21**	**9.19**

Table 2: Quantitative comparison for WB stability on the Rendered Cube and Mixedscene datasets. The top results are in bold. We discarded the sRGB images rendered by As Shot (AS) and Auto AWB (AU) due to the unfixed color temperatures.

(a) Input (b) KNN-WB (c) MixedWB (d) DWB (e) Our SWBNet (f) Ground truth

Figure 8: Qualitative comparison for WB stability. The $\Delta E2000$ value is marked in the bottom right of the image.

liminary correction. Then, it is forwarded to the patch embedding module, which adopts two CTSFS blocks to obtain the color temperature-sensitive patches and then projects them as embeddings. To capture correlations, these embeddings are added with $1D$ position encoding and then fed to the transformer encoder with a multi-headed self-attention (MSA) computing inter-similarities. A linear decoder is further applied to learn the global color temperature shifts by linear and convolution layers. In Fig.7(b) and (c), the learned mask assigns higher correction weights to the areas with larger color temperature shifts. Thus, correcting the inputs with these weights can achieve superior WB accuracy, especially for the multi-illumination sRGB images.

Experiments

Experiments Settings

Datasets Following (Afifi and Brown 2020a), we randomly selected 12000 sRGB images from Rendered WB dataset (Afifi et al. 2019a) for training and used Rendered Cube and Mixedscene datasets for evaluation.

Error Metric We used the same error metrics as the recent works (Afifi et al. 2019a; Afifi, Brubaker, and Brown 2022; Afifi and Brown 2020a): mean square error (MSE), mean angle error (MAE), and $\triangle E2000$. We reported the metrics' mean, first quantile (Q1), second quantile (Q2), and third quantile (Q3) for evaluation.

Loss Function We firstly applied the CT-contrastive loss $\mathcal{L}_{ct}$ (Eq.8) and the l_1 loss $\mathcal{L}_p$ (Eq.4) to train the CTIF extractor and decoder (Fig.4A). Secondly, we trained the whole model under the supervision of loss $\mathcal{L}_p$ (Fig.4B).

Implementation Details We implement the SWBNet on Pytorch with CUDA support. To train the CTIF extractor and decoder, we use the Adam (Kingma and Ba 2015) with $\beta_1 = 0.9$. We set the learning rate as 1×10^{-4} and then decay it to 1×10^{-5} after 150 epochs. To train the whole model, we use the AdamW optimizer (Loshchilov and Hutter 2019) with weight-decay 10^{-2}. We set batch size as 64 and train all modules for 200 epochs in two phases. Following (Afifi and Brown 2020a), we randomly select four 128×128 patches from each image and their corresponding GTs for training and apply geometric rotation and flipping as data augmentations to avoid overfitting.

Comparison with State-of-the-art Methods

We compare the proposed SWBNet with several state-of-the-art sRGB-WB methods, including KNN-WB (Afifi et al. 2019a), DWB (Afifi and Brown 2020a), Interactive WB (Afifi and Brown 2020b) and MixedWB (Afifi, Brubaker,

Methods	Mean square error (MSE)				Mean angle error (MAE)				△E 2000			
	Mean	Q1	Q2	Q3	Mean	Q1	Q2	Q3	Mean	Q1	Q2	Q3
Mixedscene dataset												
Interactive WB	1059.88	616.24	896.90	1265.62	5.86°	4.56°	5.62°	6.62°	11.41	8.92	10.99	12.84
KNN-WB	1226.57	680.65	1062.64	1573.89	5.81°	4.29°	5.76°	6.85°	12.00	9.37	11.56	13.61
DWB	1130.59	621.00	886.32	1274.72	4.53°	3.55°	4.19°	5.21°	10.93	8.59	9.82	11.96
MixedWB	819.47	655.88	845.79	1000.82	5.43°	4.27°	4.89°	6.23°	10.61	9.42	10.72	11.81
SWBNet	**816.00**	**426.01**	**554.07**	**944.08**	**3.51°**	**2.78°**	**3.18°**	**3.80°**	**9.19**	**7.01**	**8.27**	**10.35**
Rendered Cube dataset												
Interactive WB	159.88	21.94	54.76	126.02	4.64°	2.12°	3.64°	5.98°	6.2	3.28	5.17	7.45
KNN-WB	194.98	27.43	57.08	118.21	4.12°	1.96°	3.17°	5.04°	5.68	3.22	4.61	6.70
DWB	80.46	**15.43**	33.88	**74.42**	3.45°	1.87°	2.82°	4.26°	4.59	2.68	3.81	5.53
MixedWB	161.80	24.48	**19.33**	90.81	4.05°	1.40 °	2.12°	4.88 °	4.89	**2.16**	**3.10**	6.78
SWBNet	**74.35**	20.46	40.04	86.95	**3.15°**	**1.33°**	**2.09°**	**4.12°**	**4.28**	2.40	3.56	**5.09**

Table 3: Quantitative comparison for WB accuracy on the Rendered Cube and Mixedscene datasets. The top results are in bold.

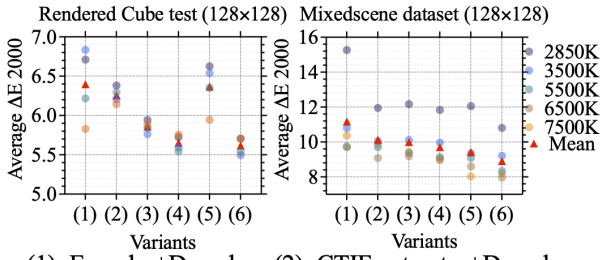
and Brown 2022). The comparison results are in Table.2, Fig.2 and Table.3 respectively.

WB Stability From Table.2 and Fig.2, WB instability appears for KNN-WB, DWB, and MixedWB, reflected by the significant differences among the average $\Delta E2000$ values for different color temperatures in Rendered Cube and Mixedscene datasets. In contrast, our SWBNet can reduce such differences significantly. In addition, from Fig.8, the color temperatures of the white-balance sRGB images generated by SWBNet are more consistent. These results indicate that our SWBNet can effectively alleviate WB instability by learning the color temperature-insensitive features.

WB Accuracy From Table.3, our SWBNet achieves the best accuracy on two public datasets. In the Mixedscene dataset, our SWBNet outperforms the previous methods by 13.38%, 25.58%, 22.85%, 12.36% in the mean, Q1, Q2, and Q3 of $\Delta\ E2000$ respectively. The improvements in the MSE and MAE are also significant. Meanwhile, in the Rendered Cube dataset, our SWBNet achieves the improvements of 6.75%, 7.96% in mean and Q2 of $\Delta\ E2000$. These results confirm that our SWBNet can achieve better WB accuracy, especially for the multi-illumination sRGB images.

Ablation Analysis

We carried out an ablation study for the SWBNet on the Rendered Cube and Mixedscene datasets. We trained six combinations with 8000 sRGB images from Rendered Cube dataset and use the remaining images (Rendered Cube test) and the Mixedscene dataset for evaluation (testing images are resized to 128×128). The results are in Fig.9. Compared with the vanilla U-Net (C1), replacing the encoder with the CTIF extractor (C2) or applying CT-contrastive loss (C3) can greatly reduce the differences of average $\Delta E2000$ values for various color temperatures. Further, as C4, the CT-contrastive loss and CTIF extractor cooperate in achieving smaller differences of average $\Delta E2000$ values. This indicates that learning color temperature-insensitive features is an effective way to alleviate WB instability. In addition, benefiting from the global view, the CTS-oriented transformer reduces the average $\Delta E2000$ values for the most color temperatures (comparing C5 with C1), especially for the multi-illumination sRGB images. According to the above, the proposed components work together to alleviate WB instability and improve WB accuracy, reflected in the closest and lowest average $\Delta E2000$ values in C6.



(1): Encoder+Decoder; (2): CTIF extractor+Decoder;
(3): Encoder+Decoder+CT-contrastive loss;
(4): CTIF extractor+Decoder+CT-contrastive loss;
(5): Encoder+Decoder+CTS-transformer; (6): SWBNet.

Figure 9: Ablation study of our SWBNet on Rendered Cube and Mixedscene datasets. The closer and lower average $\Delta E2000$ values mean better WB stability and accuracy.

Conclusion

In this paper, we propose a SWBNet to alleviate WB instability of the existing sRGB-WB methods. We achieve this by learning the color temperature-insensitive features via implicit color temperature influence reduction and explicit information-sharing promotion. Further, we correct multiple color temperature shifts differently based on the transformer's global receptive field. This helps to improve WB accuracy of stabilized SWBNet, especially for the multi-illumination sRGB images. Extensive experiments indicate that our SWBNet effectively learns the color temperature-insensitive features and corrects multiple color temperature shifts, thereby achieving stable and remarkable results. In future work, we plan to extend the SWBNet to improve the stability of temporal white balance, which is beneficial for mobile terminal filming.

Acknowledgments

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

References

Afifi, M.; and Brown, M. S. 2019. 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*, 243–252.

Afifi, M.; and Brown, M. S. 2020a. Deep White-Balance Editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1397–1406.

Afifi, M.; and Brown, M. S. 2020b. Interactive white balancing for camera-rendered images. In *Color and Imaging Conference*, 136–141. Society for Imaging Science and Technology.

Afifi, M.; Brubaker, M. A.; and Brown, M. S. 2022. Auto white-balance correction for mixed-illuminant scenes. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 1210–1219.

Afifi, M.; Price, B.; Cohen, S.; and Brown, M. S. 2019a. When Color Constancy Goes Wrong: Correcting Improperly White-Balanced Images. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 1535–1544.

Afifi, M.; Punnappurath, A.; Abdelhamed, A.; Karaimer, H. C.; Abuolaim, A.; and Brown, M. S. 2019b. Color temperature tuning: Allowing accurate post-capture white-balance editing. In *Color and Imaging Conference*, 1–6. Society for Imaging Science and Technology.

Chakrabarti, A.; Xiong, Y.; Sun, B.; Darrell, T.; Scharstein, D.; Zickler, T.; and Saenko, K. 2014. Modeling radiometric uncertainty for vision with tone-mapped color images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(11): 2185–2198.

El Helou, M.; Zhou, R.; and Süsstrunk, S. 2020. Stochastic frequency masking to improve super-resolution and denoising networks. In *European Conference on Computer Vision*, 749–766. Springer.

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

Gijsenij, A.; and Gevers, T. 2007. Color constancy using natural image statistics. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, 1–8. IEEE.

Guo, T.; Mousavi, H. S.; and Monga, V. 2019. Adaptive transform domain image super-resolution via orthogonally regularized deep networks. *IEEE Transactions on Image Processing*, 28(9): 4685–4700.

He, K.; Fan, H.; Wu, Y.; Xie, S.; and Girshick, R. 2020. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 9729–9738.

Hermann, K.; Chen, T.; and Kornblith, S. 2020. The origins and prevalence of texture bias in convolutional neural networks. *Advances in Neural Information Processing Systems*, 33: 19000–19015.

Hong, G.; Luo, M. R.; and Rhodes, P. A. 2001. 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.

Hu, Y.; Wang, B.; and Lin, S. 2017. Fc4: Fully convolutional color constancy with confidence-weighted pooling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4085–4094.

Karaimer, H. C.; and Brown, M. S. 2016. A software platform for manipulating the camera imaging pipeline. In *European Conference on Computer Vision*, 429–444. Springer.

Kim, D.; Kim, J.; Nam, S.; Lee, D.; Lee, Y.; Kang, N.; Lee, H.-E.; Yoo, B.; Han, J.-J.; and Kim, S. J. 2021. Large Scale Multi-Illuminant (LSMI) Dataset for Developing White Balance Algorithm under Mixed Illumination. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2390–2399.

Kim, S. J.; Lin, H. T.; Lu, Z.; Süsstrunk, S.; Lin, S.; and Brown, M. S. 2012. A new in-camera imaging model for color computer vision and its application. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(12): 2289–2302.

Kingma, D. P.; and Ba, J. 2015. Adam: A Method for Stochastic Optimization. In Bengio, Y.; and LeCun, Y., eds., *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Li, Y.; Fu, Q.; and Heidrich, W. 2021. Multispectral illumination estimation using deep unrolling network. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2652–2661.

Loshchilov, I.; and Hutter, F. 2019. Decoupled Weight Decay Regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Peng, X.; Bai, Q.; Xia, X.; Huang, Z.; Saenko, K.; and Wang, B. 2019. Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 1406–1415.

Ronneberger, O.; Fischer, P.; and Brox, T. 2015. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, 234–241. Springer.

Sharma, G.; Wu, W.; and Dalal, E. N. 2005. The CIEDE2000 color-difference formula: Implementation notes, supplementary test data, and mathematical observations. *Color Research & Application*, 30(1): 21–30.

Vazquez-Corral, J.; and Bertalmío, M. 2014. Color stabilization along time and across shots of the same scene, for one or several cameras of unknown specifications. *IEEE Transactions on Image Processing*, 23(10): 4564–4575.

Xie, E.; Wang, W.; Yu, Z.; Anandkumar, A.; Alvarez, J. M.; and Luo, P. 2021. SegFormer: Simple and efficient design for semantic segmentation with transformers. *Advances in Neural Information Processing Systems*, 34.

Xu, K.; Qin, M.; Sun, F.; Wang, Y.; Chen, Y.-K.; and Ren, F. 2020. Learning in the frequency domain. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1740–1749.

Zhao, Z.; Hu, H.-M.; Zhang, H.; Chen, F.; and Guo, Q. 2022. Improving Color Constancy Using Chromaticity-Line Prior. *IEEE Transactions on Multimedia*, 1–1.