A Multi-Hypothesis Approach to Color Constancy

Daniel Hernandez-Juarez¹, Sarah Parisot¹,², Benjamin Busam¹,³, Aleš Leonardis¹
Gregory Slabaugh¹, Steven McDonagh¹
dhernandez0@gmail.com,
{sarah.parisot, benjamin.busam, ales.leonardis, gregory.slabaugh, steven.mcdonagh}@huawei.com

¹Huawei Noah’s Ark Lab ²Mila, Montréal ³Technical University of Munich

Abstract

Contemporary approaches frame the color constancy problem as learning camera specific illuminant mappings. While high accuracy can be achieved on camera specific data, these models depend on camera spectral sensitivity and typically exhibit poor generalisation to new devices. Additionally, regression methods produce point estimates that do not explicitly account for potential ambiguities among plausible illuminant solutions, due to the ill-posed nature of the problem. We propose a Bayesian framework that naturally handles color constancy ambiguity via a multi-hypothesis strategy. Firstly, we select a set of candidate scene illuminants in a data-driven fashion and apply them to a target image to generate a set of corrected images. Secondly, we estimate, for each corrected image, the likelihood of the light source being achromatic using a camera-agnostic CNN. Finally, our method explicitly learns a final illumination estimate from the generated posterior probability distribution. Our likelihood estimator learns to answer a camera-agnostic question and thus enables effective multi-camera training by disentangling illuminant estimation from the supervised learning task. We extensively evaluate our proposed approach and additionally set a benchmark for novel sensor generalisation without re-training. Our method provides state-of-the-art accuracy on multiple public datasets (up to 11% median angular error improvement) while maintaining real-time execution.

1. Introduction

Color constancy is an essential part of digital image processing pipelines. When treated as a computational process, this involves estimation of scene light source color, present at capture time, and correcting an image such that its appearance matches that of the scene captured under an achromatic light source. The algorithmic process of recovering the illuminant of a scene is commonly known as computational Color Constancy (CC) or Automatic White Balance (AWB). Accurate estimation is essential for visual aesthetics [24], as well as downstream high-level computer vision tasks [2, 4, 13, 17] that typically require color-unbiased and device-independent images.

Under the prevalent assumption that the scene is illuminated by a single or dominant light source, the observed pixels of an image are typically modelled using the trichromatic model of Lambertian image formation captured under a trichromatic photosensor:
\[
\rho_k(X) = \int_{\Omega} E(\lambda)S(\lambda, X)C_k(\lambda) d\lambda \quad k \in \{R, G, B\}.
\]

where \(\rho_k(X)\) is the intensity of color channel \(k\) at pixel location \(X\), \(\lambda\) the wavelength of light such that \(E(\lambda)\) represents the spectrum of the illuminant, \(S(\lambda, X)\) the surface reflectance at pixel location \(X\) and \(C_k(\lambda)\) camera sensitivity function for channel \(k\), considered over the spectrum of wavelengths \(\Omega\).

The goal of computational CC then becomes estimation of the global illumination color \(\rho_k^E\) where:

\[
\rho_k^E = \int_{\Omega} E(\lambda)C_k(\lambda) d\lambda \quad k \in \{R, G, B\}. \tag{2}
\]

Finding \(\rho_k^E\) in Eq. (2) results in an ill-posed problem due to the existence of infinitely many combinations of illuminant and surface reflectance that result in identical observations at each pixel \(X\).

A natural and popular solution for learning-based color constancy is to frame the problem as a regression task [1, 28, 25, 10, 48, 34, 9]. However, typical regression methods provide a point estimate and do not offer any information regarding possible alternative solutions. Solution ambiguity is present in many vision domains [45, 36] and is particularly problematic in the cases where multi-modal solutions exist [35]. Specifically for color constancy we note that, due to the ill-posed nature of the problem, multiple illuminant solutions are often possible with varying probability. Data-driven approaches that learn to directly estimate the illuminant result in learning tasks that are inherently camera-specific due to the camera sensitivity function c.f. Eq. (2). This observation will often manifest as a sensor domain gap; models trained on a single device typically exhibit poor generalisation to novel cameras.

In this work, we propose to address the ambiguous nature of the color constancy problem through multiple hypothesis estimation. Using a Bayesian formulation, we discretise the illuminant space and estimate the likelihood that each considered illuminant accurately corrects the observed image. We evaluate how plausible an image is after illuminant correction, and gather a discrete set of plausible solutions in the illuminant space. This strategy can be interpreted as framing color constancy as a classification problem, similar to recent promising work in this direction [6, 7, 38]. Discretisation strategies have also been successfully employed in other computer vision domains, such as 3D pose estimation [35] and object detection [42, 43], resulting in e.g. state of the art accuracy improvement.

In more detail, we propose to decompose the AWB task into three sub-problems: a) selection of a set of candidate illuminants b) learning to estimate the likelihood that an image, corrected by a candidate, is illuminated achromatically, and c) combining candidate illuminants, using the estimated posterior probability distribution, to produce a final output.

We correct an image with all candidates independently and evaluate the likelihood of each solution with a shallow CNN. Our network learns to estimate the likelihood of white balance correctness for a given image. In contrast to prior work, we disentangle camera-specific illuminant estimation from the learning task thus allowing to train a single, device agnostic, AWB model that can effectively leverage multi-device data. We avoid distribution shift and resulting domain gap problems [1, 41, 22], associated with camera specific training, and propose a well-founded strategy to leverage multiple data. Principled combination of datasets is of high value for learning based color constancy given the typically small nature of individual color constancy datasets (on the order of only hundreds of images). See Figure 1. Our contributions can be summarised as:

1. We decompose the AWB problem into a novel multi-hypothesis three stage pipeline.
2. We introduce a multi-camera learning strategy that allows to leverage multi-device datasets and improve accuracy over single-camera training.
3. We provide a training-free model adaptation strategy for new cameras.
4. We report improved state-of-the-art performance on two popular public datasets (NUS [14], Cube+ [5]) and competitive results on Gehler-Shi [47, 23].

2. Related work

Classical color constancy methods utilise low-level statistics to realise various instances of the gray-world assumption: the average reflectance in a scene under a neutral light source is achromatic. Gray-World [12] and its extensions [18, 50] are based on these assumptions that tie scene reflectance statistics (e.g. mean, max reflectance) to the achromaticity of scene color.

Related assumptions define perfect reflectance [32, 20] and result in White-Patch methods. Statistical methods are fast and typically contain few free parameters, however their performance is highly dependent on strong scene content assumptions and these methods falter in cases where these assumptions fail to hold.

An early Bayesian framework [19] used Bayes’ rule to compute the posterior distribution for the illuminants and scene surfaces. They model the prior of the illuminant and the surface reflectance as a truncated multivariate normal distribution on the weights of a linear model. Other Bayesian works [44, 23], discretise the illuminant space and
model the surface reflectance priors by learning real world histogram frequencies; in [44] the prior is modelled as a uniform distribution over a subset of illuminants while [23] uses the empirical distribution of the training illuminants. Our work uses the Bayesian formulation proposed in previous works [44, 19, 23]. We estimate the likelihood probability distribution with a CNN which also explicitly learns to model the prior distribution for each illuminant.

**Fully supervised methods.** Early learning-based works [21, 53, 52] comprise combinational and direct approaches, typically relying on hand-crafted image features which limited their overall performance. Recent fully supervised convolutional color constancy work offers state-of-the-art estimation accuracy. Both local patch-based [9, 48, 10] and full image input [6, 34, 7, 25, 28] have been considered, investigating different model architectures [9, 10, 48] and the use of semantic information [28, 34, 7].

Some methods frame color constancy as a **classification problem**, e.g. CCC [6] and the follow-up refinement FFCC [7], by using a color space that identifies image re-illumination with a histogram shift. Thus, they elegantly and efficiently evaluate different illuminant candidates. Our method also discretises the illuminant space but we explicitly select the candidate illuminants, allowing for multi-camera training while FFCC [7] is constrained to use all histogram bins as candidates and single-camera training.

The method of [38] uses K-means [33] to cluster illuminants of the dataset and then applies a CNN to frame the problem as a classification task; network input is a single (pre-white balanced) image and output results in K class probabilities, representing the prospect of each illuminant (each class) explaining the correct image illumination. Our method first chooses candidate illuminants similarly, however, the key difference is that our model learns to infer whether an image is well white balanced or not. We ask this question K times by correcting the image, independently, with each illuminant candidate. This affords an independent estimation of the likelihood for each illuminant and thus enables multi-device training to improve results.

**Multi-device training** The method of [1] introduces a two CNN approach; the first network learns a 'sensor independent' linear transformation (3×3 matrix), the RGB image is transformed to this 'canonical' color space and then, a second network provides the predicted illuminant. The method is trained on multiple datasets except the test camera and obtains competitive results.

The work of [37] affords fast adaptation to previously unseen cameras, and robustness to changes in capture device by leveraging annotated samples across different cameras and datasets in a meta-learning framework.

A recent approach [8], makes an assumption that sRGB images collected from the web are well white balanced,
therefore, they apply a simple de-gamma correction to approximate an inverse tone mapping and then find achromatic pixels with a CNN to predict the illuminant. These web images were captured with unknown cameras, were processed by different ISP pipelines and might have been modified with image editing software. Despite additional assumptions, the method achieves promising results, however, not comparable with the supervised state-of-the-art.

In contrast we propose an alternative technique to enable multi-camera training and mitigate well understood sensor domain-gaps. We can train a single CNN using images captured by different cameras through the use of camera-dependent illuminant candidates. This property, of accounting for camera-dependent illuminants, affords fast model adaption; accurate inference is achievable for images captured by cameras not seen during training, if camera illuminant candidates are available (removing the need for model re-training or fine-tuning). We provide further methodological detail of these contributions and evidence towards their efficacy in Sections 3 and 4 respectively.

3. Method

Let $y = (y_r, y_g, y_b)$ be a pixel from an input image $Y$ in linear RGB space. We model the global illumination, Eq. (2), with the standard linear model [51] such that each pixel $y$ is the product of the surface reflectance $r = (r_r, r_g, r_b)$ and a global illuminant $\ell = (\ell_r, \ell_g, \ell_b)$ shared by all pixels such that:

$$ y_k = r_k \cdot \ell_k \quad k \in \{R, G, B\}. $$  

(3)

Given $Y = (y_1, \ldots, y_m)$, comprising $m$ pixels, and $R = (r_1, \ldots, r_m)$, our goal is to estimate $\ell$ and produce $R = \text{diag}(\ell)^{-1}Y$.

In order to estimate the correct illuminant to adjust the input image $Y$, we propose to frame the CC problem with a probabilistic generative model with unknown surface reflectances and illuminant. We consider a set $\ell_i \in \mathbb{R}^3$, $i \in \{1, \ldots, n\}$ of candidate illuminants, each of which are applied to $Y$ to generate a set of $n$ tentatively corrected images $\text{diag}(\ell_i)^{-1}Y$. Using the set of corrected images as inputs, we then train a CNN to identify the most probable illuminants such that the final estimated illuminant is a linear combination of the candidates. In this section, we first introduce our general Bayesian framework, followed by our proposed implementation of the main building blocks of the model. An overview of the method can be seen in Figure 2.

3.1. Bayesian approach to color constancy

Following the Bayesian formulation previously considered [44, 19, 23], we assume that the color of the light and the surface reflectance are independent. Formally $P(\ell, R) = P(\ell)P(R)$, i.e. knowledge of the surface reflectance provides us with no additional information about the illuminant, $P(\ell | R) = P(\ell)$. Based on this assumption we decompose these factors and model them separately.

Using Bayes’ rule, we define the posterior distribution of $\ell$ illuminants given the input image $Y$ as:

$$ P(\ell | Y) = \frac{P(Y | \ell)P(\ell)}{P(Y)}. $$  

(4)

We model the likelihood of an observed image $Y$ for a given illuminant $\ell$:

$$ P(Y | \ell) = \int_{R} P(Y | \ell, R = r)P(R = r) dr $$  

(5)

where $R$ are the surface reflectances and $\text{diag}(\ell)^{-1}Y$ is the image as corrected with illuminant $\ell$. The term $P(Y | \ell, R = r)$ is only non-zero for $R = \text{diag}(\ell)^{-1}Y$. The likelihood rates whether a corrected image looks realistic.

We choose to instantiate the model of our likelihood using a shallow CNN. The network should learn to output a high likelihood if the reflectances look realistic. We model the prior probability $P(\ell)$ for each candidate illuminant independently as learnable parameters in an end-to-end approach; this effectively acts as a regularisation, favouring more likely real-world illuminants. We note that, in practice, the function modelling the prior also depends on factors such as the environment (indoor / outdoor), the time of day, ISO etc. However, the size of currently available datasets prevent us from modelling more complex proxies.

In order to estimate the illuminant $\ell^*$, we optimise the quadratic cost (minimum MSE Bayesian estimator), minimised by the mean of the posterior distribution:

$$ \ell^* = \int_{\ell} \ell \cdot P(\ell | Y) d\ell $$  

(6)

This is done in the following three steps (c.f. Figure 2):

1. **Candidate selection** (Section 3.2): Choose a set of $n$ illuminant candidates to generate $n$ corrected thumbnail $(64 \times 64)$ images.

2. **Likelihood estimation** (Section 3.3): Evaluate these $n$ images independently with a CNN, a network designed to estimate the likelihood that an image is well white balanced $P(Y^* | \ell)$.

3. **Illuminant determination** (Section 3.4): Compute the posterior probability of each candidate illuminant and determine a final illuminant estimation $\ell^*$.

This formulation allows estimation of a posterior probability distribution, allowing us to reason about a set of
probable illuminants rather than produce a single illuminant point estimate (c.f. regression approaches). Regression typically does not provide feedback on a possible set of alternative solutions which has shown to be of high value in alternative vision problems [35].

The second benefit that our decomposition affords is a principled multi-camera training process. A single, device agnostic CNN estimates illuminant likelihoods and performs independent selection of candidate illuminants for each camera. By leveraging image information across multiple datasets we increase model robustness. Additionally, the amalgamation of small available CC datasets provides a step towards harnessing the power of large capacity models for this problem domain c.f. contemporary models.

3.2. Candidate selection

The goal of candidate selection is to discretise the illuminant space of a specific camera in order to obtain a set of representative illuminants (spanning the illuminant space). Given a collection of ground truth illuminants, measured from images containing calibration objects (i.e. a labelled training set), we compute candidates using K-means clustering [33] on the linear RGB space.

By forming n clusters of our measured illuminants, we define the set of candidates \( \ell_i \in \mathbb{R}^3, i \in \{1, \ldots, n\} \) as the cluster centers. K-means illuminant clustering is previously shown to be effective for color constancy [38] however we additionally evaluate alternative candidate selection strategies (detailed in the supplementary material); our experimental investigation confirms a simple K-means approach provides strong target task performance. Further, the effect of \( K \) is empirically evaluated in Section 4.4.

Image \( Y \), captured by a given camera, is then used to produce a set of images, corrected using the illuminant candidate set for the camera, on which we evaluate the accuracy of each candidate.

3.3. Likelihood estimation

We model the likelihood estimation step using a neural network which, for a given illuminant \( \ell \) and image \( Y \), takes the tentatively corrected image \( \text{diag}(\ell)^{-1}Y \) as input, and learns to predict the likelihood \( P(Y|\ell) \) that the image has been well white balanced i.e. has an appearance of being captured under an achromatic light source.

The success of low capacity histogram based methods [6, 7] and the inference-training tradeoff for small datasets motivate a compact network design. We propose a small CNN with one spatial convolution and subsequent layers constituting \( 1 \times 1 \) convolutions with spatial pooling. Lastly, three fully connected layers gradually reduce the dimensionality to one (see supplementary material for architecture details). Our network output is then a single value that represents the log-likelihood that the image is well white balanced:

\[
\log(P(Y|\ell)) = f^W(\text{diag}(\ell)^{-1}Y). \tag{7}
\]

Function \( f^W \) is our trained CNN parametrised by model weights \( W \). Eq. (7) estimates the log-likelihood of each candidate illuminant separately. It is important to note that we only train a single CNN which is used to estimate the likelihood for each candidate illuminant independently. However, in practice, certain candidate illuminants will be more common than others. To account for this, following [7], we compute an affine transformation of our log-likelihood \( \log(P(Y|\ell)) \) by introducing learnable, illuminant specific, gain \( G_\ell \) and bias \( B_\ell \) parameters. Gain \( G_\ell \) affords amplification of illuminant likelihoods. The bias term \( B_\ell \) learns to prefer some illuminants i.e. a prior distribution in a Bayesian sense: \( B_\ell = \log(P(\ell)) \). The log-posterior probability can then be formulated as:

\[
\log(P(\ell|Y)) = G_\ell \cdot \log(P(Y|\ell)) + B_\ell. \tag{8}
\]

We highlight that learned affine transformation parameters are training camera-dependent and provide further discussion on camera agnostic considerations in Section 3.5.

3.4. Illuminant determination

We require a differentiable method in order to train our model end-to-end, and therefore the use of a simple Maximum a Posteriori (MAP) inference strategy is not possible. Therefore to estimate the illuminant \( \ell^* \), we use the minimum mean square error Bayesian estimator, which is minimised by the posterior mean of \( \ell \) (c.f. Eq. (6)):

\[
\ell^* = \sum_{i=1}^{n} \ell_i \cdot \text{softmax}(\log(P(\ell_i|Y)))
\]

\[
= \frac{1}{\sum_{i=1}^{n} e^{\log(P(\ell_i|Y))}} \sum_{i=1}^{n} \ell_i \cdot e^{\log(P(\ell_i|Y))}. \tag{9}
\]

The resulting vector \( \ell^* \) is \( l_2 \)-normalised. We leverage our K-means centroid representation of the linear RGB space and use linear interpolation within the convex hull of feasible illuminants to determine the estimated scene illuminant \( \ell^* \). For Eq. (9), we take inspiration from [29, 38], who have successfully explored similar strategies in CC and stereo regression, e.g. [29] introduced an analogous \( \text{soft-argmin} \) to estimate disparity values from a set of candidates. We apply a similar strategy for illuminant estimation and use the \( \text{soft-argmax} \) which provides a linear combination of all candidates weighted by their probabilities.

We train our network end-to-end with the commonly used angular error loss function, where \( \ell^* \) and \( \ell^{GT} \) are the prediction and ground truth illuminant, respectively:
is device-independent, we fix previously learnable parameters during model training. In order to ensure that our CNN specific candidates yet update a single set of CNN parameters during model training. Use camera-specific candidates, yet learn only a single model. Specifically, we hypothesize experimentally in Section 4.

Our CNN learns to produce the likelihood that an input image is well white balanced. We claim that framing part of the CC problem in this fashion results in a device-independent learning task. We evaluate the benefit of this hypothesis experimentally in Section 4.

To train with multiple cameras we use camera-specific candidates, yet learn only a single model. Specifically, we train with a different camera for each batch, use camera-specific candidates yet update a single set of CNN parameters during model training. In order to ensure that our CNN is device-independent, we fix previously learnable parameters that depend on sensor specific illuminants, i.e. \( B_k = 0 \) and \( G_k = 1 \). The absence of these parameters, learned in a camera-dependent fashion, intuitively restricts model flexibility however we observe this drawback to be compensated by the resulting ability to train using amalgamated multi-camera datasets i.e. more data. This strategy allows our CNN to be camera-agnostic and affords the option to refine existing CNN quality when data from novel cameras becomes available. We however clarify that our overarching strategy for white balancing maintains use of camera-specific candidate illuminants.

4. Results

4.1. Training details

We train our models for 120 epochs and use \( K \)-mean [33] with \( K = 120 \) candidates. Our batch size is 32, we use the Adam optimiser [30] with initial learning rate \( 5 \times 10^{-3} \), divided by two after 10, 50 and 80 epochs. Dropout [27] of 50% is applied after average pooling. We take the log transform of the input before the first convolution. Efficient inference is feasible by concatenating each candidate corrected image into the batch dimension. We use PyTorch 1.0 [39] and an Nvidia Tesla V100 for our experiments. The first layer is the only spatial convolution, it is adapted from [49] and pretrained on ImageNet [16]. We fix the weights of this first layer to avoid over-fitting. The total amount of weights is \( 22.8K \). For all experiments calibration objects are masked, black level subtracted and over-saturated pixels are clipped at 95% threshold. We resize the image to \( 64 \times 64 \) and normalise.

4.2. Datasets

We experiment using three public datasets. The Gehler-Shi dataset [47, 23] contains 568 images of indoor and outdoor scenes. Images were captured using Canon 1D and Canon 5D cameras. We highlight our awareness of the existence of multiple sets of non-identical ground-truth labels for this dataset (see [26] for further detail). Our Gehler-Shi evaluation is conducted using the SFU ground-truth labels [47] (consistent with the label naming convention in [26]). The NUS dataset [14] originally consists of 8 subsets of \( \sim 210 \) images per camera providing a total of \( 1736 \) images. The Cube+ dataset [5] contains 1707 images captured with Canon 550D camera, consisting of predominantly outdoor imagery.

For the NUS [14] and Gehler-Shi [47, 23] datasets we perform three-fold cross validation (CV) using the splits provided in previous work [7, 6]. The Cube+ [5] dataset does not provide splits for CV so we use all images for learning and evaluate using a related set of test images, provided for the recent Cube+ ISPA 2019 challenge [31]. We compare with the results from the challenge leader-board.

For the NUS dataset [14], we additionally explore training multi-camera models and thus create a new set of CV folds to facilitate this. We are careful to highlight that the NUS dataset consists of eight image subsets, pertaining to eight capture devices. Each of our new folds captures a distinct set of scene content (i.e. sets of up to eight similar images for each captured scene). This avoids testing on similar scene content seen during training. We define our multi-camera CV such that multi-camera fold \( i \) is the concatenation of images, pertaining to common scenes, captured from all eight cameras. The folds that we define are made available in our supplementary material.

4.3. Evaluation metrics

We use the standard angular error metric for quantitative evaluation (c.f. Eq. (10)). We report standard CC statistics to summarise results over the investigated datasets: Mean, Median, Trimean, Best 25%, Worst 25%. We further report method inference time in the supplementary material. Other works’ results were taken from corresponding papers, resulting in missing statistics for some methods. The NUS [14] dataset is composed of 8 cameras, we report the geometric mean of each statistic for each method across all cameras as standard in the literature [7, 6, 28].

4.4. Quantitative evaluation

Accuracy experiments. We report competitive results on the dataset of Gehler-Shi [47, 23] (c.f. Table 1). This dataset

\[
\mathcal{L}_{error} = \arccos \left( \frac{G^T \cdot \ell^*}{\|G^*\| \|\ell^*\|} \right) \tag{10}
\]
Table 1. Angular error statistics for Gehler-Shi dataset [47, 23].

<table>
<thead>
<tr>
<th>Method</th>
<th>Mean</th>
<th>Med.</th>
<th>Tri.</th>
<th>Best 25%</th>
<th>Worst 25%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gray-world [12]</td>
<td>6.28</td>
<td>6.28</td>
<td>2.33</td>
<td>10.58</td>
<td></td>
</tr>
<tr>
<td>Bayesian [23]</td>
<td>4.82</td>
<td>3.46</td>
<td>3.88</td>
<td>1.26</td>
<td>10.19</td>
</tr>
<tr>
<td>Quasi-unsupervised [8]</td>
<td>2.91</td>
<td>1.98</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Afifi et al. [19]</td>
<td>2.77</td>
<td>1.93</td>
<td>-</td>
<td>0.55</td>
<td>6.53</td>
</tr>
<tr>
<td>Meta-AWB [37]</td>
<td>2.57</td>
<td>1.84</td>
<td>1.94</td>
<td>0.47</td>
<td>6.11</td>
</tr>
<tr>
<td>Cheng et al. [2015]</td>
<td>2.42</td>
<td>1.65</td>
<td>1.75</td>
<td>0.38</td>
<td>5.87</td>
</tr>
<tr>
<td>CM 2019 [25]</td>
<td>2.48</td>
<td>1.61</td>
<td>1.60</td>
<td>0.37</td>
<td>12.01</td>
</tr>
<tr>
<td>Ob et al. [38]</td>
<td>2.16</td>
<td>1.47</td>
<td>1.61</td>
<td>0.37</td>
<td>5.12</td>
</tr>
<tr>
<td>CCC [6]</td>
<td>1.95</td>
<td>1.22</td>
<td>1.38</td>
<td>0.35</td>
<td>4.76</td>
</tr>
<tr>
<td>DS-Net [48]</td>
<td>1.90</td>
<td>1.12</td>
<td>1.33</td>
<td>0.31</td>
<td>4.84</td>
</tr>
<tr>
<td>FC4 [28] (SqueezeNet)</td>
<td>1.65</td>
<td>1.18</td>
<td>1.27</td>
<td>0.38</td>
<td>3.78</td>
</tr>
<tr>
<td>FC4 [28] (AlexNet)</td>
<td>1.77</td>
<td>1.12</td>
<td>1.29</td>
<td>0.34</td>
<td>4.29</td>
</tr>
<tr>
<td>FFCC [7] (model P)</td>
<td>1.61</td>
<td>0.86</td>
<td>1.02</td>
<td>0.23</td>
<td>4.27</td>
</tr>
<tr>
<td>Ours</td>
<td>2.53</td>
<td>1.34</td>
<td>1.63</td>
<td>0.40</td>
<td>5.80</td>
</tr>
<tr>
<td>Ours (pretrained)</td>
<td>2.10</td>
<td>1.32</td>
<td>1.53</td>
<td>0.36</td>
<td>5.10</td>
</tr>
</tbody>
</table>

Table 1. Angular error statistics for Gehler-Shi dataset [47, 23].

can be considered very challenging as the number of images per camera is imbalanced: There are 86 Canon 1D and 482 Canon 5D images. Our method is not able to outperform the state-of-the-art likely due to the imbalanced nature and small size of Canon 1D. Pretraining on a combination of NUS [14] and Cube+ [5] provides moderate accuracy improvement despite the fact that the Gehler-Shi dataset has a significantly different illuminant distribution compared to those seen during pre-training. We provide additional experiments, exploring the effect of varying $K$, for $K$-means candidate selection in the supplementary material.

Results for NUS [14] are provided in Table 2. Our method obtains competitive accuracy and the previously observed trend, pre-training using additional datasets (here Gehler-Shi [47, 23] and Cube+ [5]), again improves results.

In Table 3, we report results for our multi-device setting on the NUS [14] dataset. For this experiment we introduce a new set of training folds to ensure that scenes are well separated and refer to Sections 3.5 for multi-device training and 4.2 for related training folds detail. We draw multi-device comparison with FFCC [7], by choosing to center the FFCC histogram with the training set (of amalgamated camera datasets). Note that results are not directly comparable with Table 2 due to our redefinition of CV folds. Our method is more accurate than the state-of-the-art when training considers all available cameras at the same time. Note that multi-device training improves the median angular error of each individual camera dataset (we provide results in the supplementary material). Overall performance is improved by $\sim$11\% in terms of median accuracy.

We also outperform the state-of-the-art on the recent Cube challenge [31] as shown in Table 4. Pretraining together on Gehler-Shi [47, 23] and NUS [14] improves our Mean and Worst 95\% statistics.

In summary, we observe strong generalisation when using multiple camera training (e.g. NUS [14] results c.f. Tables 2 and 3). These experiments illustrate the large benefit achievable with multi-camera training when illuminant distributions of the cameras are broadly consistent. Gehler-Shi [47, 23] has a very disparate illuminant distribution with respect to alternative datasets and we are likely unable to exploit the full advantage of multi-camera training. We note the FFCC [7] state of the art method is extremely shallow and therefore optimised for small datasets. In contrast, when our model is trained on large and relevant datasets we are able to achieve superior results.

**Run time.** Regarding run-time; we measure inference speed at $\sim$10 milliseconds, implemented in unoptimised PyTorch (see supplementary material for further detail).
4.5. Training on novel sensors

To explore camera agnostic elements of our model, we train on a combination of the full NUS [14] and Gehler-Shi [47, 23] datasets. As described in Section 3.5, the only remaining device dependent component involves performing illuminant candidate selection per device. Once the model is trained, we select candidates from Cube+ [5] and test on the Cube challenge dataset [31]. We highlight that neither Cube+ nor Cube challenge imagery is seen during model training. For meaningful evaluation, we compare against both classical and recent learning-based [1] camera-agnostic methods. Results are shown in Table 5. We obtain results that are comparable to Table 4 without seeing any imagery from our target camera, outperforming both baselines and [1]. We clarify that our method performs candidate selection using Cube+ [5] to adapt the candidate set to the novel device while [1] does not see any information from the new camera.

We provide additional experimental results for differing values of $K$ ($K$-means candidate selection) in the supplementary material. We observe stability for $K \geq 25$. The low number of candidates required is likely linked to the two Cube datasets having reasonably compact distributions.

4.6. Qualitative evaluation

We provide visual results for the Gehler-Shi [47, 23] dataset in Figure 3. We sort inference results by increasing angular error and sample 5 images uniformly. For each row, we show (a) the input image (b) our estimated illuminant color and resulting white-balanced image (c) the ground truth illuminant color and resulting white-balanced image. Images are first white-balanced, then, we apply an estimated CCM (Color Correction Matrix), and finally, sRGB gamma correction. We mask out the Macbeth Color Checker calibration object during both training and evaluation.

Our most challenging example (c.f. last row of Figure 3) is a multi-illuminant scene (indoor and outdoor lights), we observe our method performs accurate correction for objects illuminated by the outdoor light, yet the ground truth is only measured for the indoor illuminant, hence the high angular error. This highlights the limitation linked to our single global illuminant assumption, common to the majority of CC algorithms. We show additional qualitative results in the supplementary material.

5. Conclusion

We propose a novel multi-hypothesis color constancy model capable of effectively learning from image samples that were captured by multiple cameras. We frame the problem under a Bayesian formulation and obtain data-driven likelihood estimates by learning to classify achromatic imagery. We highlight the challenging nature of multi-device learning due to camera color space differences, spectral sensitivity and physical sensor effects. We validate the benefits of our proposed solution for multi-device learning and provide state-of-the-art results on two popular color constancy datasets while maintaining real-time inference constraints. We additionally provide evidence supporting our claims that framing the learning question as a classification task c.f. regression can lead to strong performance without requiring model re-training or fine-tuning.

![Figure 3](image-url)

Figure 3. Example results taken from the Gehler-Shi [47, 23] dataset. Input, our result and ground truth per row. Images to visualise are chosen by sorting all test images using increasing error and evenly sampling images according to that ordering. Images are rendered in sRGB color space.
References


[22] 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.


