Outdoor daytime multi-illuminant color constancy

Ilija Domislović, Donik Vršnak, Marko Subašić, Sven Lončarić
University of Zagreb Faculty of Electrical Engineering and Computing, Zagreb, Croatia
Email: ilija.domislovic@fer.hr

***Abstract*—White-balancing is an important part of the image processing pipeline and is used in many computer vision applications. It removes the chromatic influence of the illumination on objects in the scene. White balancing is important in tasks such as object detection and object tracking. This problem is tackled in a myriad of ways, but most methods use the assumption that images contain only one dominant uniform illuminant. In recent years, neural networks have been used to create state-of-the-art methods for single illuminant white-balancing, but the problem of multi-illuminant white-balancing has been largely ignored. The main reason for this is the lack of multi-illuminant datasets. In this paper, we introduce a convolutional neural network for multi-illuminant (sun and shadow) illumination estimation. For the training and testing of the created model over 100 outdoor daytime images were taken using the Canon EOS 550D camera. We show that the model outperforms existing statistics-based methods on the test data.**

***Index Terms*—multi-illuminant estimation, multi-illuminant dataset, color constancy, convolutional neural networks.**

I. Introduction

All modern cameras use methods that perform white-balancing. There are many different methods, but they all attempt to emulate the human visual system's ability to perceive an object's intrinsic color even when the color is altered by a chromatic illuminant, also known as color constancy.

In recent years, the best-performing methods for illuminant estimation have been convolutional neural networks (CNN) [15]. The problem with these CNN models is that they were designed, trained, and tested on images that have only one dominant illuminant. The assumption that there is only a single illuminant is often violated. For images with multiple illuminants, these methods produce erroneous illumination estimations, resulting in images that look unnatural.

While there are methods [3, 21] that perform multi-illuminant estimation using neural networks, they mostly use artificially colored images. This is because there are no large datasets of labeled multi-illuminant images.

The main challenge in the creation of a larger multi-illuminant dataset is the need for correct image labeling. An example of labeling can be seen in [8], where for a single dataset there are three different ground truths. A method's accuracy greatly depends on the ground truth used. The dataset in [8] contains images with only one illuminant and the problem is only exacerbated when more illuminants are present. When there are more illuminants, determining the number of illuminants can also be a problem. Scene selection is important since a scene could have multiple illuminants where the human eye sees no visible difference. Some examples are streets at nighttime and hallways with multiple light bulbs. The camera picks up the subtle differences between these illuminants causing color correction to look unnatural. The simplest two illuminant images are outdoor daytime images with clear skies. In such situations, the delineation between illuminants is easily recognizable and there are rarely additional illuminants present.

In this paper, we created over 100 real-world outdoor images that contain two illuminants. The convolutional neural network that is trained on the images is based on [15] that performs single-illuminant estimation.

The paper is divided into sections as follows. Section II presents a color constancy problem formulation. In this section an overview of methods for illumination estimation is presented. Section III presents existing datasets and gives an overview of how the new images were collected. Section IV presents the model. Section V shows results obtained on the dataset using existing methods. Finally, in Section VI the conclusion and future directions are presented.

II. Color constancy overview

A. Problem formulation

To achieve color constancy the chromatic effect of illumination needs to be removed. This is achieved in two steps. The first step is the estimation of the image illuminant. The second step is the color correction of the image using the estimated illuminant.

Digital images are constructed from pixels. A pixel contains three values the red, green, and blue color intensity in it's location $f = (f_r, f_g, f_b)$. A popular image formation model is the Lambertian model [11].

$$f_c = \int_\omega I(\lambda) S(\mathbf{x}, \lambda) p_c(\lambda)\, d\lambda \tag{1}$$

A single pixel's value depends on the color of the illuminant $I(\lambda)$, surface reflectance $S(\mathbf{x}, \lambda)$, and the camera sensitivity function for each of the three values $p(\lambda) = (p_r(\lambda), p_g(\lambda), p_b(\lambda))$. $\mathbf{x}$ represents the spatial coordinates and λ represents the light wavelength.

For the second step of color correction, the von Kriss model [23] is commonly used. It uses a diagonal matrix, as it was shown that a diagonal matrix is sufficient [9] for image color correction. The model assumes that the sensor responses are independent.

$$I^c = \Lambda^{u,c} * I^u \tag{2}$$

2021 12th International Symposium on Image and Signal Processing and Analysis (ISPA) | 978-1-6654-2639-8/21/$31.00 ©2021 IEEE | DOI: 10.1109/ISPA52656.2021.9552092

Fig. 1. Example images from the dataset. For display purposes the images were tone mapped.

I^c is the image under the canonical illuminant, $\Lambda^{u,c}$ is the von Kriss diagonal matrix, and I^u is the image under the unknown illuminant. The diagonal matrix can be expressed as:

$$\Lambda^{u,c} = \begin{bmatrix} \frac{L_r^c}{L_r^u} & 0 & 0 \\ 0 & \frac{L_g^c}{L_g^u} & 0 \\ 0 & 0 & \frac{L_b^c}{L_b^u} \end{bmatrix} \tag{3}$$

where L_r^u, L_g^u, L_b^u are the red, green, and blue values of the unknown illuminant and L_r^c, L_g^c, L_b^c are the red, green, and blue values of the canonical illuminant. The canonical illuminant is the white light or an L value of $(1, 1, 1)^T$.

B. Related work

Illumination estimation methods can be split into two groups: statistics-based and learning-based.

Statistics-based methods are simpler and rely on low-level statistical information present in an image. An example of statistics-based methods is the White-Patch[18] method. It assumes that the color channel maximum response is caused by perfect reflectance. The illuminant is extracted by taking the maximum value of each color channel. Another statistics-based method is Grey-World[5]. It assumes that the average reflectance of a scene is achromatic and any divergence is caused by the illuminant. The illuminant is extracted by computing the average value of each color channel.

The assumptions from these methods are often violated resulting in unnaturally looking images. White-Patch does not work well with very bright images where camera sensors are over-saturated and the maximum value cannot be extracted. Grey-world does not work well with images that have a few surfaces since the average color in the image becomes the color of the largest surface.

These two methods are part of a larger framework called Grey-Edge[22]. Here the assumption is that the average edge difference in a scene is achromatic.

$$\left(\int \left| \frac{\partial^n f_{c,\sigma}(\mathbf{x})}{\partial \mathbf{x}^n} \right|^p d\mathbf{x} \right)^{\frac{1}{p}} = k L_c^{n,p,\sigma} \tag{4}$$

These methods are meant for single illuminant estimation, but some methods use them for multi-illuminant estimation. They achieve this by splitting the image into small patches. They use the assumption that these small patches contain only one illuminant.

In [12], to segment the image they used several different methods: grid-based, segmentation-based, and keypoint-based segmentation. After the image has been segmented, a method from the Grey-Edge framework is used to extract the illuminant for each patch. The global estimations are created by grouping the local estimations using any clustering algorithm.

In [4], they separate the image into a set of superpixels based on the color value of the pixels. For each superpixel, a Grey-Edge framework method is used to estimate the illuminant. To combine the local estimation, they use several different

methods. One is Gradient tree boosting [10] and the other is Random Forest regression [13].

In [2], the image is segmented into patches using a uniform grid. They also tried segmenting the image into superpixels, but they opted for uniform-grid since the results were comparable and uniform-grid is simpler. The illuminant of each patch is estimated using a Grey-Edge framework method. To obtain the global illumination estimations they use Random conditional fields.

The other more recent group of methods are the learning-based methods. It can be seen [15, 24, 1] that these methods perform far better than the statistics-based methods, with the downside being that these methods are more complex than the statistics-based methods.

In [15], the authors used and repurposed two neural networks created for image classification on ImageNet[7]. They used AlexNet[17] and SqueezeNet[16] pre-trained in ImageNet. These models were repurposed by removing the final Fully-connected layers from the models and replacing them with Convolutional layers and a Confidence-weighted pooling layer. The final Convolutional layer produces a four-channel output. The first three channels represent the local illumination estimations, while the final channel represents the confidence mask of the local illuminant estimations. This four-channel output is then fed into the Confidence-weighted pooling layer. This layer multiplies the confidence mask with the illumination estimations, sums all the estimations into a global illumination estimation, and finally normalizes the estimation to produce the model output.

In [1], they transform the problem by reducing the problem into a spatial localization task. To transform the problem they use the fast Fourier transformation to operate in the frequency domain. They perform convolutions over histograms in the log-chromatic color space. This results in an efficient method that performs $250-3000$ times faster than other learning-based methods and a method that works in real-time on a mobile device.

There are many more methods that are used for single illumination estimation, but there are a few methods that were created for multi-illuminant estimation. They also perform illumination estimation by splitting the image into patches, estimating local illumination, and clustering the local estimation into global illumination estimations.

In [3], they used a simple neural network with 2 convolutional and 2 fully connected layers. To determine the number of illuminants in a scene they normalize the local illuminant estimations and employ a 2D kernel density estimation (KDE). The KDE determines whether there are one or more illuminants in the scene. If there is only one illuminant they perform support vector regression [6] to get a global illumination estimation. If there are multiple illuminants, the patch estimations are combined into an illumination mask that is then compared to the ground truth illumination mask.

In [21], to estimate an illuminant the authors use two different neural networks: HypNet and SelNet. The HypNet is the network that performs patch illumination estimation but unlike other approaches, it has two separate predictions for the patch illumination. The authors argue that in such an architecture each prediction specializes in a different type of patch (e.g. bright regions or textured regions). The SelNet model's job is to select the HypNet prediction that more accurately estimates the patch illuminant.

The problem with the patch-based approach is that some patches do not have enough useful information for accurate illumination estimation. For example, a patch that only shows a yellow wall. In this situation, there is not enough information to deduce whether the wall is yellow or the wall is white and is illuminated by a yellow illuminant. For this reason, the model in this paper performs global illumination estimation over the entire image.

III. Dataset

The problem with neural networks is that they need a large amount of training data. There are some multi-illuminant datasets available, but none of them have enough data for neural network training.

In [4], a multi-illuminant dataset is presented. The problem is that this dataset only contains 36 multi-illuminant images and all of the images were taken under laboratory conditions. The Multiple Light Sources dataset [12] has more images. The dataset contains 59 images taken under laboratory conditions and 9 real-world images.

The Multiple-Illuminant Multi-Object dataset [2] has 60 images taken under laboratory conditions and 20 real-world images. This paper also introduces a way to automatically label the segmentation mask of an image. They do this by taking three images of the scenes, one with both illuminants and two with only one of the illuminants. For this to work, you need the ability to turn off the illuminants, which is often not possible.

These datasets have less than 100 images and the number of real-world images is even smaller. The number of real-world images is far too small for effective neural network training. While images taken under laboratory conditions are good they cannot perfectly emulate all the situations that can arise in real-world situations.

Therefore, for the training and testing of the model, we created over 100 real-world images. The images contain outdoor daytime scenes, where one of the illuminants is the sun and the other is the ambient light of the sky present in the shadows.

These images were taken in the northwest region of Croatia.

The illuminants in each image were calculated using SpyderCubes, which can be seen in Figure 2. Each SpyderCube has four faces: two grey and two white, with a grey and white face sharing a flat surface.

Illuminants were extracted by calculating the average value from a grey face. One SpyderCube was placed under direct sunlight and the other was placed in a shadow.

To select which grey face will be used as ground truth the following process was used. Firstly, an illuminant was extracted from each of the two white and two grey faces. Then the angle between the grey and white face of each surface

Fig. 2. SpyderCube used for illumination extraction

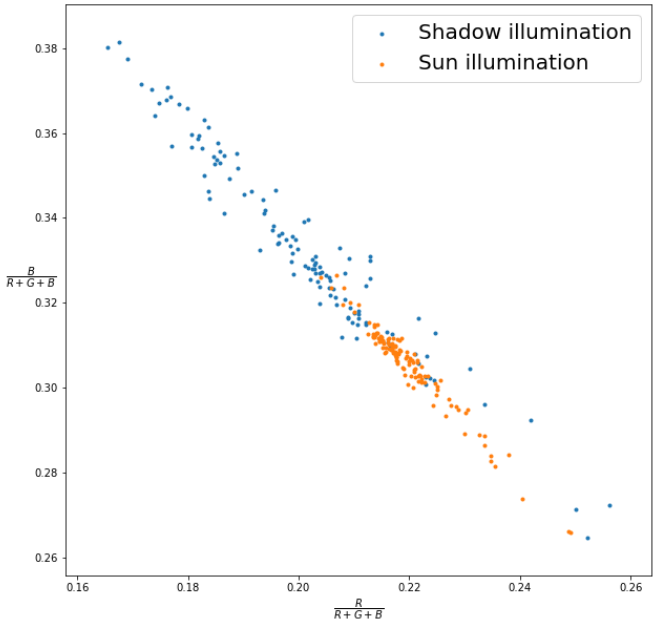

Fig. 3. Illumination ground truth for each image

was calculated using cosine similarity. The grey face with the smaller angle was chosen as the ground truth. If only one of the sides of the cube was under direct sunlight that side is chosen for the sunlight ground truth.

To ensure there are two illuminants the angle between the sunlight and shadow illuminant is examined. In [14] it is stated that humans have difficulties distinguishing illuminants where the angle is less than 2°. The authors of [14] state those findings are not conclusive, so images with an angle of of 1° were also used.

To use images for training and testing, the following preprocessing steps were performed. Firstly, the black level or the level of brightness of the darkest pixels has to be subtracted from the whole image. The camera used for dataset creation is the Canon EOS 550D and it has a black level value of 2048. The image regions that contain SpyderCubes were blacked out. This was done to make it impossible for the model to simply learn to find the SpyderCubes and extract the illuminant. Finally, image pixels that have a response value that is greater than or equal to m-2, where m is the maximum channel response in an image, were set to 0. This was done to remove the effect of oversaturated image pixels.

IV. Proposed Model

To evaluate the dataset and see how well neural networks compare to statistics-based estimation methods, a convolutional neural network was created. The used convolutional neural network is based on the FC4 [15] model.

FC4 uses SqueezeNet [16] pre-trained on ImageNet [7] as an image feature extractor, where the final few layers are replaced by two randomly initialized convolutional layers. The output from the final convolutional layer are the local illumination estimations. These local illumination estimations are fed into a Confidence-weighted pooling layer also introduced in [15]. This layer contains an attention mask that is used to filter out local estimations the model deems too inaccurate for estimation.

For multi-illuminant images, the model was modified to have two outputs, one for each of the illuminants. This was done by duplicating the randomly initialized convolutional layers so that both illumination estimators use the same feature extractor, but use different illumination estimation feature masks. The process of taking an existing illumination estimation model and modifying it to have two outputs could have been done with any model, but FC4 [15] was used because of the Confidence-weighted pooling layer.

In [15], this layer is used to train the model to ignore regions of an image that do not contain enough information for accurate illumination estimation. An example of such a region is a single color wall. In multi-illuminant images, regions where the illuminant is not present are not useful for illumination estimation. Since the model has separate outputs for sunlight and shadow, this layer is used to train the model to also ignore the illuminant not associated with output.

This model was also selected because it performs illumination estimation over the entire image unlike [3] and [21].

The model was trained for 200 epochs with batch size of 28 on the Nvidia 2080Ti GPU and Ryzen 7 3700X CPU. The AdamW [20] with weight decay 5e-5 and a learning rate of 2e-4, was used as the optimizer. The mean squared error loss function was used. The size of the input image is 227x227. The training and valudation plot can be seen in Figure 4.

V. Results

To evaluate the new dataset three different tests were performed. In the first test regions in the shadows are estimated, in the second test regions under direct sunlight are estimated, and in the final test, both illuminants are estimated. For testing

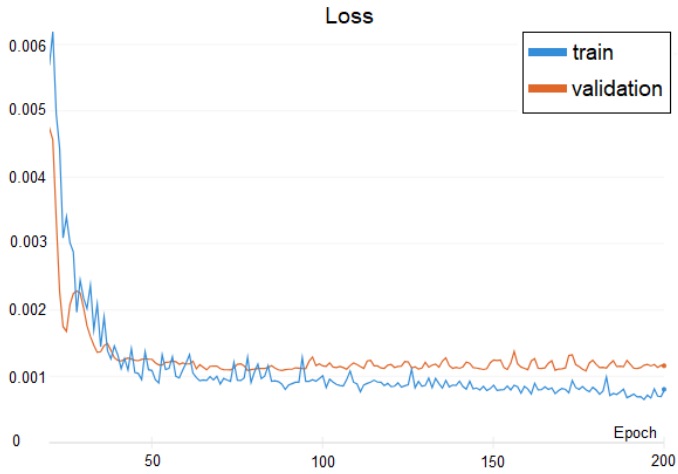

Fig. 4. Plot of model loss on train and validation datasets

TABLE I
COMPARISON OF RESULTS OBTAINED WHEN ONLY SHADOW ILLUMINATION ERROR IS EXAMINED.

Method	mean	med.	tri.	best 25%	worst 25%
do nothing	18.19	18.15	18.19	17.15	19.22
Grey-world (32,32)[5]	14.44	15.31	15.18	10.25	17.13
Grey-world (64,64)[5]	14.4	15.25	15.11	10.34	17.06
White-patch (32,32)[18]	12.92	13.81	13.6	8.1	16.11
White-patch (64,64)[18]	12.38	13.16	13.03	7.32	15.74
general Grey-world (32,32)[22]	14.12	15.17	14.93	9.41	17.05
general Grey-world (64,64)[22]	13.85	15.05	14.68	8.95	16.93
1st order Grey-edge (32,32)[22]	8.64	4.44	5.19	1.41	22.89
1st order Grey-edge (64,64)[22]	6.76	3.84	4.33	1.29	17.39
2nd order Grey-edge (32,32)[22]	7.4	6.44	6.77	2.09	14.69
2nd order Grey-edge (64,64)[22]	6.64	5.5	5.32	1.81	14.18
Proposed model	2.14	1.78	1.88	0.57	4.28

several Grey-edge framework variants as well as the proposed CNN model were used.

To compare the results of the models the angular error in degrees was used. The evaluation metrics used were the standard metrics used by similar papers. The included metrics are mean, median, trimean, best 25% mean, and worst 25% mean.

$$Angular\ error = cos^{-1}\left(\frac{\mathbf{L}\cdot\hat{\mathbf{L}}}{||\mathbf{L}||_2||\hat{\mathbf{L}}||_2}\right) \quad (5)$$

When estimating using the Grey-edge framework the images are divided into patches for local illuminant estimation. The local estimations are then clustered into two illuminants using k-means [19] clustering. To create patches a uniform grid was

TABLE II
COMPARISON OF RESULTS OBTAINED WHEN ONLY SUN ILLUMINATION ERROR IS EXAMINED.

Method	mean	med.	tri.	best 25%	worst 25%
do nothing	17.66	17.67	17.67	17.40	17.90
Grey-world (32,32)[5]	10.0	9.28	9.4	7.25	13.95
Grey-world (64,64)[5]	10.07	9.43	9.53	7.36	13.88
White-patch (32,32)[18]	8.59	7.63	7.73	5.93	13.14
White-patch (64,64)[18]	8.3	7.23	7.62	5.85	12.61
general Grey-world (32,32)[22]	9.56	8.55	8.75	6.78	13.96
general Grey-world (64,64)[22]	9.33	8.28	8.5	6.57	13.7
1st order Grey-edge (32,32)[22]	5.99	3.8	4.09	1.26	14.54
1st order Grey-edge (64,64)[22]	5.71	3.8	4.02	1.21	13.39
2nd order Grey-edge (32,32)[22]	4.92	3.52	3.75	1.67	10.51
2nd order Grey-edge (64,64)[22]	4.97	3.57	3.72	1.71	10.61
Proposed model	0.92	0.68	0.73	0.26	2.04

TABLE III
COMPARISON OF RESULTS OBTAINED WHEN BOTH ILLUMINATIONS ERROR ARE EXAMINED.

Method	mean	med.	tri.	best 25%	worst 25%
do nothing	17.93	17.81	17.84	17.26	18.81
Grey-world (32,32)[5]	12.22	12.6	12.44	7.72	16.51
Grey-world (64,64)[5]	12.24	12.56	12.39	7.86	16.43
White-patch (32,32)[18]	10.75	10.96	10.9	6.3	15.46
White-patch (64,64)[18]	10.34	10.01	10.19	6.12	15.04
general Grey-world (32,32)[22]	11.84	12.35	12.06	7.18	16.42
general Grey-world (64,64)[22]	11.59	11.89	11.73	6.99	16.26
1st order Grey-edge (32,32)[22]	7.32	4.15	4.42	1.33	19.1
1st order Grey-edge (64,64)[22]	6.23	3.82	4.08	1.25	15.55
2nd order Grey-edge (32,32)[22]	6.16	4.8	4.82	1.77	13.28
2nd order Grey-edge (64,64)[22]	5.8	4.15	4.31	1.75	12.73
Proposed model	1.53	1.02	1.21	0.34	3.49

used. Patch sizes of $(32, 32)$ and $(64, 64)$ were used to see how the patch size affects accuracy. Bigger patches were not used since the assumption of a single illuminant in a patch is violated in larger patches. It can bee seen in Tables I II III that the patch size does not affect Grey-world [5] and White-patch [18], but the other Grey-edge [22] methods see significant improvement with the larger patch size. For the CNN model, a 3-fold split was used so that the model could be properly compared to the other methods since they do not need training and can be tested on all the images.

As the Tables I II III show the CNN approach outperforms all the Grey-edge methods. An interesting fact that can be observed in Tables I II is that sunlight estimation is significantly more accurate than the shadow illumination estimation. This can be explained by the fact that the sunlight illumination gamut is much smaller than the gamut of the shadow illumination as can be seen in Figure 3.

VI. Conclusion

This paper presents a new multi-illuminant dataset and the details of how the images were collected and annotated. A convolutional neural network for illumination estimation in outdoor daytime images is also presented. This model outperforms statistics-based methods. The presetned work is preliminary and in the future we hope to create a proper dataset with a diverse set of images taken by multiple different cameras.

References

[1] Jonathan T Barron and Yun-Ta Tsai. "Fast fourier color constancy". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 886–894.

[2] Shida Beigpour et al. "Multi-illuminant estimation with conditional random fields". In: *IEEE Transactions on Image Processing* 23.1 (2013), pp. 83–96.

[3] Simone Bianco, Claudio Cusano, and Raimondo Schettini. "Single and multiple illuminant estimation using convolutional neural networks". In: *IEEE Transactions on Image Processing* 26.9 (2017), pp. 4347–4362.

[4] Michael Bleier et al. "Color constancy and non-uniform illumination: Can existing algorithms work?" In: *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*. IEEE. 2011, pp. 774–781.

[5] Gershon Buchsbaum. "A spatial processor model for object colour perception". In: *Journal of the Franklin institute* 310.1 (1980), pp. 1–26.

[6] Corinna Cortes and Vladimir Vapnik. "Support-vector networks". In: *Machine learning* 20.3 (1995), pp. 273–297.

[7] Jia Deng et al. "Imagenet: A large-scale hierarchical image database". In: *2009 IEEE conference on computer vision and pattern recognition*. Ieee. 2009, pp. 248–255.

[8] G. Finlayson et al. "A Curious Problem with Using the Colour Checker Dataset for Illuminant Estimation". In: 2017.

[9] Graham D Finlayson, Mark S Drew, and Brian V Funt. "Diagonal transforms suffice for color constancy". In: *1993 (4th) International Conference on Computer Vision*. IEEE. 1993, pp. 164–171.

[10] Jerome H Friedman. "Greedy function approximation: a gradient boosting machine". In: *Annals of statistics* (2001), pp. 1189–1232.

[11] Arjan Gijsenij, Theo Gevers, and Joost Van De Weijer. "Computational color constancy: Survey and experiments". In: *IEEE Transactions on Image Processing* 20.9 (2011), pp. 2475–2489.

[12] Arjan Gijsenij, Rui Lu, and Theo Gevers. "Color constancy for multiple light sources". In: *IEEE Transactions on Image Processing* 21.2 (2011), pp. 697–707.

[13] Tin Kam Ho. "Random decision forests". In: *Proceedings of 3rd international conference on document analysis and recognition*. Vol. 1. IEEE. 1995, pp. 278–282.

[14] Steven D Hordley. "Scene illuminant estimation: past, present, and future". In: *Color Research & Application: Endorsed by Inter-Society Color Council, The Colour Group (Great Britain), Canadian Society for Color, Color Science Association of Japan, Dutch Society for the Study of Color, The Swedish Colour Centre Foundation, Colour Society of Australia, Centre Français de la Couleur* 31.4 (2006), pp. 303–314.

[15] Yuanming Hu, Baoyuan Wang, and Stephen Lin. "Fc4: Fully convolutional color constancy with confidence-weighted pooling". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 4085–4094.

[16] Forrest N Iandola et al. "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and¡ 0.5 MB model size". In: *arXiv preprint arXiv:1602.07360* (2016).

[17] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "Imagenet classification with deep convolutional neural networks". In: *Advances in neural information processing systems* 25 (2012), pp. 1097–1105.

[18] Edwin H Land. "The retinex theory of color vision". In: *Scientific american* 237.6 (1977), pp. 108–129.

[19] Stuart Lloyd. "Least squares quantization in PCM". In: *IEEE transactions on information theory* 28.2 (1982), pp. 129–137.

[20] Ilya Loshchilov and Frank Hutter. "Decoupled weight decay regularization". In: *arXiv preprint arXiv:1711.05101* (2017).

[21] Wu Shi, Chen Change Loy, and Xiaoou Tang. "Deep specialized network for illuminant estimation". In: *European conference on computer vision*. Springer. 2016, pp. 371–387.

[22] Joost Van De Weijer, Theo Gevers, and Arjan Gijsenij. "Edge-based color constancy". In: *IEEE Transactions on image processing* 16.9 (2007), pp. 2207–2214.

[23] J. VON KRIES. "Influence of adaptation on the effects produced by luminous stimuli". In: *handbuch der Physiologie des Menschen* 3 (1905), pp. 109–282. URL: https://ci.nii.ac.jp/naid/10030415665/en/.

[24] Jin Xiao, Shuhang Gu, and Lei Zhang. "Multi-Domain Learning for Accurate and Few-Shot Color Constancy". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 3258–3267.