Multi-Block-Attention-based Color Constancy

Oguzhan Ulucan, Diclehan Ulucan, and Marc Ebner

University of Greifswald, 17489 Greifswald, Germany
{oguzhan.ulucan, diclehan.ulucan, marc.ebner}@uni-greifswald.de

Abstract. Color constancy algorithms estimate and remove the effect of the illuminant from a captured image, enabling consistent color perception under varying illumination conditions. Learning-based methods achieve high accuracy but tend to be biased toward the spectral characteristics of training cameras. Traditional algorithms such as white-patch Retinex are computationally efficient and camera-independent, but underperform when scene statistics deviate from their assumptions. Therefore, in this paper, we propose a hybrid approach that combines classical assumptions with learned features. Our model processes images in a block-based manner through two complementary branches. The backbone branch extracts scene features and identify reliable regions for illuminant estimation. Guided by these saliency maps, the statistics branch computes per-block illumination priors using a soft maximum formulation that provides a differentiable approximation to the classical white-patch Retinex. Features from both branches are fused and refined through multi-head self-attention, which enables blocks to exchange information across the image, addressing the aggregation bias common to block-based methods. Experiments on five benchmarks demonstrate that our model achieves state-of-the-art performance with a compact architecture (∼1.1 MB) that requires no camera-specific calibration or metadata.

Keywords: Color constancy · Illuminant estimation · Hybrid model

1 Introduction

The human visual system perceives object colors consistently regardless of scene illumination [70]. This unconscious ability, known as *color constancy*, enables us to recognize the true colors of objects under varying lighting conditions [29]. For instance, we can perceive a white sheet of paper as white whether viewed under warm tungsten light or cool daylight. In contrast to the human visual system, machine vision systems lack this ability. The sensor of a camera captures scene colors that are biased by the illuminant, causing the same object to appear differently under varying lighting conditions [21, 34]. This makes color an unreliable feature for many applications from traffic light recognition to crop health monitoring [24, 34]. To address this limitation, the field of computational color constancy has emerged, which aims to estimate and remove the effect of the illuminant from captured images to obtain canonical, *i.e.*, white-balanced,

images that enable the reliable use of color features across applications [4,5]. To obtain a canonical image, the standard assumptions of Lambertian reflectance and a single uniform illuminant are followed [24, 34]. Thus, a linear image I is formed as the element-wise product of its white-balanced image W and the global illuminant color vector $\boldsymbol{\ell}$ at every pixel location x:

$$I^x = W^x \circ \boldsymbol{\ell}, \quad \boldsymbol{\ell} = [\ell_r, \ell_g, \ell_b]^T, \quad \forall x. \tag{1}$$

Even with these standard assumptions and the simplified image formation model, color constancy remains an ill-posed problem because sensor characteristics vary across devices and illumination conditions differ across scenes [2,3,14]. To tackle this challenge, numerous algorithms have been proposed, which can be broadly grouped into traditional methods [20,23,25,28,31,32,54,61] and learning-based models [8,9,11,13,18,30,33,38,43,44,49,64,68]. Many traditional methods rely on statistical assumptions. For instance, the white-patch Retinex method [47] assumes that the brightest pixels in each channel correspond to a white or highly reflective surface whose observed color directly reveals the illuminant. This algorithm has become a fundamental building block of several studies due to its computational simplicity and effectiveness [26, 27]. Yet, the assumption fails when scenes lack high-luminance cues or when specular highlights dominate the scene [24, 42]. In contrast, learning-based models have achieved state-of-the-art accuracy by learning complex mappings from scene appearance to illumination without relying on explicit statistical assumptions [52]. However, these methods implicitly learn the spectral response of the training cameras, limiting their ability to generalize to unseen sensors with different characteristics [2, 16]. In practice, deploying such models on a new camera typically requires collecting new training data and retraining. Recent efforts address this dependency via pre-calibrated color correction matrices [43], few-shot learning [64], or requiring additional images from the target camera at inference [1].

While traditional and learning-based methods have their differences, both reveal that not all pixels contribute equally to color constancy [10,38,42,53,54,58,66]. However, identifying the informative, *i.e.*, *salient*, pixels remains challenging. For instance, the classical white-patch assumption relies on a single maximum value per channel, which is sensitive to noise and specular highlights, and percentile-based alternatives still depend on fixed thresholds that do not generalize across scene content [24]. On the learning side, the quasi-unsupervised color constancy method [10] estimates the illuminant as a learned weighted combination of pixels, but operates globally and does not model how illumination cues vary across different regions of the scene. Block-based operations address these issues by computing statistics within local regions, which allow the model to capture spatially varying illumination cues rather than relying on a single global estimate [28,40,58,60]. However, determining which pixels within each block are informative and how to combine estimates across blocks remain open problems. Directly averaging per-block estimates introduces bias, as scenes containing large uniform regions such as sky or grass produce blocks that outnumber those with diverse color content and disproportionately affect the final estimate, degrading

overall performance [32,40,59]. Addressing this requires both an adaptive mechanism to identify reliable pixels within each block and a mechanism for blocks to exchange information about their illumination content, so that unreliable estimates can be identified and suppressed during combination. Self-attention is a natural candidate for the latter, as it allows any block to attend to any other regardless of spatial distance. While this mechanism has been explored in color constancy, it has not been used to address aggregation bias in block-based illuminant estimation. Prior works employ self-attention either over feature-domain representations [39,45], which lack spatial structure and therefore cannot determine dominant regions, or at the pixel level for multi-illuminant estimation [48], which targets spatially varying illumination rather than the aggregation bias that arises when combining local estimates into a global prediction.

In this study, we propose a hybrid model that embeds these observations directly into a learnable architecture consisting of two complementary branches. The backbone branch extracts features and predicts saliency maps that adaptively identify reliable pixels for illuminant estimation, eliminating the need for hand-tuned thresholds. The statistics branch computes block-wise illumination priors guided by these saliency maps, using a differentiable soft maximum formulation that considers multiple bright pixels rather than relying on a single potentially unreliable maximum. To enable effective combination of per-block estimates, features from both branches are fused and refined through multi-head self-attention, allowing blocks with limited color diversity or poor illumination cues to benefit from blocks with richer, more reliable content. The refined features are then used to predict per-block illuminant estimates and corresponding weights, which are combined to produce the final global illuminant estimate [1].

Our contributions can be summarized as follows:

- We propose a compact color constancy model that does not require any camera-specific calibration or metadata.
- We introduce a differentiable soft-maximum formulation guided by learned saliency maps that generalizes white-patch Retinex by replacing hand-tuned percentile thresholds with a learnable parameter optimized end-to-end.
- We employ multi-head self-attention over block-level representations to enable cross-block information exchange, allowing unreliable blocks to be refined by informative ones. This directly addresses the known limitation of block averaging bias, where dominant uniform regions disproportionately affect the estimate, and is, to the best of our knowledge, the first use of self-attention for this purpose in color constancy.

2 Proposed Method

Figure 1 presents the modules of our model *HybridCC*, while the following sections describe each component in detail.

[1] Source code and trained models available at https://github.com/OguzhanUlucan/HybridCC.

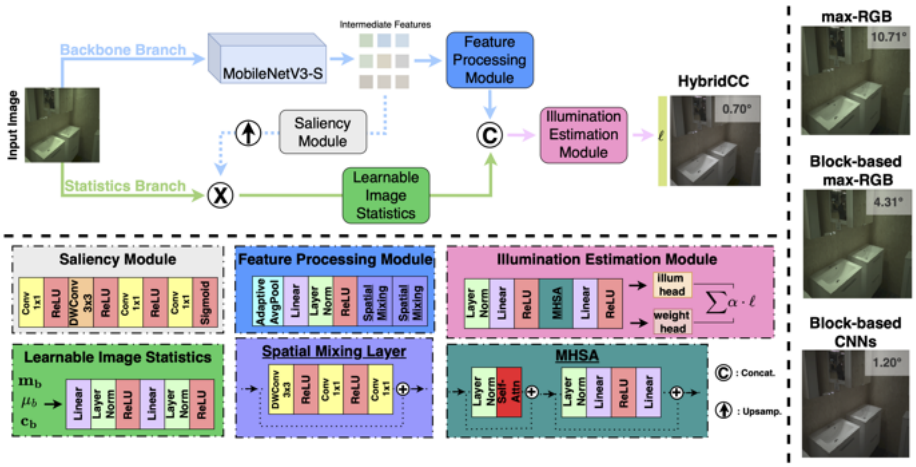

Fig. 1: Overview of our hybrid model. The backbone branch extracts features that are processed by a saliency module and a feature processing module, while the statistics branch computes block-wise illumination priors guided by the learned saliency map. Both branches are fused and fed into the illumination estimation module, which refines per-block representations through self-attention and produces the final illuminant estimate via weighted combination. (Right) A visual example illustrating the strengths of the proposed model compared to traditional and learning-based baselines.

2.1 Backbone Branch

The backbone branch extracts features and predicts saliency maps that identify reliable regions for illuminant estimation. As our feature extractor, we employ MobileNetV3-Small, which is suitable for resource-constrained environments [37]. Since the backbone is pretrained on ImageNet, the input is normalized using ImageNet statistics before feature extraction. Particularly, for a given input image $I \in \mathbb{R}^{H \times W \times 3}$, we extract a feature map $F \in \mathbb{R}^{h \times w \times C}$ from an intermediate layer, where C denotes the number of channels (Table 2). This feature map is then processed by two parallel modules—namely, a saliency module and a feature processing module.

The saliency module addresses the fact that not all pixels equally contribute to illuminant estimation by predicting per-pixel reliability weights $S = \sigma(f(F)) \in [0,1]^{h \times w}$, where σ denotes the sigmoid function. The saliency head f follows a bottleneck design with a 1×1 convolution that reduces channels by half, a 3×3 dilated depthwise convolution with dilation rate 2 for an increased receptive field, a 1×1 convolution for channel restoration, and a final 1×1 convolution that produces the single-channel output. Each layer except the last is followed by batch normalization and ReLU activation. Since no explicit saliency supervision is provided, the module learns to emphasize informative regions through the illuminant estimation objective, *i.e.*, regions that consistently contribute to accurate estimates are reinforced, while unreliable ones are suppressed. At the

end, the saliency map is upsampled to the input resolution before being passed to the statistics branch.

In parallel, the feature processing module obtains block-level representations by pooling F into a grid of $N_h \times N_w$ blocks via adaptive average pooling, yielding $P = N_h \times N_w$ block-level feature vectors, *e.g.*, $P = 144$ for an input resolution of 384×384 with block size 32×32 (Table 2). These features are projected to a D-dimensional space ($D = 64$) using a linear projection followed by layer normalization, ReLU, and dropout for regularization. Furthermore, to exploit the spatial smoothness of illumination, we apply spatial mixing layers that enable neighboring blocks to exchange local information through iterative refinement. Each layer consists of a depthwise 3×3 convolution operating on the block grid, followed by a pointwise expansion with a factor of 2 and projection, with batch normalization and ReLU activation after each convolution. Each layer applies a residual update with a learnable scaling factor:

$$F_b^{(t+1)} = F_b^{(t)} + \gamma \cdot \phi\Big(\{F_{b'}^{(t)}\}_{b' \in \mathcal{N}(b)}\Big), \quad \forall b \in \{1, \ldots, P\}. \tag{2}$$

where t denotes the iteration index, $\mathcal{N}(b)$ is the 3×3 spatial neighborhood of block b on the block grid (including b itself), $\phi(\cdot)$ denotes a local spatial mixing operator that combines information from neighboring blocks, and γ is a learnable scalar initialized to 0.1 which controls the strength of contextual refinement. We apply two spatial mixing layers sequentially to increase the effective receptive field over the block grid (Table 2). The final output is a set of block-level feature vectors $F^{\text{backbone}} \in \mathbb{R}^{P \times D}$, which are later fused with their corresponding statistical features for illuminant estimation.

2.2 Statistics Branch

The statistics branch extracts block-level illumination estimation cues. Given the input image I and the saliency map, the image is divided into P non-overlapping blocks. Within each block, saliency weights s_k are normalized so that $\sum_{k \in b} s_k = 1$, where k indexes the pixels belonging to block b. Since our model builds upon the white-patch Retinex assumption, we assume that within each block b there exists at least one sufficiently bright pixel that provides a reliable cue for estimating the illuminant. Furthermore, not all pixels contribute to color constancy, and among those that do, their contributions are not equal [6,53,59]. We address the first through saliency weighting that emphasizes informative pixels, and the second by weighting pixel contributions according to the relative brightness of each block. To this end, we define two descriptors: *(i)* a saliency-weighted soft maximum color descriptor $\mathbf{m}_b \in \mathbb{R}^3$ that emphasizes bright informative pixels in each channel, and *(ii)* a saliency-weighted mean intensity scalar $\mu_b \in \mathbb{R}$ that captures the overall brightness of the block.

To compute our first descriptor $\mathbf{m}_b$, we employ a temperature-controlled soft maximum formulation parameterized by τ, which provides a differentiable

approximation to a per-channel maximum:

$$\mathbf{m}_b = \tau \log \sum_{k \in b} s_k \exp\left(\frac{\mathbf{x}_k}{\tau}\right), \tag{3}$$

where $\mathbf{x}_k \in \mathbb{R}^3$ denotes the linear RGB value of pixel k within block b, and τ is a learnable parameter. As $\tau \to 0$, the descriptor becomes increasingly peaked and approaches a hard maximum over the brightest pixels, whereas larger values of τ distribute the contribution over a broader set of pixels. This can be viewed as a soft, differentiable generalization of the max-RGB assumption, where τ controls how sharply the descriptor focuses on the brightest pixels, while the saliency weights s_k determine which pixels are considered reliable. When τ converges to smaller values it leads to improved performance (Table 2), consistent with the observation that max-based cues provide more stable information in areas with limited color diversity [25,42]. Notably, the saliency maps that weight this descriptor are not supervised explicitly, nor constrained to favor bright or achromatic pixels. Yet, the model implicitly assigns higher saliency to pixels whose contribution to the statistics consistently reduces training loss.

While $\mathbf{m}_b$ captures chromatic information from the most informative pixels, it does not reflect the overall brightness of the block. We model this separately using a saliency-weighted mean intensity $\mu_b \in \mathbb{R}$:

$$\mu_b = \sum_{k \in b} s_k \cdot \frac{1}{3} \sum_{c \in \{r,g,b\}} x_{k,c}, \tag{4}$$

where c denotes the color channel. Given both descriptors, the block-level illumination prior $\mathbf{c}_b \in \mathbb{R}^3$ is obtained by adaptively scaling the chromatic descriptor $\mathbf{m}_b$ using the corresponding mean intensity μ_b:

$$\mathbf{c}_b = \frac{\mu_b}{\|\mathbf{m}_b\|_2^2}\, \mathbf{m}_b. \tag{5}$$

This formulation preserves the chromatic direction captured by the maximum descriptor while weighting its magnitude according to block brightness. As a result, brighter blocks, which typically provide more informative illumination cues [51,57], contribute more to the estimation.

Importantly, the $\mathbf{c}_b$ values are not treated as final illuminant estimates. Instead, they serve as statistical priors that guide the estimation process. This distinction is visualized in Fig. 2, where treating $\mathbf{c}_b$ directly as an illuminant estimate yields high per-block errors, particularly in blocks with limited color diversity or uniform content. However, using them as features in further stages of our model reduces these errors substantially, confirming that the statistical descriptors are most effective when used as features rather than standalone estimates. To this end, a two-layer perceptron transforms the low-dimensional descriptors into dense feature representations $F^{\text{stat}} \in \mathbb{R}^{P \times D}$ that allow the model to learn how to effectively use these priors. Each layer consists of a linear projection followed by layer normalization and ReLU activation, with dropout applied

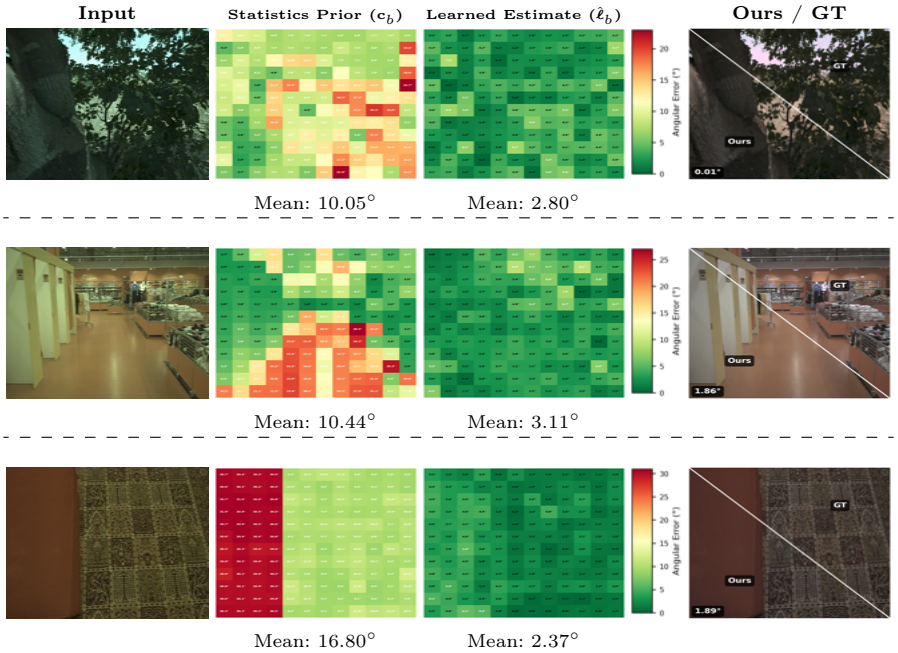

Fig. 2: Per-block angular error comparison between the statistics prior $\mathbf{c}_b$ and the learned estimate $\hat{\boldsymbol{\ell}}_b$. The learned estimates consistently reduce per-block errors across the image, demonstrating that the model effectively refines the initial statistical priors.

after the first layer. These encoded features are later fused with the corresponding backbone features, and subsequently processed by the illumination estimation module to find block-level illuminants and their corresponding weights, as described in the following section.

2.3 Illumination Estimation Module

The final stage fuses features from both branches to produce per-block illuminant estimates, which are then combined into a global prediction. For each block b, the encoded statistical features $F_b^{\text{stat}} \in \mathbb{R}^D$ and the corresponding backbone features $F_b^{\text{backbone}} \in \mathbb{R}^D$ are concatenated to form a fused block representation (Eqn. 6). The fused features are normalized and projected to D dimensions through a linear layer with ReLU activation.

$$F_b = \left[F_b^{\text{stat}}\,;\, F_b^{\text{backbone}}\right] \in \mathbb{R}^{2D}, \quad \forall b. \tag{6}$$

Afterwards, these projected features need to be refined as not all blocks contribute equally to the task of illumination estimation. Some contain strong cues while others provide weak or ambiguous information due to shadows or limited

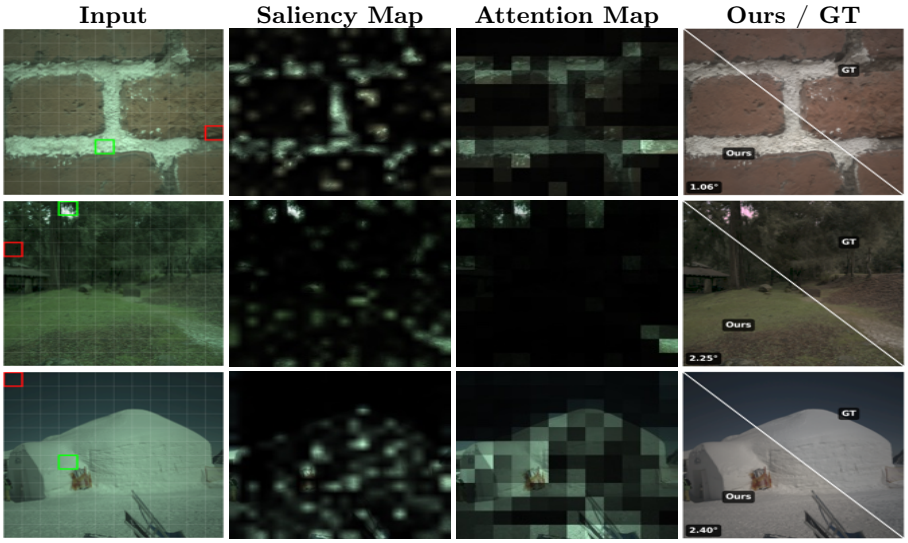

Fig. 3: Visualization of learned saliency and block communication. The maps have two distinct functions. The saliency map determines *which* pixels contribute to the statistics, while the attention map determines *how* blocks exchange information during refinement. In the former, the most informative regions are emphasized. In the latter, total attention received per block (blocks with higher attention are visualized more brightly) is presented, where the darkest (red) and brightest (green) blocks are marked on the input. The large uniform regions are suppressed by both maps, showing the model's ability to down-weight uninformative areas.

color diversity, and directly using their features can introduce bias. This is particularly problematic in scenes containing dominant regions such as shadows, sky, grass, or close-up objects, where features from these areas disproportionately affect the final result. Therefore, we use a multi-head self-attention mechanism [63] to refine per-block representations. This mechanism allows any block to attend to any other regardless of spatial distance, which enables unreliable blocks, such as those in dark or shadowed regions, to benefit from illuminant cues from well-lit, chromatically informative regions elsewhere in the image. Specifically, attention weights are computed via scaled dot-product attention with 2 heads over the D-dimensional block representations. The attention output is integrated through residual connections and a feed-forward network with an expansion factor of 2, with pre-norm layer normalization applied before each sub-layer [65]. A subsequent linear layer with ReLU activation produces the refined block representations. We present examples of this mechanism in Fig. 3, where the blocks receiving the highest attention correspond to the bright regions in the attention map. For instance, in the third row, the darkest block lies in the sky, and the attention map shows that the sky region receives no attention,

whereas the model directs its focus toward the building surfaces and ground, effectively suppressing the dominant sky region which could bias the estimates.

As a last step, from the refined features, two lightweight prediction heads produce per-block estimates. The illuminant head maps each block's D-dimensional representation to a 3-dimensional RGB estimate via a linear projection, followed by a softplus activation to ensure strictly positive values, yielding the per-block illuminant estimate $\hat{\boldsymbol{\ell}}_b \in \mathbb{R}^3$ (Fig. 2). The weight head maps the same representation to a scalar weight $w_b \in \mathbb{R}$ via a separate linear projection. The raw weights are normalized across blocks via temperature-scaled softmax to obtain combination coefficients α_b, and the final illuminant is computed as:

$$\hat{\boldsymbol{\ell}} = \frac{\boldsymbol{\ell}'}{\|\boldsymbol{\ell}'\|}, \quad \text{where} \quad \boldsymbol{\ell}' = \sum_{b=1}^{P} \alpha_b \, \hat{\boldsymbol{\ell}}_b, \quad \alpha_b = \frac{\exp(w_b/T)}{\sum_{j=1}^{P} \exp(w_j/T)}, \tag{7}$$

where T is a learnable parameter initialized to 1.2 that controls the sharpness of the weight distribution, *i.e.*, smaller values concentrate weight on fewer blocks, while larger values yield more uniform contributions. One might consider uniform averaging here; however, we observed that it degrades performance (Table 2).

3 Experiments

3.1 Experimental Setup

Datasets and augmentation. We evaluate our model on five benchmarks—namely, INTEL-TAU [46], Gehler-Shi [35], NUS-8 [20], Cube+ [7], and the test set of TA-AWB [5]. Intel-TAU contains $7,022$ images captured by 3 cameras across diverse conditions. In our experiments, we use only the *field1* and *field3* subsets, since the calibration object is unmasked in the other sets and no masks are provided for it. Gehler-Shi contains 568 indoor and outdoor images captured by 2 DSLR cameras. NUS-8 contains $1,736$ images with outdoor, indoor, and nighttime setups, and scenes are from 8 different cameras. Cube+ contains $1,707$ images captured by a single camera, primarily outdoor scenes. The test set of TA-AWB contains 400 indoor and outdoor scenes captured by Samsung S24 Ultra's main camera at various times of day. To ensure fair comparison, when required, we follow standard preprocessing, including black-level normalization and masking out the calibration object and over-saturated pixels.

During training, each image is represented by the original image resized to 384×384 alongside 3 augmented crops. Each crop is obtained from the original image by randomly selecting a square region whose side length ranges from 0.1 to 1.0 times the shorter edge, followed by random rotation in $[-180°, +180°]$, resizing to 384×384, and random horizontal and vertical flipping each with probability 0.5. To improve robustness, we also apply Gaussian blur (kernel size randomly chosen from $\{3, 5, 7\}$, $\sigma \in [0.05, 0.8]$) and additive Gaussian noise ($\sigma \in [0.001, 0.015]$), each with probability 0.5, along with a random global intensity gain in $[0.8, 1.2]$ with probability 0.5. Also, following [38], we apply independent

per-channel RGB gains sampled from $[0.8, 1.2]$ to both the image and the ground truth illuminant, effectively simulating varying illumination conditions.

Evaluation protocols. We evaluate our model under two protocols. Our primary protocol follows prior works [1, 43], by adopting a leave-one-dataset-out strategy. Under this protocol, the model is trained on three datasets and evaluated on the held-out one, ensuring no camera overlap between training and testing. For instance, when evaluating on Intel-TAU, the model is trained on NUS-8, Gehler–Shi, and Cube+. Our second protocol targets cross-camera generalization through two evaluations. First, we apply the cross-sensor (CS) protocol [2] on NUS-8, holding out each camera in turn and averaging results across folds. Second, we evaluate on the TA-AWB test set by training on all other datasets. Since this set is captured by a smartphone while the training data is largely DSLR-based, it provides a more challenging test of camera generalization.

Baselines and metrics. We compare our model against learning-free and state-of-the-art learning-based color constancy methods. To follow common practice, we measure the performance using the angular error between the estimated and ground truth illuminations [36]. Particularly, we report the mean angular error, the median, and the mean of the best 25% and worst 25% angular errors.

Implementation details. We implement our model in PyTorch and train it on a single NVIDIA RTX 4090 GPU with a batch size of 32 using 384×384 input images. We train our model for a maximum of 300 epochs, and optimize it by using AdamW [50] with a learning rate of 3×10^{-4} and weight decay of 5×10^{-5}. The learning rate schedule follows a linear warm-up for 20 epochs followed by cosine decay down to a minimum learning rate of 5×10^{-6}. To stabilize fine-tuning, we apply a smaller learning rate to the backbone, scaled by $0.1\times$ relative to the remaining parameters. When training on multiple datasets, we adopt balanced sampling where each dataset contributes equally to each epoch regardless of its size, to prevent larger datasets from dominating the training. Our model is trained end-to-end by minimizing the angular error between the predicted unit-norm illuminant $\hat{\boldsymbol{\ell}}$ and the ground truth $\boldsymbol{\ell}^*$, together with a sparsity regularizer that minimizes the entropy of the combination weights:

$$\mathcal{L}(\boldsymbol{\ell}^*, \hat{\boldsymbol{\ell}}) = \arccos\left(\frac{\hat{\boldsymbol{\ell}} \cdot \boldsymbol{\ell}^*}{\|\hat{\boldsymbol{\ell}}\| \, \|\boldsymbol{\ell}^*\|}\right) + \lambda \, \mathcal{L}_{sp}(\boldsymbol{\alpha}), \tag{8}$$

where $\mathcal{L}_{sp}(\boldsymbol{\alpha}) = -\sum_{b=1}^{P} \alpha_b \log(\alpha_b)$ is the entropy of the weight distribution. Since entropy is maximized when all weights are uniform, minimizing $\mathcal{L}_{sp}$ encourages the model to concentrate weight on a small subset of reliable blocks. We set $\lambda = 5 \times 10^{-4}$ in all experiments and select the best model based on the lowest validation loss, with early stopping after 30 epochs without improvement.

3.2 Results and Discussion

Table 1 summarizes our results. Our hybrid model achieves the lowest mean angular error on Gehler-Shi, NUS-8, Cube+, and INTEL-TAU, as well as on

Table 1: Statistical results. We report results on Gehler-Shi, NUS-8, and Cube+ from the recent comprehensive study of Kim *et al.* [43], results on INTEL-TAU from Ulucan *et al.* [59] and C5 [1], and results on TA-AWB from Afifi *et al.* [5].

Gehler-Shi	Mean	B.25%	Med.	W.25%
max-RGB [47]	7.78	1.49	5.43	17.47
SoG [25]	4.93	1.14	4.01	10.20
BB-CC [58]	3.48	1.07	2.71	7.37
GI [54]	3.07	0.43	1.87	7.62
FFCC [8]	2.95	0.57	2.19	6.75
SIIE [2]	2.77	0.55	1.93	6.53
C5 [1]	2.50	0.53	1.99	5.46
GCC [18]	2.35	0.78	2.02	**4.57**
Tra-CC [61]	2.32	0.68	1.76	4.85
CCMNet [43]	2.23	**0.36**	**1.53**	5.46
Ours	**2.11**	0.53	1.60	4.65

Cube+	Mean	B.25%	Med.	W.25%
GW [15]	3.52	0.60	2.55	7.98
SoG [25]	3.22	0.43	2.12	7.77
GE-1st [62]	3.06	0.55	2.05	7.22
Quasi-U [10]	2.69	0.49	1.76	6.45
FFCC [8]	2.69	0.46	1.89	6.31
DMCC [68]	2.23	0.49	1.63	4.95
SIIE [2]	2.14	0.44	1.44	5.06
C5 [1]	1.92	0.44	1.32	4.44
CCMNet [43]	1.68	**0.38**	1.16	3.89
Ours	**1.56**	**0.38**	**1.11**	**3.49**

INTEL-TAU	Mean	B.25%	Med.	W.25%
wGE [32]	6.00	0.90	4.20	14.20
SoG [25]	4.00	0.70	2.90	9.00
SIIE [2]	3.42	0.73	2.42	7.80
FFCC [8]	3.42	0.70	2.38	7.96
GI [54]	3.32	0.56	2.18	8.03
One-Net [22]	3.30	1.10	3.20	5.90
Quasi-U [10]	3.12	0.60	2.19	7.28
Rev. CNNs [60]	2.90	0.79	2.27	6.11
C5 [1]	2.52	**0.52**	**1.70**	5.96
Ours	**2.21**	0.59	**1.70**	**4.75**

NUS-8	Mean	B.25%	Med.	W.25%
GW [15]	4.59	1.16	3.46	9.85
SoG [25]	3.67	0.98	2.94	7.75
Quasi-U [10]	3.00	-	2.25	-
MS-CC [59]	2.93	0.74	2.21	6.37
PCA-CC [20]	2.93	0.78	2.33	6.13
GI [54]	2.91	0.56	1.97	6.67
FFCC [8]	2.87	0.71	2.14	6.23
C5 [1]	2.54	0.61	1.90	5.61
CCMNet [43]	2.32	**0.53**	**1.71**	5.18
Ours	**2.26**	0.60	1.79	**4.78**

NUS-8 (CS)	Mean	B.25%	Med.	W.25%
DMCC [68]	2.80	0.74	2.12	5.88
SIIE [2]	2.05	0.52	1.50	4.48
GCC [18]	2.03	0.77	1.78	3.69
C5 [1]	1.77	**0.48**	1.37	3.75
CCMNet [43]	1.71	**0.48**	**1.31**	3.62
Ours$_{\text{wo Stats}}$	2.01	0.79	1.74	3.71
Ours	**1.58**	0.55	1.36	**2.97**

TA-AWB	Mean	B.25%	Med.	W.25%
GI [54]	4.70	0.44	3.19	11.63
RGP [19]	4.59	0.43	3.13	11.13
PCA-CC [20]	4.42	0.70	3.54	9.59
TECC [12]	4.12	0.74	3.23	9.10
SIIE* [2]	4.09	0.91	3.25	8.97
max-RGB [47]	4.01	1.06	2.92	8.57
Quasi-U* [10]	3.85	0.55	2.97	8.47
C5* [1]	3.22	0.78	2.51	6.97
CFCC [17]	3.07	0.73	2.20	6.87
PCC [67]	3.03	0.53	2.13	7.08
TLCC [56]	2.71	0.66	2.06	5.89
FFCC [8]	2.62	0.37	1.46	6.89
TA-AWB [5]	**1.84**	**0.35**	**1.24**	**4.41**
Ours	2.63	0.52	1.95	5.80

the NUS-8 cross-sensor protocol, with particularly strong performance on the worst-25% of cases. On TA-AWB, our model remains competitive without using any data from this dataset during training.

Fig. 4 presents visual comparisons with GI [54], a competitive learning-free method based on salient pixels, and GCC [18], a state-of-the-art diffusion-based model. The first two rows show outdoor scenes dominated by uniform regions such as sky and ground, where our model suppresses these areas through learned saliency and attention. The fourth row illustrates an indoor scene with multiple light sources and highlights, which cause a challenge to the algorithms, but our model learns to identify salient regions adaptively to overcome this issue. Generally, GCC handles these cases better, yet it formulates color constancy as a color chart inpainting problem, which introduces additional complexity, and as

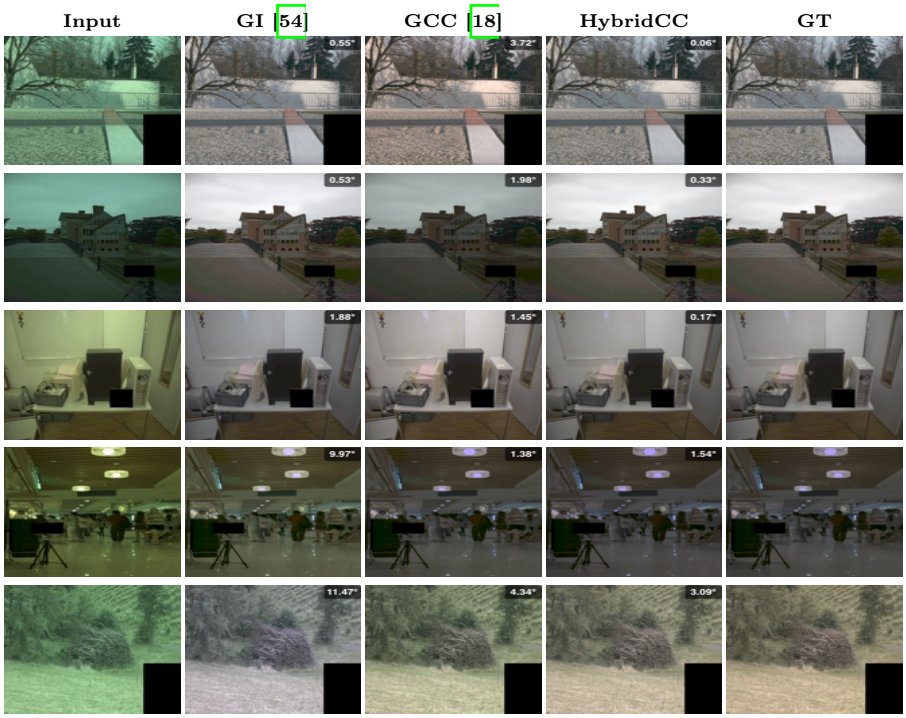

Fig. 4: Visual comparison of methods. Angular errors are shown on the images.

a diffusion model, GCC processes a single image in ∼180 ms [18]. In contrast, our model achieves better performance with faster inference (∼2.2 ms) on the same GPU, while remaining compact. However, despite its strong performance, our model has certain limitations as presented in the last row. The scene contains limited color diversity and very few informative pixels. With a limited number of salient pixels and few blocks carrying reliable information, our model has fewer cues to propagate through attention, which limits its ability to refine the estimate.

Model Size. Our model is 1.07 MB, comparable to CCMNet (1.05 MB) and smaller than SIIE (10.3 MB) and C5 (1.74 MB). Since the statistics branch adds only a single learned parameter and a compact perceptron, increasing the number of blocks does not significantly affect model size.

Cross-Camera Generalization. Our model generalizes to unseen cameras without requiring any explicit camera-specific information, adaptation, or fine-tuning. We observed that our model's practical generalization arises primarily from the statistics branch. For instance, when we remove it on NUS-8 (CS), the camera generalization decreases substantially by 22%, which demonstrates that the statistical priors are critical and the backbone alone is insufficient. We further examine our generalization ability on the TA-AWB benchmark, where

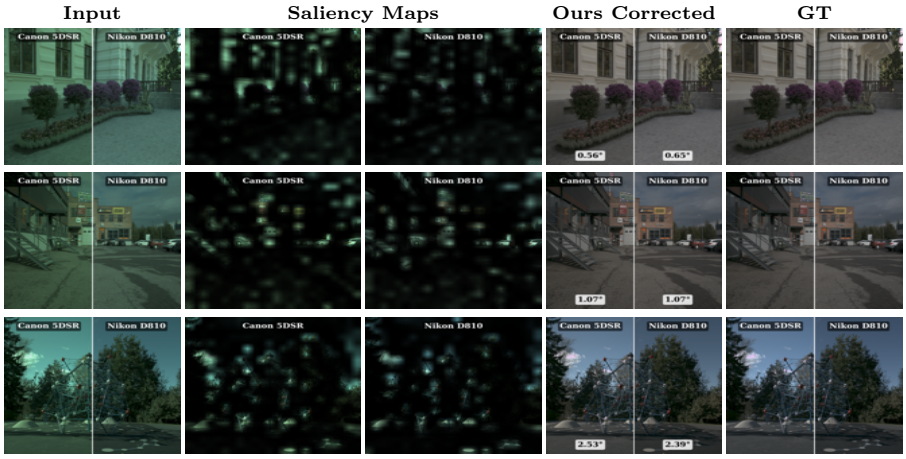

Fig. 5: Visualization of saliency stability. Each row shows the same scene from INTEL-TAU captured by Canon 5DSR and Nikon D810. Despite differing camera responses and slight viewpoint variations, our model produces largely consistent saliency patterns and accurate illuminant estimates.

our model achieves strong performance without using any data from this dataset during training. However, the competing learning-based methods, except for the ones marked with an asterisk, are trained on the dataset's own training split [5]. Despite this advantage, our model remains competitive on an unseen smartphone sensor, fundamentally different from the DSLR-dominated training data.

Another aspect worth addressing is that the backbone is trained on images from specific cameras. However, its role is fundamentally different from that in purely learning-based methods. Rather than directly mapping pixel values to illuminant estimates, where camera-specific spectral biases are directly encoded, our backbone predicts saliency maps and contextual features that determine *which* block-level statistical priors are reliable. The generalizability of our model is further illustrated in Fig. 5, where we analyze the same scenes captured with different cameras. Since these cameras have different spectral sensitivities, the same scene produces distinct sensor responses. Yet, the model tends to attend to similar informative regions regardless of the camera type, suggesting that the learned saliency is driven primarily by scene content rather than sensor-specific features.

Ablation study. Table 2 presents ablation results by reporting the mean angular error. We select the best configuration based on the lowest average error across all three datasets. First, we examine the contribution of each branch and observe that removing either one degrades performance substantially, confirming that both branches provide supporting information. Among the two, removing the backbone branch causes the largest drop, as the model loses access to learned saliency maps, local spatial mixing, and the features that complement the sta-

Table 2: Ablation study on model components and training strategy (left), block size (top-right), and backbone depth and architecture versus model size (bottom-right). The best configuration is selected based on the lowest average error across all datasets.

Model components	Gehler-Shi	NUS-8	Cube+	Avg
w/o Stats	2.41	2.59	2.00	2.33
w/ $\mathbf{m}_b$ only	2.23	2.21	1.86	2.10
w/ smooth τ (τ_{init}=0.5)	2.11	2.44	1.77	2.11
w/o Backbone	2.51	2.57	1.95	2.34
w/o Saliency	2.13	2.32	1.73	2.06
w/o Self-Attention	2.16	2.36	1.69	2.07
w/o Spatial Mixing	2.18	2.35	1.75	2.09
1 mixing layer	2.14	2.30	1.74	2.06
3 mixing layers	2.17	2.27	1.79	2.08
Training strategy				
w/o Sparsity Loss	2.33	2.32	1.78	2.14
w/ equal block weights	2.36	2.35	1.81	2.17
Ours (2 mixing layers, τ_{init}=0.02)	2.11	2.26	1.56	**1.97**

Block size	Gehler-Shi	NUS-8	Cube+	Avg
16 × 16	2.13	2.41	1.65	2.06
32 × 32	2.11	2.26	1.56	**1.97**
48 × 48	2.13	2.34	1.71	2.06
96 × 96	2.21	2.56	1.91	2.23

Backbone / (Depth)	Gehler-Shi	NUS-8	Cube+	Avg	Size (MB)
MobileNetV3-S (4)	2.53	2.61	2.24	2.46	0.9
MobileNetV3-S (8)	2.11	2.26	1.56	**1.97**	**1.1**
MobileNetV3-S (10)	2.03	2.28	1.74	2.02	3.1
MobileNetV3-S (12)	2.16	2.27	1.85	2.09	5.9
MnasNet [55]	2.45	2.41	1.84	2.23	3.3
SqueezeNet [41]	2.27	2.24	1.78	2.10	1.8

tistical descriptors. Within the statistics branch, using only the soft maximum descriptor $\mathbf{m}_b$ without weighting it by local brightness μ_b (Eq. 5) degrades results since the per-block statistic then captures only the chromatic direction of the brightest pixels but discards the brightness context that distinguishes informative blocks from dark or low-contrast ones. Similarly, the initialization of τ in Eq. 3 affects performance. Initializing $\tau = 0.02$, which remains near its lower bound during training, indicates that it focuses on the brightest pixels, whereas a smoother initialization ($\tau_{init} = 0.5$) degrades performance, confirming that precise pixel selection is more beneficial than broad averaging. During training, we observed that the learned τ converges to consistently small values across datasets (0.010 for Cube+ and Gehler-Shi, 0.014 for INTEL-TAU and TA-AWB, and within $[0.013, 0.019]$ across the 8 NUS folds), indicating stable behavior of the soft-maximum across different sensors and capture conditions.

Beyond the two branches, disabling learned saliency, self-attention, or spatial mixing results in consistent performance drops, which validates that all components contribute to the final result. Moreover, increasing the number of spatial mixing layers, which extends the receptive field over the grid, does not consistently improve performance. Similarly, block size 32×32 achieves the best overall performance, as smaller blocks reduce the spatial context available within each block, while larger blocks limit the ability to capture locally varying illumination cues. We would also like to note that while training, removing the sparsity regularizer or replacing learned weights with uniform averaging degrades performance, confirming that selective block weighting is essential. Without it, blocks from dominant uniform areas contribute equally to the final estimate, reintroducing the averaging bias that block-based methods are known to suffer from.

For the backbone analysis, we observed that extracting features from layer 8 provides the best balance between feature quality and model size at ~ 1.1 MB. Deeper layers offer marginal gains on individual benchmarks but at significantly increased model size, making depth 8 the best overall choice. This is consis-

tent with the observation that deeper layers of ImageNet-pretrained networks progressively capture more abstract and class-specific representations that are increasingly invariant to appearance variations such as illumination [69]. While this invariance is desirable for recognition tasks, we observed that deeper layers degrade color constancy performance. We attribute this to the loss of low-level appearance information at greater depth, combined with an increased risk of overfitting to the characteristics of the training cameras due to the larger number of parameters. As a final note, we compared alternative backbones under the same protocol to demonstrate that different backbones produce comparable results. We observed that they generalize across held-out datasets, but we chose MobileNetV3-Small as it achieves the best accuracy at the smallest size.

4 Conclusion

A key limitation of block-based color constancy is the block averaging bias, where dominant uniform regions disproportionately affect the final estimate. In this paper, we presented HybridCC, a compact model that addresses this through multi-head self-attention over block-level representations. HybridCC enables blocks to exchange information regardless of their spatial location so that unreliable ones can be refined by informative blocks.

Our model consists of two branches. The backbone branch extracts features and predicts saliency maps that identify reliable pixels for illuminant estimation, while the statistics branch uses these maps to compute block-wise illumination priors that approximate the white-patch Retinex assumption in a differentiable manner. Features from both branches are fused and refined through the self-attention mechanism. The refined features are then used to predict per-block illuminant estimates and combination coefficients, which are fused into the final prediction. Despite its compact size ($\sim$ 1.1 MB) and fast inference ($\sim$ 2.2 ms), HybridCC achieves state-of-the-art accuracy on standard benchmarks, with particularly strong behavior in worst-case scenarios, while remaining competitive on an unseen smartphone sensor. These results demonstrate our model's practical generalizability.

Our model achieves state-of-the-art performance, yet two limitations remain. First, although it generalizes well across the cameras used in our analyses, this generalization is not guaranteed for sensors whose spectral sensitivities differ substantially from the training data. While we use statistical priors to guide the estimation process, HybridCC further processes them through learned layers. Thus, even though the descriptors originate from classical statistical priors, the learned parameters may still encode sensor-specific characteristics seen during training. Second, our model is built on the assumption of a uniform illuminant source. This assumption may be violated in the presence of multiple light sources or emissive surfaces, which can attract excessive saliency and attention since the model may not distinguish them from surfaces that reflect the scene illuminant. Future work will address the latter by extending the framework to spatially varying illuminant estimation for more challenging conditions.

References

1. Afifi, M., Barron, J.T., LeGendre, C., Tsai, Y.T., Bleibel, F.: Cross-camera convolutional color constancy. In: ICCV (2021)
2. Afifi, M., Brown, M.S.: Sensor-independent illumination estimation for DNN models. BMVC (2019)
3. Afifi, M., Abuolaim, A.: Semi-supervised raw-to-raw mapping. In: BMVC (2021)
4. Afifi, M., Price, B., Cohen, S., Brown, M.S.: When color constancy goes wrong: Correcting improperly white-balanced images. In: CVPR (2019)
5. Afifi, M., Zhao, L., Punnappurath, A., Abdelsalam, M.A., Zhang, R., Brown, M.S.: Time-aware auto white balance in mobile photography. In: ICCV (2025)
6. Akazawa, T., Kinoshita, Y., Shiota, S., Kiya, H.: N-white balancing: White balancing for multiple illuminants including non-uniform illumination. IEEE Access **10**, 89051–89062 (2022)
7. Banić, N., Koščević, K., Lončarić, S.: Unsupervised learning for color constancy. arXiv preprint arXiv:1712.00436 (2017)
8. Barron, J.T., Tsai, Y.T.: Fast fourier color constancy. In: CVPR (2017)
9. Barron, J.T.: Convolutional color constancy. In: ICCV (2015)
10. Bianco, S., Cusano, C.: Quasi-unsupervised color constancy. In: CVPR (2019)
11. Bianco, S., Cusano, C., Schettini, R.: Single and multiple illuminant estimation using convolutional neural networks. IEEE Transaction on Image Processing **26**(9), 4347–4362 (2017)
12. Bianco, S., Buzzelli, M.: Truncated edge-based color constancy. In: ICCE. IEEE (2022)
13. Bianco, S., Cusano, C., Schettini, R.: Color constancy using cnns. In: CVPRW (2015)
14. Brown, M.: Color processing for digital cameras. Fundamentals and applications of colour engineering pp. 81–98 (2023)
15. Buchsbaum, G.: A spatial processor model for object colour perception. Journal of the Franklin institute **310**, 1–26 (1980)
16. Buzzelli, M., Zini, S., Bianco, S., Ciocca, G., Schettini, R., Tchobanou, M.K.: Analysis of biases in automatic white balance datasets and methods. Color Research & Application **48**(1), 40–62 (2023)
17. Buzzelli, M., Bianco, S.: A convolutional framework for color constancy. IEEE Transactions on Neural Networks and Learning Systems **36**(7), 13387–13401 (2024)
18. Chang, C.W., Fan, C.D., Chang, C.C., Lo, Y.C., Tseng, Y.C., Huang, J.L., Liu, Y.L.: Gcc: Generative color constancy via diffusing a color checker. In: CVPR (2025)
19. Cheng, C., Yang, K.F., Wan, X.M., Chan, L.L.H., Li, Y.J.: Nighttime color constancy using robust gray pixels. Journal of the Optical Society of America A **41**(3), 476–488 (2024)
20. Cheng, D., Prasad, D.K., Brown, M.S.: Illuminant estimation for color constancy: Why spatial-domain methods work and the role of the color distribution. Journal of the Optical Society of America A **31**, 1049–1058 (2014)
21. Cheng, D., Price, B., Cohen, S., Brown, M.S.: Beyond white: Ground truth colors for color constancy correction. In: ICCV (2015)
22. Domislović, I., Vršnak, D., Subašić, M., Lončarić, S.: One-net: Convolutional color constancy simplified. Pattern Recognition Letters **159**, 31–37 (2022)
23. Ebner, M.: Color constancy using local color shifts. In: ECCV. pp. 276–287 (2004)
24. Ebner, M.: Color Constancy, 1st ed. Wiley Publishing, Hoboken, NJ, USA (2007)

25. Finlayson, G.D., Trezzi, E.: Shades of gray and colour constancy. In: Color Imaging Conference. pp. 37–41. Scottsdale, AZ, USA (2004)
26. Funt, B., Shi, L.: The effect of exposure on maxrgb color constancy. In: Human Vision and Electronic Imaging XV. vol. 7527, pp. 282–288 (2010)
27. Funt, B., Shi, L.: The rehabilitation of maxrgb. In: Color Imaging Conference (2010)
28. Gao, S., Han, W., Yang, K., Li, C., Li, Y.: Efficient color constancy with local surface reflectance statistics. In: ECCV (2014)
29. Gegenfurtner, K.R.: Cortical mechanisms of colour vision. Nature Reviews Neuroscience **4**(7), 563–572 (2003)
30. Gehler, P.V., Rother, C., Blake, A., Minka, T., Sharp, T.: Bayesian color constancy revisited. In: CVPR (2008)
31. Gijsenij, A., Gevers, T., Van De Weijer, J.: Physics-based edge evaluation for improved color constancy. In: CVPR. Miami, FL, USA (2009)
32. Gijsenij, A., Gevers, T., Van De Weijer, J.: Improving color constancy by photometric edge weighting. IEEE Transactions on Pattern Analysis and Machine Intelligence **34**, 918–929 (2011)
33. Gijsenij, A., Gevers, T., Van De Weijer, J.: Generalized gamut mapping using image derivative structures for color constancy. International Journal of Computer Vision **86**(2), 127–139 (2010)
34. Gijsenij, A., Gevers, T., Van De Weijer, J.: Computational color constancy: Survey and experiments. IEEE Transactions on Image Processing **20**(9), 2475–2489 (2011)
35. Hemrit, G., Finlayson, G.D., Gijsenij, A., Gehler, P., Bianco, S., Funt, B., Drew, M., Shi, L.: Rehabilitating the colorchecker dataset for illuminant estimation. In: Color Imaging Conference (2018)
36. Hordley, S.D., Finlayson, G.D.: Re-evaluating colour constancy algorithms. In: ICPR (2004)
37. Howard, A., Sandler, M., Chu, G., Chen, L.C., Chen, B., Tan, M., Wang, W., Zhu, Y., Pang, R., Vasudevan, V., et al.: Searching for mobilenetv3. In: ICCV (2019)
38. Hu, Y., Wang, B., Lin, S.: Fc4: Fully convolutional color constancy with confidence-weighted pooling. In: CVPR (2017)
39. Huang, Z., Wu, T., Cui, T., Xie, X.: Saaf-svr for computational color constancy. Journal of the Optical Society of America A **42**(10), 1473–1482 (2025)
40. Hussain, M.A., Akbari, A.S.: Color constancy adjustment using sub-blocks of the image. IEEE Access **6**, 46617–46629 (2018)
41. Iandola, F.N., Han, S., Moskewicz, M.W., Ashraf, K., Dally, W.J., Keutzer, K.: Squeezenet: Alexnet-level accuracy with 50x fewer parameters and$<$ 0.5 mb model size. arXiv preprint arXiv:1602.07360 (2016)
42. Joze, H.R.V., Drew, M.S., Finlayson, G.D., Rey, P.A.T.: The role of bright pixels in illumination estimation. In: Color Imaging Conference (2012)
43. Kim, D., Afifi, M., Kim, D., Brown, M.S., Kim, S.J.: Ccmnet: Leveraging calibrated color correction matrices for cross-camera color constancy. In: ICCV (2025)
44. Koskinen, S., Yang, D., Kämäräinen, J.: Cross-dataset color constancy revisited using sensor-to-sensor transfer. In: BMVC (2020)
45. Laakom, F., Passalis, N., Raitoharju, J., Nikkanen, J., Tefas, A., Iosifidis, A., Gabbouj, M.: Bag of color features for color constancy. IEEE Transactions on Image Processing **29**, 7722–7734 (2020)
46. Laakom, F., Raitoharju, J., Nikkanen, J., Iosifidis, A., Gabbouj, M.: INTEL-TAU: A color constancy dataset. IEEE Access **9**, 39560–39567 (2021)
47. Land, E.H.: The retinex theory of colour vision. Proc. Roy. Institution Gr. Britain **47**, 23–58 (1974)

48. Li, S., Wang, J., Brown, M.S., Tan, R.T.: Mimt: Multi-illuminant color constancy via multi-task local surface and light color learning. arXiv preprint arXiv:2211.08772 (2022)
49. Lo, Y.C., Chang, C.C., Chiu, H.C., Huang, Y.H., Chen, C.P., Chang, Y.L., Jou, K.: Clcc: Contrastive learning for color constancy. In: CVPR (2021)
50. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. In: ICLR (2019)
51. Morimoto, T., Kusuyama, T., Fukuda, K., Uchikawa, K.: Human color constancy based on the geometry of color distributions. Journal of Vision **21**(3), 7–7 (2021)
52. Oh, S.W., Kim, S.J.: Approaching the computational color constancy as a classification problem through deep learning. Pattern Recognition **61**, 405–416 (2017)
53. Ono, T., Kondo, Y., Sun, L., Kurita, T., Moriuchi, Y.: Degree-of-linear-polarization-based color constancy. In: CVPR (2022)
54. Qian, Y., Kämäräinen, J.K., Nikkanen, J., Matas, J.: On finding gray pixels. In: CVPR (2019)
55. Tan, M., Chen, B., Pang, R., Vasudevan, V., Sandler, M., Howard, A., Le, Q.V.: Mnasnet: Platform-aware neural architecture search for mobile. In: CVPR (2019)
56. Tang, Y., Kang, X., Li, C., Lin, Z., Ming, A.: Transfer learning for color constancy via statistic perspective. In: AAAI (2022)
57. Uchikawa, K., Fukuda, K., Kitazawa, Y., MacLeod, D.I.: Estimating illuminant color based on luminance balance of surfaces. Journal of the Optical Society of America A **29**(2), A133–A143 (2012)
58. Ulucan, O., Ulucan, D., Ebner, M.: Block-based color constancy: The deviation of salient pixels. In: ICASSP (2023)
59. Ulucan, O., Ulucan, D., Ebner, M.: Multi-scale color constancy based on salient varying local spatial statistics. The Visual Computer pp. 1–17 (2023)
60. Ulucan, O., Ulucan, D., Ebner, M.: Revisiting color constancy using cnns: Including recent observations. In: CCIW (2024)
61. Ulucan, O., Ulucan, D., Ebner, M.: A traditional approach for color constancy and color assimilation illusions with its applications to low-light image enhancement. International Journal of Computer Vision **134**(1), 33 (2026)
62. Van De Weijer, J., Gevers, T., Gijsenij, A.: Edge-based color constancy. IEEE Transaction on Image Processsing **16**, 2207–2214 (2007)
63. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Advances in Neural Information Processing Systems **30** (2017)
64. Xiao, J., Gu, S., Zhang, L.: Multi-domain learning for accurate and few-shot color constancy. In: CVPR (2020)
65. Xiong, R., Yang, Y., He, D., Zheng, K., Zheng, S., Xing, C., Zhang, H., Lan, Y., Wang, L., Liu, T.: On layer normalization in the transformer architecture. In: ICML (2020)
66. Yang, K.F., Gao, S.B., Li, Y.J.: Efficient illuminant estimation for color constancy using grey pixels. In: CVPR (2015)
67. Yue, S., Wei, M.: Color constancy from a pure color view. Journal of the Optical Society of America A **40**(3), 602–610 (2023)
68. Yue, S., Wei, M.: Effective cross-sensor color constancy using a dual-mapping strategy. Journal of the Optical Society of America A **41**(2), 329–337 (2024)
69. Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional networks. In: ECCV (2014)
70. Zeki, S.: A Vision of the Brain. Blackwell Science, Oxford (1993)