A Multi-Hypothesis Approach to Color Constancy

Daniel Hernandez-Juarez[1], Sarah Parisot[1,2], Benjamin Busam[1,3], Aleš Leonardis[1]
Gregory Slabaugh[1], Steven McDonagh[1]

dhernandez0@gmail.com,
{sarah.parisot, benjamin.busam, ales.leonardis, gregory.slabaugh, steven.mcdonagh} @huawei.com

[1]Huawei Noah's Ark Lab [2]Mila, Montréal [3]Technical 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.

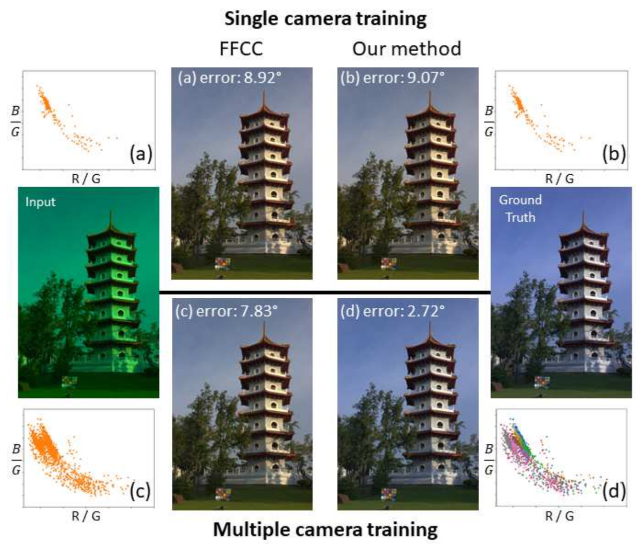

Figure 1. Our multi-hypothesis strategy allows us to leverage multi-camera datasets. Example image taken from the NUS dataset [14]. Single camera training: (a) state of the art method FFCC [7] and (b) our method obtains similar angular-error. Training with all 8 dataset cameras: aggregate all images to (c) define FFCC histogram center and (d) use an illuminant candidate set per camera. $[\frac{r}{g}, \frac{b}{g}]$ color space plots show training set illuminant distributions. Each camera is encoded with a different color in (d) to highlight camera-specific illuminants. Our model leverages the extra data to achieve lower angular error. Images are rendered in sRGB color space.

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 physical model of Lambertian image formation captured under a trichromatic photosensor:

$$\rho_k(X) = \int_\Omega E(\lambda)S(\lambda, X)C_k(\lambda)d\lambda \quad k \in \{R, G, B\}. \tag{1}$$

where $\rho_k(X)$ is the intensity of color channel k at pixel location X, λ 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 Ω.

The goal of computational CC then becomes estimation of the global illumination color ρ_k^E where:

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

Finding ρ_k^E in Eq. (2) results in a 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

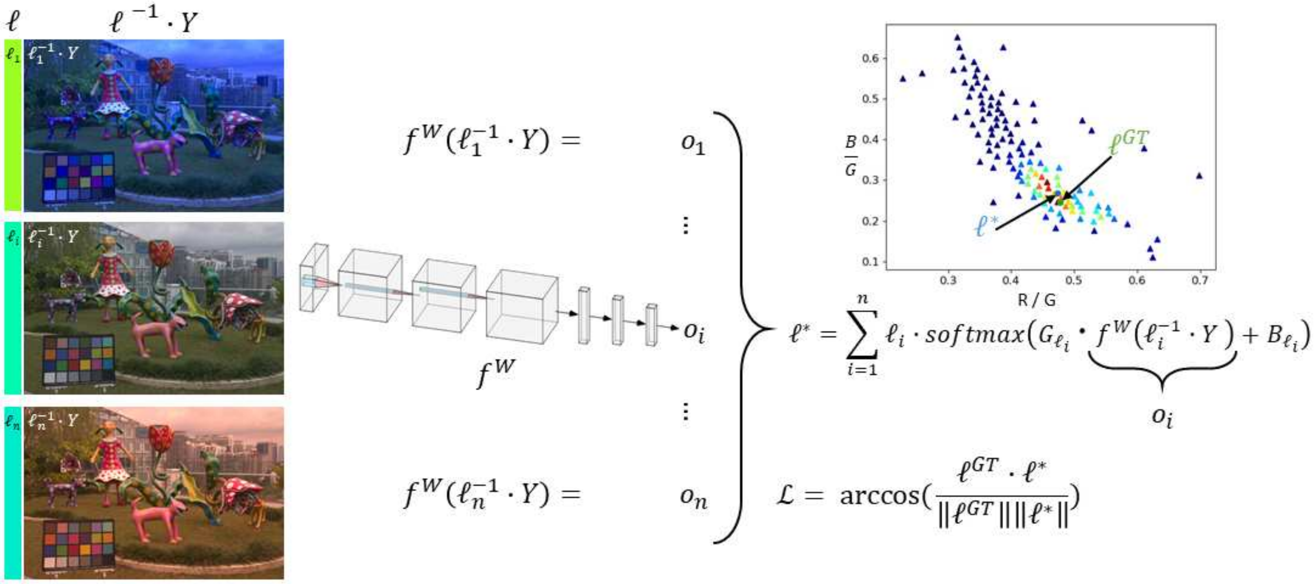

Figure 2. Method overview: we first generate a list of n candidate illuminants $\boldsymbol{\ell}_i$ (candidate illuminants are shown left of the respective corrected images) using K-means clustering [33]. We correct the input image with each of the n candidates independently and then estimate the likelihood o_i of each corrected image with our network. We combine illuminant candidates using the posterior probability distribution to generate an illuminant estimation $\boldsymbol{\ell}^*$. The error is back-propagated through the network using angular error loss $\mathcal{L}$. The [$\frac{r}{g}$, $\frac{b}{g}$] plot in the upper-right illustrates the posterior probability distribution (triangles encoded from blue to red) of the candidates $\boldsymbol{\ell}_i$, the final prediction vector $\boldsymbol{\ell}^*$ (blue circle) and the ground-truth illuminant $\boldsymbol{\ell}^{GT}$ (green circle). Images are rendered in sRGB color space.

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 $\mathbf{y} = (y_r, y_g, y_b)$ 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 $\mathbf{y}$ is the product of the surface reflectance $\mathbf{r} = (r_r, r_g, r_b)$ and a global illuminant $\boldsymbol{\ell} = (\ell_r, \ell_g, \ell_b)$ shared by all pixels such that:

$$y_k = r_k \cdot \ell_k \qquad k \in \{R, G, B\}. \tag{3}$$

Given $Y = (\mathbf{y}_1, \dots, \mathbf{y}_m)$, comprising m pixels, and $R = (\mathbf{r}_1, \dots, \mathbf{r}_m)$, our goal is to estimate $\boldsymbol{\ell}$ and produce $R = \operatorname{diag}(\boldsymbol{\ell})^{-1} Y$.

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 $\boldsymbol{\ell}_i \in \mathbb{R}^3, i \in \{1, \dots, n\}$ of candidate illuminants, each of which are applied to Y to generate a set of n tentatively corrected images $\operatorname{diag}(\boldsymbol{\ell}_i)^{-1} Y$. 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 $\mathrm{P}(\boldsymbol{\ell}, R) = \mathrm{P}(\boldsymbol{\ell})\, \mathrm{P}(R)$, *i.e.* knowledge of the surface reflectance provides us with no additional information about the illuminant, $\mathrm{P}(\,\boldsymbol{\ell} \mid R) = \mathrm{P}(\boldsymbol{\ell})$. Based on this assumption we decompose these factors and model them separately.

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

$$\mathrm{P}(\,\boldsymbol{\ell} \mid Y) = \frac{\mathrm{P}(\,Y \mid \boldsymbol{\ell}\,)\, \mathrm{P}(\boldsymbol{\ell})}{\mathrm{P}(Y)}. \tag{4}$$

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

$$\begin{aligned} \mathrm{P}(\,Y \mid \boldsymbol{\ell}\,) &= \int_r \mathrm{P}(\,Y \mid \boldsymbol{\ell}, R = r)\, \mathrm{P}(R = r)\, dr \\ &= \mathrm{P}(R = \operatorname{diag}(\boldsymbol{\ell})^{-1} Y) \end{aligned} \tag{5}$$

where R are the surface reflectances and $\operatorname{diag}(\boldsymbol{\ell})^{-1} Y$ is the image as corrected with illuminant $\boldsymbol{\ell}$. The term $\mathrm{P}(\,Y \mid \boldsymbol{\ell}, R = r)$ is only non-zero for $R = \operatorname{diag}(\boldsymbol{\ell})^{-1} Y$. 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 $\mathrm{P}(\boldsymbol{\ell})$ 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 $\boldsymbol{\ell}^*$, we optimise the quadratic cost (minimum MSE Bayesian estimator), minimised by the mean of the posterior distribution:

$$\boldsymbol{\ell}^* = \int_{\boldsymbol{\ell}} \boldsymbol{\ell} \cdot \mathrm{P}(\,\boldsymbol{\ell} \mid Y)\, d\boldsymbol{\ell} \tag{6}$$

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 (64×64) 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 $\mathrm{P}(Y \mid \boldsymbol{\ell})$.

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

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 $\boldsymbol{\ell}_i \in \mathbb{R}^3, i \in \{1, \ldots, n\}$ 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 $\boldsymbol{\ell}$ and image Y, takes the tentatively corrected image $\mathrm{diag}(\boldsymbol{\ell})^{-1}Y$ as input, and learns to predict the likelihood $P(Y|\boldsymbol{\ell})$ 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 1×1 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:

$$\log(\mathrm{P}(\,Y \mid \boldsymbol{\ell}\,)) = f^W(\mathrm{diag}(\boldsymbol{\ell})^{-1}Y\,). \tag{7}$$

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 $\log(\mathrm{P}(\,Y \mid \boldsymbol{\ell}\,))$ by introducing learnable, illuminant specific, gain $G_{\boldsymbol{\ell}}$ and bias $B_{\boldsymbol{\ell}}$ parameters. Gain G_l affords amplification of illuminant likelihoods. The bias term $B_{\boldsymbol{\ell}}$ learns to prefer some illuminants *i.e*. a prior distribution in a Bayesian sense: $B_{\boldsymbol{\ell}} = \log(\mathrm{P}(\boldsymbol{\ell}))$. The log-posterior probability can then be formulated as:

$$\log(\mathrm{P}(\,\boldsymbol{\ell} \mid Y\,)) = G_{\boldsymbol{\ell}} \cdot \log(\mathrm{P}(\,Y \mid \boldsymbol{\ell}\,)) + B_{\boldsymbol{\ell}}. \tag{8}$$

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 $\boldsymbol{\ell}^*$, we use the minimum mean square error Bayesian estimator, which is minimised by the posterior mean of $\boldsymbol{\ell}$ (*c.f*. Eq. (6)):

$$\begin{aligned}\boldsymbol{\ell}^* &= \sum_{i=1}^{n} \boldsymbol{\ell}_i \cdot \mathrm{softmax}(\log(\mathrm{P}(\boldsymbol{\ell}_i \mid Y))) \\ &= \frac{1}{\sum e^{\log(\mathrm{P}(\boldsymbol{\ell}_i|Y))}} \sum_{i=1}^{n} \boldsymbol{\ell}_i \cdot e^{\log(\mathrm{P}(\boldsymbol{\ell}_i|Y))}.\end{aligned} \tag{9}$$

The resulting vector $\boldsymbol{\ell}^*$ 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 $\boldsymbol{\ell}^*$. 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 *soft-argmin* to estimate disparity values from a set of candidates. We apply a similar strategy for illuminant estimation and use the *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 $\boldsymbol{\ell}^*$ and $\boldsymbol{\ell}^{GT}$ are the prediction and ground truth illuminant, respectively:

$$\mathcal{L}_{error} = \arccos\left(\frac{\boldsymbol{\ell}^{GT} \cdot \boldsymbol{\ell}^{*}}{\|\boldsymbol{\ell}^{GT}\|\|\boldsymbol{\ell}^{*}\|}\right) \tag{10}$$

3.5. Multi-device training

As discussed in previous work [1, 41, 22], CC models typically fail to train successfully using multiple camera data due to distribution shifts between camera sensors, making them intrinsically device-dependent and limiting model capacity. A device-independent model is highly appealing due to the small number of images commonly available in camera-specific public color constancy datasets. The cost and time associated with collecting and labelling new large data for specific novel devices is expensive and prohibitive.

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.* $B_{\boldsymbol{\ell}} = 0$ and $G_{\boldsymbol{\ell}} = 1$. 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 5×10^{-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 oversaturated pixels are clipped at 95% threshold. We resize the image to 64×64 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

Method	Mean	Med.	Tri.	Best 25%	Worst 25%
Gray-world [12]	6.36	6.28	6.28	2.33	10.58
White-Patch [11]	7.55	5.86	6.35	1.45	16.12
Bayesian [23]	4.82	3.46	3.88	1.26	10.49
Quasi-unsupervised [8]	2.91	1.98	-	-	-
Afifi *et al*. 2019 [1]	2.77	1.93	-	0.55	6.53
Meta-AWB [37]	2.57	1.84	1.94	0.47	6.11
Cheng *et al*. 2015 [15]	2.42	1.65	1.75	0.38	5.87
CM 2019 [25]	2.48	1.61	1.80	0.47	5.97
Oh *et al*. [38]	2.16	1.47	1.61	0.37	5.12
CCC [6]	1.95	1.22	1.38	0.35	4.76
DS-Net [48]	1.90	1.12	1.33	0.31	4.84
FC4 [28] (SqueezeNet)	1.65	1.18	1.27	0.38	**3.78**
FC4 [28] (AlexNet)	1.77	1.11	1.29	0.34	4.29
FFCC [7] (model P)	**1.61**	**0.86**	**1.02**	**0.23**	4.27
Ours	2.35	1.43	1.63	0.40	5.80
Ours (pretrained)	2.10	1.32	1.53	0.36	5.10

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

Method	Mean	Med.	Tri.	Best 25%	Worst 25%
White-patch [11]	9.91	7.44	8.78	1.44	21.27
Gray-world [12]	4.59	3.46	3.81	1.16	9.85
Bayesian [23]	3.50	2.36	2.57	0.78	8.02
Oh *et al*. [38]	2.36	2.09	-	-	**4.16**
Quasi-unsupervised [8]	1.97	1.91	-	-	-
CM 2019 [25]	2.25	1.59	1.74	0.50	5.13
FC4 [28] (SqueezeNet)	2.23	1.57	1.72	0.47	5.15
FC4 [28] (AlexNet)	2.12	1.53	1.67	0.48	4.78
Afifi *et al*. 2019 [1]	2.05	1.50	-	0.52	4.48
CCC [6]	2.38	1.48	1.69	0.45	5.85
Cheng *et al*. 2015 [15]	2.18	1.48	1.64	0.46	5.03
DS-Net [48]	2.21	1.46	1.68	0.48	6.08
Meta-AWB [37]	**1.89**	1.34	1.44	0.45	4.28
FFCC [7] (model Q)	2.06	1.39	1.53	0.39	4.80
FFCC [7] (model M)	1.99	**1.31**	**1.43**	**0.35**	4.75
Ours	2.39	1.61	1.74	0.50	5.67
Ours (pretrained)	2.35	1.55	1.73	0.46	5.62

Table 2. Angular error statistics for NUS [14].

Method	Mean	Med.	Tri.	Best 25%	Worst 25%
One model per device					
FFCC [7] (model Q)	2.37	1.50	1.69	0.46	5.76
Ours (pretrained)	2.35	1.48	1.67	0.47	5.71
Multi-device training					
FFCC [7] (model Q)	2.59	1.77	1.94	0.52	6.14
Ours (pretrained)	2.22	1.33	1.53	0.44	5.49

Table 3. Angular error statistics for NUS [14] using multi-device cross-validation folds (see Section 4.2). FFCC *model Q* is considered for fair comparison (thumbnail resolution input).

Method	Mean	Med.	Tri.	Best 25%	Worst 25%
Gray-world [12]	4.44	3.50	-	0.77	9.64
1st-order Gray-Edge [50]	3.51	2.30	-	0.56	8.53
V Vuk *et al*. [31]	6.00	1.96	2.25	0.99	18.81
Y Qian *et al*. [31]	2.21	1.32	1.41	0.43	5.65
K Chen *et al*. [31]	1.84	1.27	1.32	0.39	**4.41**
Y Qian *et al*. [40]	2.27	1.26	1.35	0.39	6.02
Afifi *et al*. 2019 [1]	2.10	1.23	-	0.47	5.38
FFCC [7] (model J)	2.10	1.23	1.34	0.47	5.38
A Savchik *et al*. [46]	2.05	1.20	1.30	0.40	5.24
WB-sRGB [3, 1]	**1.83**	1.15	-	**0.35**	4.60
Ours	1.99	**1.06**	**1.14**	**0.35**	5.35
Ours (pretrained)	1.95	1.16	1.25	0.39	4.99

Table 4. Angular error for Cube challenge [31].

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 $K >= 25$. 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.

Method	Mean	Med.	Tri.	Best 25%	Worst 25%
Gray-world [12]	4.44	3.50	-	0.77	9.64
1st-order Gray-Edge [50]	3.51	2.30	-	0.56	8.53
Afifi *et al*. 2019 [1]	2.89	1.72	-	0.71	7.06
Ours	**2.07**	**1.31**	1.43	**0.41**	**5.12**

Table 5. Angular error for the Cube challenge [31] trained solely on the dataset of NUS [14] and Gehler-Shi [47, 23]. For our method, candidate selection is performed on Cube+ [5] dataset.

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.

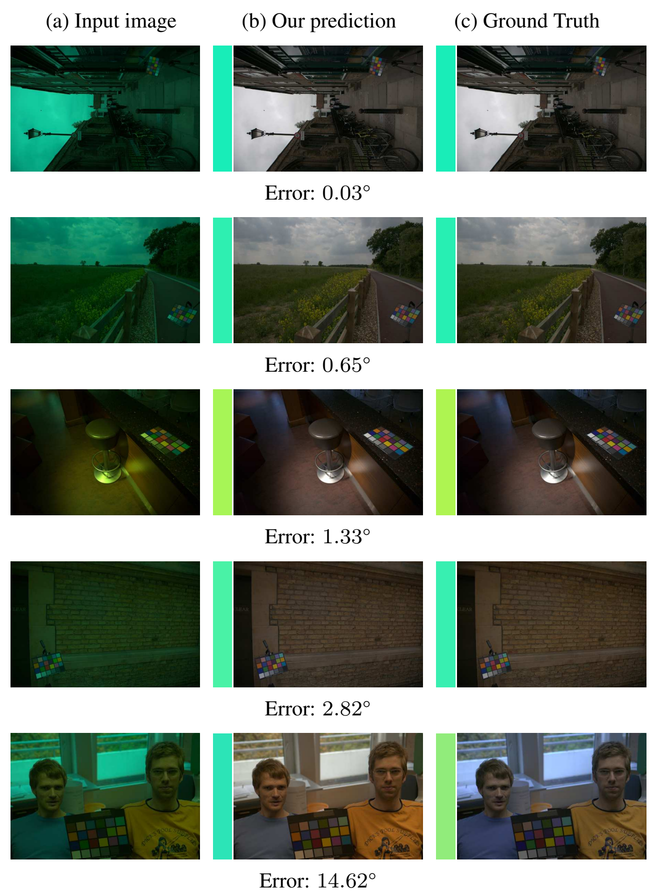

Figure 3. Example results taken from the Gehler-Shi [47, 23] dataset. Input, our result and ground truth per row. Images to visualise are chosen by sorting all test images using increasing error and evenly sampling images according to that ordering. Images are rendered in sRGB color space.

References

[1] Mahmoud Afifi and Michael Brown. Sensor-Independent Illumination Estimation for DNN Models. In *Proceedings of the British Machine Vision Conference 2019, BMVC 2019, Cardiff University, Cardiff, UK, September 9-12, 2019*, 2019.

[2] Mahmoud Afifi and Michael S. Brown. What else can fool deep learning? addressing color constancy errors on deep neural network performance. In *2019 IEEE International Conference on Computer Vision, ICCV 2019, Seoul, Korea, October 29-November 1, 2019*, 2019.

[3] Mahmoud Afifi, Brian L. Price, Scott Cohen, and Michael S. Brown. When color constancy goes wrong: Correcting improperly white-balanced images. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 1535–1544, 2019.

[4] Alexander Andreopoulos and John K. Tsotsos. On sensor bias in experimental methods for comparing interest-point, saliency, and recognition algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(1):110–126, 2012.

[5] Nikola Banic and Sven Loncaric. Unsupervised learning for color constancy. In *Proceedings of the 13th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2018) - Volume 4: VISAPP, Funchal, Madeira, Portugal, January 27-29, 2018*, pages 181–188, 2018.

[6] Jonathan T. Barron. Convolutional color constancy. In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, pages 379–387, 2015.

[7] Jonathan T. Barron and Yun-Ta Tsai. Fast fourier color constancy. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 6950–6958, 2017.

[8] Simone Bianco and Claudio Cusano. Quasi-unsupervised color constancy. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 12212–12221, 2019.

[9] Simone Bianco, Claudio Cusano, and Raimondo Schettini. Color constancy using cnns. In *2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2015, Boston, MA, USA, June 7-12, 2015*, pages 81–89, 2015.

[10] Simone Bianco, Claudio Cusano, and Raimondo Schettini. Single and multiple illuminant estimation using convolutional neural networks. *IEEE Transactions on Image Processing*, 26(9):4347–4362, 2017.

[11] David H Brainard and Brian A Wandell. Analysis of the retinex theory of color vision. *JOSA A*, 3(10):1651–1661, 1986.

[12] Gershon Buchsbaum. A spatial processor model for object colour perception. *Journal of the Franklin institute*, 310(1):1–26, 1980.

[13] Alexandra Carlson, Katherine A. Skinner, and Matthew Johnson-Roberson. Modeling camera effects to improve deep vision for real and synthetic data. *CoRR*, abs/1803.07721, 2018.

[14] Dongliang Cheng, Dilip K Prasad, and Michael S Brown. Illuminant estimation for color constancy: why spatial-domain methods work and the role of the color distribution. *JOSA A*, 31(5):1049–1058, 2014.

[15] Dongliang Cheng, Brian L. Price, Scott Cohen, and Michael S. Brown. Effective learning-based illuminant estimation using simple features. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pages 1000–1008, 2015.

[16] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Fei-Fei Li. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20-25 June 2009, Miami, Florida, USA*, pages 248–255, 2009.

[17] Steven Diamond, Vincent Sitzmann, Stephen P. Boyd, Gordon Wetzstein, and Felix Heide. Dirty pixels: Optimizing image classification architectures for raw sensor data. *CoRR*, abs/1701.06487, 2017.

[18] Graham D. Finlayson and Elisabetta Trezzi. Shades of gray and colour constancy. In *The Twelfth Color Imaging Conference: Color Science and Engineering Systems, Technologies, Applications, CIC 2004, Scottsdale, Arizona, USA, November 9-12, 2004*, pages 37–41, 2004.

[19] William T. Freeman and David H. Brainard. Bayesian decision theory, the maximum local mass estimate, and color constancy. In *Procedings of the Fifth International Conference on Computer Vision (ICCV 95), Massachusetts Institute of Technology, Cambridge, Massachusetts, USA, June 20-23, 1995*, pages 210–217, 1995.

[20] Brian V. Funt and Lilong Shi. The rehabilitation of maxrgb. In *18th Color and Imaging Conference, CIC 2010, San Antonio, Texas, USA, November 8-12, 2010*, pages 256–259, 2010.

[21] Brian V. Funt and Weihua Xiong. Estimating illumination chromaticity via support vector regression. In *The Twelfth Color Imaging Conference: Color Science and Engineering Systems, Technologies, Applications, CIC 2004, Scottsdale, Arizona, USA, November 9-12, 2004*, pages 47–52, 2004.

[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.

[23] Peter V. Gehler, Carsten Rother, Andrew Blake, Thomas P. Minka, and Toby Sharp. Bayesian color constancy revisited. In *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2008), 24-26 June 2008, Anchorage, Alaska, USA*, 2008.

[24] Arjan Gijsenij, Theo Gevers, and Marcel P Lucassen. Perceptual analysis of distance measures for color constancy algorithms. *JOSA A*, 26(10):2243–2256, 2009.

[25] Han Gong. Convolutional mean: A simple convolutional neural network for illuminant estimation. In *Proceedings of the British Machine Vision Conference 2019, BMVC 2019, Cardiff University, Cardiff, UK, September 9-12, 2019*, 2019.

[26] Ghalia Hemrit, Graham D Finlayson, Arjan Gijsenij, Peter Gehler, Simone Bianco, Brian Funt, Mark Drew, and Lilong

Shi. Rehabilitating the colorchecker dataset for illuminant estimation. In *Color and Imaging Conference*, volume 2018, pages 350–353. Society for Imaging Science and Technology, 2018.

[27] Geoffrey E. Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *CoRR*, abs/1207.0580, 2012.

[28] Yuanming Hu, Baoyuan Wang, and Stephen Lin. Fcˆ4: Fully convolutional color constancy with confidence-weighted pooling. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 330–339, 2017.

[29] Alex Kendall, Hayk Martirosyan, Saumitro Dasgupta, and Peter Henry. End-to-end learning of geometry and context for deep stereo regression. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pages 66–75, 2017.

[30] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[31] Karlo Koscevic and Nikola Banic. ISPA 2019 Illumination Estimation Challenge. `https://www.isispa.org/illumination-estimation-challenge`. Accessed November 14, 2019.

[32] Edwin H Land and John J McCann. Lightness and retinex theory. *Josa*, 61(1):1–11, 1971.

[33] Stuart P. Lloyd. Least squares quantization in PCM. *IEEE Transactions on Information Theory*, 28(2):129–136, 1982.

[34] Zhongyu Lou, Theo Gevers, Ninghang Hu, and Marcel P. Lucassen. Color constancy by deep learning. In *Proceedings of the British Machine Vision Conference 2015, BMVC 2015, Swansea, UK, September 7-10, 2015*, pages 76.1–76.12, 2015.

[35] Siddharth Mahendran, Haider Ali, and René Vidal. A mixed classification-regression framework for 3d pose estimation from 2d images. In *British Machine Vision Conference 2018, BMVC 2018, Northumbria University, Newcastle, UK, September 3-6, 2018*, page 72, 2018.

[36] Fabian Manhardt, Diego Arroyo, Christian Rupprecht, Benjamin Busam, Tolga Birdal, Nassir Navab, and Federico Tombari. Explaining the ambiguity of object detection and 6d pose from visual data. *2019 IEEE International Conference on Computer Vision, ICCV 2019, Seoul, Korea, October 29-November 1, 2019*, 2019.

[37] Steven McDonagh, Sarah Parisot, Zhenguo Li, and Gregory G. Slabaugh. Meta-learning for few-shot camera-adaptive color constancy. *CoRR*, abs/1811.11788, 2018.

[38] Seoung Wug Oh and Seon Joo Kim. Approaching the computational color constancy as a classification problem through deep learning. *Pattern Recognition*, 61:405–416, 2017.

[39] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*, pages 8024–8035, 2019.

[40] Yanlin Qian, Ke Chen, and Huanglin Yu. Fast fourier color constancy and grayness index for ISPA illumination estimation challenge. In *11th International Symposium on Image and Signal Processing and Analysis, ISPA 2019, Dubrovnik, Croatia, September 23-25, 2019*, pages 352–354, 2019.

[41] Nguyen Ho Man Rang, Dilip K. Prasad, and Michael S. Brown. Raw-to-raw: Mapping between image sensor color responses. In *2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, Columbus, OH, USA, June 23-28, 2014*, pages 3398–3405, 2014.

[42] Joseph Redmon, Santosh Kumar Divvala, Ross B. Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 779–788, 2016.

[43] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster R-CNN: towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6):1137–1149, 2017.

[44] Charles R. Rosenberg, Thomas P. Minka, and Alok Ladsariya. Bayesian color constancy with non-gaussian models. In *Advances in Neural Information Processing Systems 16 [Neural Information Processing Systems, NIPS 2003, December 8-13, 2003, Vancouver and Whistler, British Columbia, Canada]*, pages 1595–1602, 2003.

[45] Christian Rupprecht, Iro Laina, Robert S. DiPietro, and Maximilian Baust. Learning in an uncertain world: Representing ambiguity through multiple hypotheses. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pages 3611–3620, 2017.

[46] A. Savchik, Egor I. Ershov, and Simon M. Karpenko. Color cerberus. In *11th International Symposium on Image and Signal Processing and Analysis, ISPA 2019, Dubrovnik, Croatia, September 23-25, 2019*, pages 355–359, 2019.

[47] Lilong Shi and Brian Funt. Re-processed version of the gehler color constancy dataset. `https://www2.cs.sfu.ca/~colour/data/shi_gehler/`. Accessed November 14, 2019.

[48] Wu Shi, Chen Change Loy, and Xiaoou Tang. Deep specialized network for illuminant estimation. In *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part IV*, pages 371–387, 2016.

[49] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[50] Joost van de Weijer, Theo Gevers, and Arjan Gijsenij. Edge-based color constancy. *IEEE Transactions on Image Processing*, 16(9):2207–2214, 2007.

[51] Johannes Von Kries. Influence of adaptation on the effects produced by luminous stimuli. *handbuch der Physiologie des Menschen*, 3:109–282, 1905.

[52] Ning Wang, De Xu, and Bing Li. Edge-based color constancy via support vector regression. *IEICE Transactions on Information and Systems*, 92-D(11):2279–2282, 2009.

[53] Weihua Xiong and Brian Funt. Estimating illumination chromaticity via support vector regression. *Journal of Imaging Science and Technology*, 50(4):341–348, 2006.