

# Generative Models for Multi-Illumination Color Constancy

Partha Das<sup>1,2</sup>

p.das@uva.nl

Yang Liu<sup>2</sup>

yang@3duniversum.com

Sezer Karaoglu<sup>1,2</sup>

s.karaoglu@3duniversum.com

Theo Gevers<sup>1,2</sup>

th.gevers@uva.nl

University of Amsterdam, The Netherlands<sup>1</sup>

3DUniversum, Amsterdam, The Netherlands<sup>2</sup>

## Abstract

*In this paper, the aim is multi-illumination color constancy. However, most of the existing color constancy methods are designed for single light sources. Furthermore, datasets for learning multiple illumination color constancy are largely missing. We propose a seed (physics driven) based multi-illumination color constancy method. GANs are exploited to model the illumination estimation problem as an image-to-image domain translation problem. Additionally, a novel multi-illumination data augmentation method is proposed. Experiments on single and multi-illumination datasets show that our methods outperform *sota* methods.*

## 1. Introduction

The color of an object is influenced by the illumination color. The process of recovering the original object color, independent of the illumination color, is called Color constancy. Previous approaches solve the problem by enforcing various priors [31, 12, 37]. More recently, with the availability of relatively large scale datasets [16, 19, 4, 3], learning based approaches are explored [33, 9]. Various learning based approaches are proposed such as local-global patch correction [25] and exemplar patch based correction [30], that show robust performance. However, these methods predict the illumination vector directly. This limits the learning capability of the network by limiting all the pixels to be supervised by the illumination vector. Modelling the problem as an image-to-image translation could provide a solution to this, allowing finer pixel wise supervision. However, standard encoder-decoder network works by minimising the Euclidean distance between the prediction and the ground truth. This may lead to blurry outputs and border artefacts, while being fully dependent only on the input.

Generative Adversarial Networks [26] (GAN) solve this problem by learning an image-to-image translation as a domain transformation. In this setting, the generator learns an expanded pixel-wise supervision, while the discrimina-

tor allows the network to learn domain specific information. To exploit the unified framework, we propose to use GANs to model color constancy as a domain transformation, instead of image to illumination prediction.

However, the previous methods are often limited to a single illumination assumption. For multi-illumination scenes, each pixel in an image is influenced by different light sources. GAN models like Pix2Pix or CycleGAN learn a global transformation. Even with pixel wise supervision, Pix2Pix learns a pixel wise transformation, excluding the spatial context. In contrast, we propose a hybrid learning and seed based multi-illumination color constancy. Using physics based (initial) seed points, the network is provided with illumination regions. The network then diffuses these seed points over the image for a multi-illumination prediction. To model mixtures of lights at a pixel, an illumination probability is predicted. This allows the model to compute per-pixel mixture probability belonging to the different illuminations. Finally, due to the lack of multi-illumination datasets, a data augmentation method is proposed to create a new multi-illumination dataset. In summary, the proposed contributions of this work are as follows:

- We propose to model the problem of color constancy as a domain transformation with GANs.
- We propose a new method for exploiting physics based initial seed points for multi-illumination color constancy.
- We propose a novel data augmentation method to create multi-illumination dataset using off-the-shelf methods and single illumination datasets.

## 2. Related Work

Color constancy methods can be classified into optimization-based methods and learning based methods.

**Optimization based methods:** These methods exploit various low-level image statistics or priors [31, 12, 22, 28, 20, 42]. New features are proposed in [37] tackling both single and multi-illumination in a learning free manner. Obvi-

ously, these methods may fail when the underlying assumptions/priors are violated.

**Learning based methods:** Various learning based methods are proposed such as [39, 21, 10, 23, 8, 14] using classical learning methods like Bayesian frameworks, Support Vector Machines, Decision Forests, and Conditional Random Fields. Exemplar based methods [30] explore multi-illumination settings using pixel independent features. With the availability of larger datasets, CNNs are proposed [33, 11, 35, 27]. These methods employ deep learning frameworks in an end-to-end manner to model the problem as an illumination prediction. While these previous works focused on a single image, [36] integrates an implicit illumination constraint by using multiple frames and a deep recurrent network. [1], on the other hand, uses a two-camera-one-subject setup to estimate the illumination. Finally, [41, 40] formulated the problem into an image-to-image transformation. However, the methods suffer from dataset illumination bias.

In contrast to previous work, this paper proposes to approach the problem of color constancy as a domain transformation. Following our work [17], different GANs are first studied for the problem of single illumination color constancy. Their shortcomings for a multi-illumination setting are analyzed. These observations are then employed to propose an end-to-end hybrid multi-illumination color correction pipeline.

### 3. Methodology

#### 3.1. Color Constancy

The problem of color constancy can be defined as a process of chromatic adaptation [18]:

$$I = W * L, \quad (1)$$

where,  $W$  is the image under a canonical white illumination and  $L$  is the unknown colored illumination. The  $*$  operation denotes the pixel-wise multiplication between the components. Thus,  $L$  acts as a scaling term on  $W$  to generate the final  $I$ . All the components are in linear  $RGB$ . For a single light source,  $L$  is uniformly constant, while for multi-illumination  $L$  is varied.

The process of color correction can thus be approximated as modifying the gains of the  $RGB$  channels independently. To recover the color corrected image ( $W$ ), the Von Kries [44] method can be used as follows:

$$\begin{pmatrix} R_c \\ G_c \\ B_c \end{pmatrix} = \begin{pmatrix} L_R & 0 & 0 \\ 0 & L_G & 0 \\ 0 & 0 & L_B \end{pmatrix} \begin{pmatrix} R_u \\ G_u \\ B_u \end{pmatrix}, \quad (2)$$

where  $R_u, G_u, B_u$  denote  $RGB$  channels of an image under an unknown colored illumination source,  $R_c, G_c, B_c$

represent  $RGB$  channels under a canonical (corrected) illumination, and  $L_R, L_G$  and  $L_B$  correspond to the unknown illumination color.

Traditional approaches directly recover the illumination color from the input image. This forces the network to use the same supervision and “copy” it over all the pixels. In contrast, it is proposed to model the process of color constancy as domain mapping, i.e. from color biased ( $I$ ) image to color corrected ( $W$ ) image. This allows the network to be flexible to both single illumination and multi-illumination cases. GANs are used for this image-to-image translation, due to their ability to learn effective domain translation.

##### 3.1.1 Single illumination Color Constancy

In this section, three GAN models are adapted for the problem of color constancy under a single unknown illumination. The color biased image is denoted as  $I$ , while the color corrected image is denoted as  $W$ . The entire domain of color biased images is denoted by  $X$  and color corrected image domain as  $Y$ , that is  $I \in X$  and  $W \in Y$ .

**Pix2Pix [29]:** Pix2pix employs a U-net architecture [38] as the generator. Skip connections [34] allow the net to transfer high frequency details. *PatchGAN* is introduced as the discriminator that enforces high frequency consistency through a Markov random field. Standard GAN losses and L1 losses on paired cross domain images are used to train the network. This allows the network to learn a pixel-wise transfer between domains while keeping the underlying image structure consistent.

This formulation is particularly well suited to the problem of color constancy. The problem is defined as a domain transformation of the image taken under an unknown illumination ( $I$ ) to the color corrected image ( $W$ ). The L1 loss makes the network approximate a color correction function, while the adversarial loss ensures that the transformation is close to the target domain. This allows the network to predict color corrected images directly, while removing the unknown illumination color influence. Unfortunately, the L1 loss necessitates dense ground-truth pairs.

**CycleGAN [45]:** CycleGAN improves on Pix2Pix by introducing a forward and backward cycle. The network learns a closed cycle through: 1) forward transformation ( $I \rightarrow W$ ), and 2) backward transformation ( $W \rightarrow I$ ). This allows the network to be trained unsupervised. The supervision signal is obtained from the outputs of the forward and backward cycles, called the cycle consistency loss. Given two generators,  $G$  (the forward generator) and  $F$  (the backward generator), we have:

$$\mathcal{L}_{cycle}(G, F) = E_{x \sim p_{data}(x)} \|F(G(x)) - x\|_1^1, \quad (3)$$

The network then learns the cycles  $G(X) \rightarrow Y'$  and

$F(Y') \rightarrow X'$ . This leads to learning a global transformation to approximate the color correction. Due to the unsupervised nature of training, any color biased and color corrected image can be assigned to the respective domain without the need for a dense ground-truth correspondence to train the network. Unfortunately, the network can learn only a single type of illumination correction for the color biased domain.

**StarGAN [15]:** To handle multi-domain translations simultaneously, StarGAN introduces an additional conditioning on the domain. The base model architecture of CycleGAN is used, with a domain vector on top of it. This vector specifies which target domain the network should map the input to. Consequently, the discriminator learns to classify the domain, in the form of an additional output. This allows the network to adversarially learn specific domain transformations conditioned on the input vector. Both the forward and backward cycles from CycleGAN are preserved, allowing the network to be trained in an unsupervised manner.

The ability to learn a transformation between various domains translates well to the problem of color constancy. This allows the network to learn a common transformation between different illumination types using a single model. Furthermore, the common domain defined transformation, yields a network having a consistency check in the form of whether the model can recover the same image from different illumination colors. The domains are also learned in an unsupervised manner, suited to train with limited data.

**Limitations:** The different GANs provide useful formulations for the problem of color constancy. However, (1) the networks are merely suited to the problem of single illumination color constancy, (2) the networks are not tailored to model the color correction process, (3) the networks implicitly learn the illumination as a latent state, and (4) the transformations learned are often global and are lacking spatial context. As a result, these networks need to be adapted to multi-illumination color constancy.

### 3.1.2 Multi-Illumination Color Constancy

In this section, a sample-based learning approach to solve the multi-illumination color constancy is proposed. In this, no assumption on the number or spatial influence of the illuminants are made. Instead, random initial seed points are provided to the network as inputs. The ground truth illumination map is used to randomly sample seed points for different light sources during training. The network uses these initial points to disentangle the illumination influence for all pixels. However, this might lead the network to be biased to the initially selected seeds. To avoid the bias, the color biased image ( $I$ ) is also provided to the network as an additional input. The color biased image provides the network with a global cue, while the initial seed points pro-

vide a local cue. Using these cues, the network can learn to diffuse the initial sample points through the image and inpaint an illumination map. The input modalities are given in figure 1.

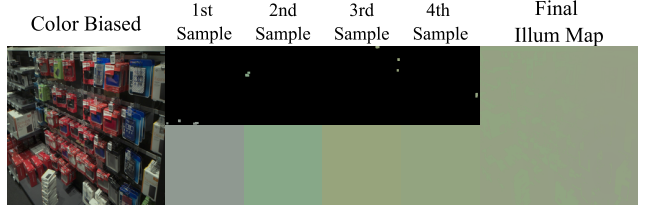


Figure 1. Examples of the dataset augmentation. The 1st-4th samples are the random samples from the respective illuminants. The color patches below it are the illumination themselves. The Final Illum Map is the total illumination maps consisting of the four illuminants. All inputs are of the same spatial resolution as the color biased image. They have been resized here for visualisation.

For multi-illumination, every pixel can have varying influence from the different illuminants. To model this, the network predicts a probability map, instead of the color corrected image. In this, each pixel is assigned a probability of belonging to a specific illumination. This transforms the problem to a per-pixel classification task, rather than a global transformation. As a result, the network does not have to cluster the illuminants in the latent space, while having a realistic illumination mixture for each pixel. The output of the network is a  $N$ -channel probability map for  $N$  illuminants. Each of the channels represents the probability of a single pixel belonging to the  $N$ -th illumination. The predicted illumination map is reconstructed using the following:

$$L_{pred} = \sum_{i=1}^N P_i \times I_i \quad (4)$$

where,  $P_i$  is the  $i$ -th channel in the  $N$  channel probability map predicted by the network and  $I_i$  is the corresponding  $i$ -th illumination.  $L_{pred}$  is the predicted illumination map obtained from the probability map.

Finally, to let the network learn a realistic transformation, a PatchGAN discriminator is used. This helps to keep the network from learning a dataset bias. An overview of the generator is shown in figure 2.

During inference, Grayness Index [37] is used to determine the illumination clusters and randomly sample from those clusters. The grayness index is a physics-based, non-learning estimation. Pix2Pix is used as the base model architecture since it is more suited for local illumination estimation which is beneficial for multi-illumination color constancy. The direct supervision in Pix2Pix is also useful for the probability output of the network. In CycleGAN, the transformation would have to be done from image and sampled seed to probability, in the forward cycle, and then from

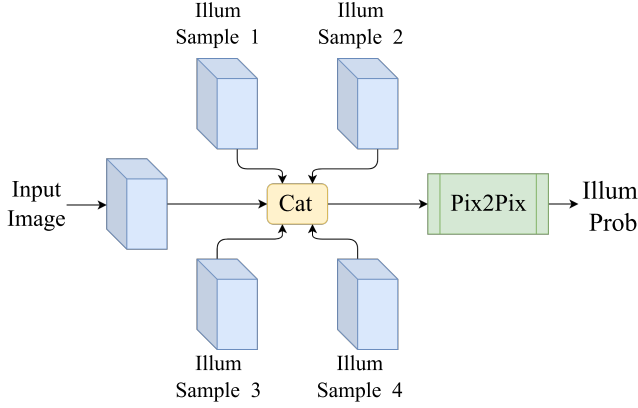


Figure 2. Overview of the proposed multi-illumination color constancy network. The samples are a concatenation of the image samples and the corresponding masks. The cube denotes convolutions. The output of the network is a probability map, which is processed using equation (4) to obtain the illumination map. Please refer to the supplementary material for details.

probability to image and sampled seed for the backward cycle. This cycle consistency would be more difficult for a GAN to learn since the probability map from the forward cycle would not give enough information for the backward cycle. Additionally, compared to CycleGAN and StarGAN, Pix2Pix consists of just a single generator and discriminator, hence it has comparatively lower parameters, making it quicker and easier to train. The network also provides a local transformation, which is useful for scenes with multiple light sources since each pixel can be affected by the combined effect of all the light sources. To integrate the sample dispersion learning mechanism, an additional shallow network before the Pix2Pix generator is added. This consists of two layers of convolutions to condition the incoming features, increasing and then squeezing the feature channels. For the input illumination samples, the sample mask is also concatenated. This allows the network to learn a diffusion of the samples to their correct regions and back-propagate the errors from the seed sample. Details on the shallow network can be found in the supplementary materials.

**Loss Functions:** The predicted probability map is constrained by a loss on the predicted illumination calculated using (4):

$$\mathcal{L}_{illum} = \frac{1}{M} \sum_{i=1}^M ||\hat{L}_i - L_i|| \quad (5)$$

where  $M$  are the total number of pixels,  $\hat{L}_i$  is the predicted illumination map and  $L_i$  is the ground truth illumination map.

Furthermore, a loss on the color corrected image is

added:

$$\mathcal{L}_{rgb} = \frac{1}{M} \sum_{i=1}^M ||\hat{W}_i - W_i|| \quad (6)$$

where  $\hat{W}_i$  is the predicted color corrected image. This is obtained by using (2) with the input color biased image and the predicted illumination map. The  $W_i$  is the ground truth color corrected image.

Furthermore, to ensure that the probability map assigns the illuminants to the proper illumination region, an illumination mask loss is added. The initial seed points, given as input to the network, are compared to the corresponding points on the predicted illumination map.

$$\mathcal{L}_{masks} = \sum_{j=1}^N ||\hat{L}_j * B - L_j * B|| \quad (7)$$

where  $B$  are the binary masks denoting the locations of seed sampling points. This loss is calculated for each illumination.

Finally, we use an adversarial training to train the entire network:

$$\mathcal{L}_{GAN} = E[\log D(I)] + E[\log(1 - D(G(\hat{I})))] \quad (8)$$

where,  $G(x)$  is the generator and  $D(x)$  is the discriminator.

Thus, the total training objective for the network becomes:

$$\mathcal{L}_{total} = \mathcal{L}_{GAN} + \lambda(\mathcal{L}_{illum} + \mathcal{L}_{rgb} + \mathcal{L}_{masks}) \quad (9)$$

where,  $\lambda$  is a hyper-parameter. This was empirically fixed to be 100.

## 4. Experiments

### 4.1. Datasets

#### 4.1.1 Single Illumination Datasets

The GAN models are trained and evaluated on 3 standard single illumination benchmarks: (1) The SFU Gray Ball [16] dataset, (2) the recalculated version of [19] called *ColorChecker RECommended* (RCC) and (3) the Cube dataset [4]. For the experiments, the datasets are randomly split into 80% training and 20% testing. The reference objects in the images are masked out.

#### 4.1.2 Multi Illumination Datasets

Large datasets for multi-illumination color constancy are difficult to obtain. Therefore, there are only a few datasets publicly available with limited number of images [8]. To

train our network, a multi-illumination augmentation approach is proposed. Semantic segmentation algorithms are used to obtain object boundaries. Then, using a mixture model, the images are augmented with different illuminants. The illuminants are obtained from an existing multi-illumination dataset like MIMO [8]. The illuminants are then channel shuffled and combined with color corrected images using (2) to obtain a multi illumination image. This approximates images taken from multi-illumination scenes with realistic illumination interaction, textures, and colors.

For the experiments, the Intel TUT [3] dataset, consisting of 5000 single illumination images, is used. Images are composed of various scenes, including indoors and outdoors. The dataset provides single illumination ground-truth for each image, recorded using a color checker object. To augment the dataset for multi-illumination, a semantic segmentation for each image is obtained, using Semantic Soft Segmentation [2]. The segmentation is then used to create the illumination map. For each illumination map obtained, random seed pixels from each of the illumination sections are selected. Thus, for each image, with  $N$  illuminants, there are  $N \times 2 + 2$  image pairs:  $N$  illumination color images,  $N$  seed point masks, the multi-illumination color biased image and the corresponding color corrected image.

During inference, the grayness index [37] is used to obtain the initial multi-illumination clusters on the MIMO dataset. These clusters are then used to obtain the initial seed points, which are then passed on to the network. All numbers reported, and unless specifically mentioned, are from networks trained completely on the augmented Intel TUT dataset and tested on the MIMO dataset.

## 4.2. Error Metrics

The angular error between the ground-truth illumination  $e$  and the estimated illumination  $\hat{e}$  is reported:

$$d_{ang} = \arccos\left(\frac{e \cdot \hat{e}}{\|e\| \times \|\hat{e}\|}\right), \quad (10)$$

where  $\|\cdot\|$  is the L2 norm. For each image, the angular error is computed, and the mean, median, trimean, means of the lowest-error 25%, the highest-error 25%, and maximum angular errors are reported.

For the architectures that do not provide an illumination estimation directly, like the single illumination GAN experiments, the apparent illumination from the color corrected image is obtained. This is done by inverting equation (1), as follows:

$$e = I * \hat{W}^{-1}, \quad (11)$$

where  $I$  is the input image and  $\hat{W}$  is the white balanced estimation from the network. Both  $I$  and  $\hat{W}$  are converted



Figure 3. Performance of different GAN models on the color constancy task. All models can recover proper white balanced images. The images are gamma adjusted for visualization.

Model	Mean	Med.	Tri.	Best 25%	Worst 25%	Max
Pix2Pix [29]	<b>6.6</b>	<b>5.3</b>	<b>5.4</b>	<b>1.4</b>	<b>14.2</b>	<b>36.0</b>
StarGAN [15]	10.3	8.9	9.0	4.0	18.9	46.0
CycleGAN [45]	8.4	5.9	6.4	1.5	19.6	37.8

Table 1. Performance of different GAN models for SFU Gray Ball dataset [16]. Pix2Pix achieves the best performance.

Model	Mean	Med.	Tri.	Best 25%	Worst 25%	Max
Pix2Pix [29]	3.6	2.8	3.1	1.2	<b>7.2</b>	<b>11.3</b>
StarGAN [15]	5.3	4.2	4.6	1.5	11.0	21.8
CycleGAN [45]	<b>3.4</b>	<b>2.6</b>	<b>2.8</b>	<b>0.7</b>	7.3	18.0

Table 2. Performance of different GAN models for ColorChecker RECOMMENDED dataset [19]. CycleGAN achieves the best performance.

Model	Mean	Med.	Tri.	Best 25%	Worst 25%	Max
Pix2Pix [29]	1.9	1.4	1.5	0.7	4.0	8.0
StarGAN [15]	3.8	3.3	3.5	1.3	7.0	11.4
CycleGAN [45]	<b>1.5</b>	<b>1.2</b>	<b>1.3</b>	<b>0.5</b>	<b>3.0</b>	<b>6.0</b>

Table 3. Performance of different GAN models for Cube dataset [4]. CycleGAN achieves the best performance.

into the linear  $RGB$  space before computing the estimated illumination.

## 4.3. Single Illumination Color Constancy

### 4.3.1 GAN Comparison

In this experiment, the performance of 3 different state-of-the-art GAN models for color constancy is given: Pix2Pix [29], CycleGAN [45] and StarGAN [15]. All models use  $sRGB$  inputs and generate a white balanced image. The color of the light source is estimated by Equation (11). The results are presented in Tables 1 to 3 for the different datasets. Figure 3 shows some sample outputs.

Tables 2 and 3 show that *CycleGAN* outperforms *Pix2Pix* and *StarGAN*. Table 1 shows that *Pix2Pix* achieves the best performance for the SFU Gray Ball dataset. *Pix2Pix* learns a per-pixel mapping of the illumination and is agnostic about the global illumination or the spatial re-



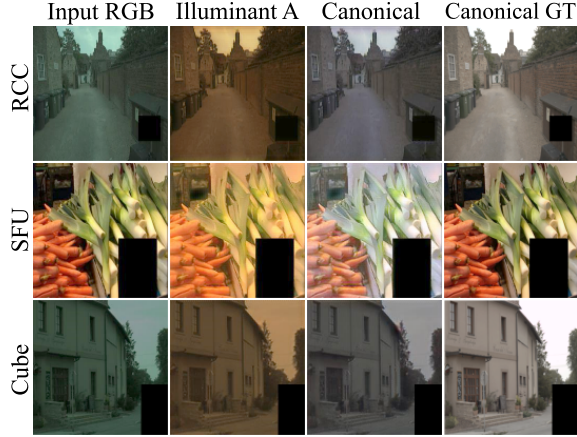


Figure 4. Visualisations of the StarGAN architecture. This architecture learns mappings to multiple domains, e.g., Illuminant A and canonical, simultaneously. The images are gamma adjusted for visualisation.

lations. On the other hand, the other GAN models learn a global representation of the scene illumination. Since CycleGAN can learn a more global transformation, it is able to outperform the other GANs for the color Checker and Cube dataset. However, for the Gray ball dataset the ground truth estimation is prone to errors. In conclusion, the global estimate of CycleGAN fits all the light sources under a single transformation function, while Pix2Pix can work around this by learning a per-pixel independent transformation.

StarGAN has much lower performance than the other 2 models. This may be due to the architecture’s latent representation that has to compensate for learning 3 different (input, canonical, Illuminant A) domain transformations. This is not the case with either Pix2Pix or CycleGAN, where the mapping is one-to-one. The multiple target illumination domain output of StarGAN can be beneficial, as shown in Figure 4, for relighting tasks. The accuracy of this internal estimation defines how close the output is to the target domain. Ideally, this estimated ground-truth illumination should be the same for all the output domains. Hence, by checking the performance of the output in estimating the ground-truth using Equation (11), the consistency of the network can be checked. This also allows for observing which illumination performs better in estimating the ground-truth illuminants. Per-illuminant consistency performance is presented in Table 4. It can be observed from the table that for the RECOMMENDED Color Checker (RCC) dataset *Illuminant A* estimation yields better results than the direct canonical estimation. For the Cube dataset, the canonical performs better. The Cube dataset consists of only outdoor images. Hence, the input images seldom have an illumination that is closer to Illuminant A. Conversely, for the RCC dataset, indoor scenes generally have incandescent lighting, which is closer to Illuminant A.

Dataset + Illumination	Mean	Med.	Tri.	Best 25%	Worst 25%	Max
RCC + Canonical	5.3	4.2	4.6	1.5	11.0	21.8
RCC + Illuminant A	2.9	2.2	2.3	0.8	6.4	13.8
Cube + Canonical	3.8	3.3	3.5	1.3	7.0	11.4
Cube + Illuminant A	4.1	3.2	3.4	1.1	8.5	20.7
SFU + Canonical	10.3	8.9	9.0	4.0	18.9	46.0
SFU + Illuminant A	12.7	11.2	11.2	3.2	24.8	61.1

Table 4. Consistency of StarGAN’s illumination estimation.

Method	Mean	Med.	Tri.	Best 25%	Worst 25%	Max
Grey-World [12]	13.0	11.0	11.5	3.1	26.0	63.0
Edge-based Gamut [5]	12.8	10.9	11.4	3.6	25.0	58.3
Spatial Correlations [13]	12.7	10.8	11.5	2.4	26.1	41.2
Second-order Grey-Edge [42]	10.7	9.0	9.4	3.2	20.9	56.0
Bottom-up & Top-down [43]	9.7	7.7	8.2	2.3	20.6	60.0
Natural Image Statistics [23]	9.9	7.7	8.3	2.4	20.8	56.1
E. B. color constancy [30]	8.0	6.5	6.8	2.0	16.6	53.6
Pix2Pix [29]	<b>6.6</b>	<b>5.3</b>	<b>5.4</b>	<b>1.4</b>	<b>14.2</b>	<b>36.0</b>
CycleGAN [45]	8.4	5.9	6.4	1.5	19.6	37.8
StarGAN [15]	11.4	9.2	10.0	3.8	21.8	41.4

Table 5. Performance on SFU Gray Ball [16].

Method	Mean	Med.	Tri.	Best 25%	Worst 25%	Max
Grey-World [12]	9.7	10.0	10.0	5.0	13.7	24.8
White-Patch [32]	9.1	6.7	7.8	2.2	18.9	43.0
Shades-of-Grey [20]	7.3	6.8	6.9	2.3	12.8	22.5
AlexNet + SVR [11]	7.0	5.3	5.7	2.9	14.0	29.1
Pixel-based Gamut [5]	6.0	4.4	4.9	1.7	12.9	25.3
Intersection-based Gamut [5]	6.0	4.4	4.9	1.7	12.8	26.3
Edge-based Gamut [5]	5.5	3.3	3.9	0.7	13.8	29.8
Deep Colour Constancy [11]	4.6	3.9	4.2	2.3	7.9	14.8
Second-order Grey-Edge [42]	4.1	3.6	3.8	1.5	8.5	16.9
First-order Grey-Edge [42]	4.0	3.1	3.3	1.4	8.4	20.6
FFCC (model Q) [7]	<b>2.0</b>	<b>1.1</b>	<b>1.4</b>	<b>0.3</b>	<b>4.6</b>	<b>25.0</b>
Pix2Pix [29]	3.6	2.8	3.1	1.2	7.2	<b>11.3</b>
CycleGAN [45]	3.4	2.6	2.8	0.7	7.3	18.0
StarGAN [15]	5.7	4.9	5.2	1.7	10.5	19.5

Table 6. Performance on ColorChecker RECOMMENDED [19].

### 4.3.2 Comparison to the State-of-the-Art

In this section, the GAN models are compared with different benchmarking algorithms including Grey-World [12], Second-order Grey-Edge [42], Pixel-based Gamut [5], Intersection-based Gamut [5], Edge-based Gamut [5], Spatial Correlations[13], Natural Image Statistics [23], High Level Visual Information [43], Exemplar-Based color constancy [30], Color Tiger [4]. Furthermore, comparisons to 2 convolutional approaches are given: Deep color constancy [11] and Fast Fourier color constancy [7]. Some of the results are taken from related papers, resulting in missing entries for some datasets. Tables 5 to 7 provide quantitative results for 3 different benchmarks.

It is shown in Table 5 (SFU Gray Ball), Pix2Pix [29] achieves the best performance in all the metrics with 17.5% improvement in mean angular error, 18.4% in median and 20.6% in trimean. For Table 6 (RCC), CycleGAN [45] framework achieves the second best place in different metrics, only worse than Fast Fourier color constancy [7]. In Table 7 (Cube), CycleGAN [45] achieves the best performance in the mean error and worst and comparable results

Method	Mean	Med.	Tri.	Best 25%	Worst 25%
Grey-World [12]	3.8	2.9	3.2	0.7	8.2
White-Patch [32]	6.6	4.5	5.3	1.2	15.2
Shades-of-Grey [20]	2.6	1.8	2.9	0.4	6.2
General Grey-World [6]	2.5	1.6	1.8	0.4	6.2
Second-order Grey-Edge [42]	2.5	1.6	1.8	0.5	6.0
First-order Grey-Edge [42]	2.5	1.6	1.8	0.5	5.9
Color Tiger [4]	3.0	2.6	2.7	0.6	5.9
Restricted Color Tiger [4]	1.6	<b>0.8</b>	<b>1.0</b>	<b>0.2</b>	4.4
Pix2Pix [29]	1.9	1.4	1.5	0.7	4.0
CycleGAN [45]	<b>1.5</b>	1.2	1.3	0.5	<b>3.0</b>
StarGAN [15]	4.8	3.9	4.2	1.9	8.9

Table 7. Performance on Cube dataset[4].

Dataset	Mean	Median
Cube	18.7	18.1
REC-CC	14.8	14.5
SFU	20.1	17.8

Table 8. Performance of Pix2Pix on the MIMO dataset.

Dataset	Mean	Median
Cube	12.7	11.9
REC-CC	13.3	12.9
SFU	20.1	17.4

Table 9. Performance of CycleGAN on the MIMO dataset.

with all other methods.

#### 4.3.3 GAN Performance for Multiple Light Sources

In this section, the performance of GANs in a multi-illumination setting is studied. The models are trained on 3 datasets and tested on the MIMO dataset. The results are shown in Tables 8 and 9. The performance of StarGAN is not provided since, from the experiments, it is observed: 1) to be generally the worst performing, and 2) CycleGAN is a simplified version of StarGAN with just one domain transform.

The results show that both the algorithms are unable to cope with images with multiple light sources. This is because both models are trained to learn a single illumination transformation. As a result, the models confuse object colors with illuminants. Pix2Pix shows slightly larger errors because no contextual constraints are imposed on the pixels. As such, ambiguous points like edges and boundaries contribute to higher errors. Inclusion of an explicit illumination context becomes a necessity for the network to be able to model a multi-illumination setting.

#### 4.4. Multi Illumination Color Constancy

In this section, the proposed method for multi-illumination color constancy is analysed. For all the experiments, four illuminants are chosen,  $N = 4$ . However, the proposed method is not dependent on the choice of  $N$  and can be arbitrarily scaled. All corresponding hyper-parameters of the model are kept constant. This is to ensure that the influence of the components being tested is not in-

Algorithm	Mean	Median
Without Adv. Loss	3.7	3.5
Proposed	<b>3.5</b>	<b>2.9</b>

Table 10. Influence of the GAN loss.

Algorithm	Mean	Median
Without Probability Output	5.30	5.24
Proposed	<b>3.5</b>	<b>2.9</b>

Table 11. Influence of a probability map as the output.

fluenced by a change in the hyper-parameters. For training, the entire set of 5000 images from Intel TUT is used, while testing is done exclusively on the MIMO dataset.

#### 4.4.1 Influence of Adversarial Loss

In this experiment, the influence of the adversarial loss is studied. Since there is a direct supervision from the ground truth, the network is trained in the traditional manner, where an encoder-decoder architecture is used to produce the output. Both the generator and the discriminator losses are removed and the network is trained using only the L1 losses on the predicted illumination map (5), the color corrected image (6) and the individual seed consistency (7). The results for the experiment are shown in table 4.4.1.

It is observed that omitting the adversarial loss reduces the performance. Adding the adversarial loss helps the model to cope with outliers, while also improving the average performance of the model. This can be explained as follows. The discriminator in a GAN is actively seeking for unrealistic outputs, which are often outliers. Furthermore, without the adversarial losses, the network only minimizes a distance function, which can lead to blurry outputs or ambiguous edges and borders. Since the proposed method is pixel-wise, these errors propagate more strongly through the network. In conclusion, this experiment shows that the inclusion of the adversarial loss is a necessary component for the proposed formulation.

#### 4.4.2 Influence of Illumination Probability Map Prediction

In the previous experiment, we modelled the problem as an image-to-image translation. In this experiment, we study such a formulation for the proposed method. For this, instead of predicting a per-pixel probability map, the color corrected image is directly predicted. For the illumination losses, the predicted illumination map is obtained by using the predicted color corrected image and the input color bias image itself. This experiment verifies whether adding the samples as an extra input to Pix2Pix improves the performance. It is also shown to what extent Pix2Pix can adapt to a multi-illumination setting. The results are given in table 4.4.2. Figure 5 shows some outputs.

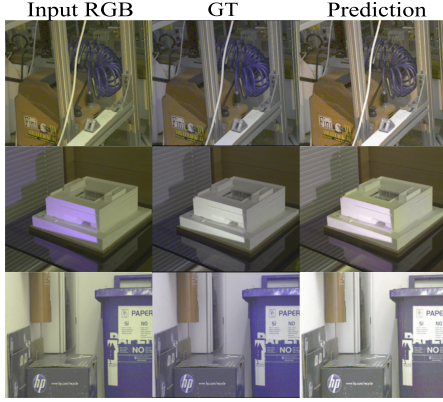


Figure 5. Comparison of the modified direct correction model to the ground truth. Here, the model directly outputs the color corrected image, instead of producing an illumination probability map. It is observed that the model is not able to fully correct for the illumination and as such, has residual illumination colors (2nd row, the printer.) This can be explained by the lack of the explicit probability map, which forces the network to instead predict a uniform illumination.

The results show that regressing the color corrected image directly using the proposed formulation yields higher errors. Since the model takes initial seed points and tries to constrain the output to be consistent with the seed points, the model fails to perform a full correction. This is because with a probability output, the initial seed points are directly linked to the illuminants. However, having a color corrected output needs the network to learn this probability implicitly. This, coupled with the illumination constraints on the loss functions, causes the network to leak the illumination over to the image directly. From the figure it can also be observed that the outputs are a bit more washed out and contain illumination color as residuals. This is because without the probability map support, the network compensates by averaging the illumination distribution more uniformly.

#### 4.4.3 Comparison to State of the Art

In this experiment, the performance of the proposed method is compared with the state-of-the-art methods. The results are shown in table 12. A visual comparison is also provided in figure 6. The Grayness Index algorithm is compared against 4-illumination clustering since the proposed method is tested on the 4-illumination assumption. A multi-illumination setting, like indoors, is more likely to have 4 or less major illuminants. Hence  $N$  is set to 4 for the experiments.

It is shown that the proposed model can outperform all the baselines by a comfortable margin. From the figure, it is observed that the proposed method can properly recover color corrected images. The proposed method can correct for the illumination on the printer, while Grayness

Algorithm	Mean	Median
Doing Nothing	8.9	8.8
[24]	3.8	4.2
CRF ([8])	4.1	3.3
GI(M=4) ([37])	4.0	3.5
Proposed	<b>3.5</b>	<b>2.9</b>

Table 12. Comparison to State-of-the-Art algorithms.

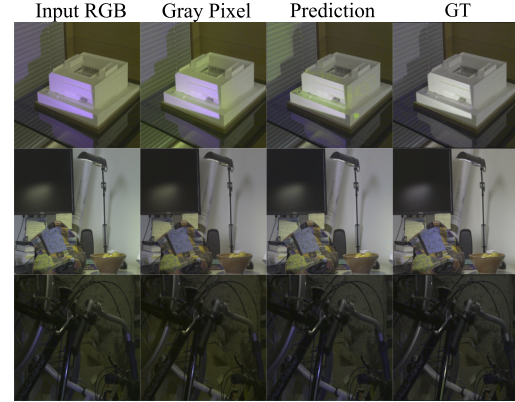


Figure 6. Comparison of the proposed method to sota methods. The proposed method can properly recover color corrected images. Gray Pixel can somewhat handle multi-illumination. However, it fails for ambiguous cases like the bike (3rd row). Gray Pixel fails to remove the yellow color cast of the light. The proposed method can recover such ambiguities.

Index classifies that as part of the object color. Similarly, Grayness Index is unable to color correct the reflection on the monitor while the proposed method can. The only requirement for the proposed model is the corresponding seed points for each illumination, only required for the training phase. This requirement can be easily satisfied as discussed previously using a single illumination dataset and augmenting it, without requiring any specialised hardware. For edge cases of more illuminants in the test image, compared to the  $N$  illumination number during the training, the network will simply collapse them into  $N$  illuminants. Explicit handling of such cases is beyond the scope of this work and can be a future research.

## 5. Conclusion

In this paper, we modelled the problem of color constancy as an image-to-image translation. GANs are used to learn domain transfers. A hybrid approach has been proposed. A single-illumination dataset augmentation has also been proposed to address the problem of the lack of large multi-illumination datasets to enable learning. Experiments on both single and multi-illumination datasets show that our methods outperform sota methods.

## References

- [1] Abdelrahman Abdelhamed, Abhijith Punnappurath, and Michael S. Brown. Leveraging the availability of two



- cameras for illuminant estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6637–6646, June 2021.
- [2] Yağız Aksoy, Tae-Hyun Oh, Sylvain Paris, Marc Pollefeys, and Wojciech Matusik. Semantic soft segmentation. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 37(4):72:1–72:13, 2018.
  - [3] Çağlar Aytekin, Jarno Nikkanen, and Moncef Gabbouj. INTEL-TUT dataset for camera invariant color constancy research. *CoRR*, abs/1703.09778, 2017.
  - [4] N. Banic and S. Loncaric. Unsupervised learning for color constancy. In *International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, 2018.
  - [5] K. Barnard. Improvements to gamut mapping color constancy algorithms. In *European Conference on Computer Vision*, 2000.
  - [6] K. Barnard, L. Martin, A. Coath, and B. Funt. A comparison of computational color constancy algorithms - part ii: Experiments with image data. *IEEE Transactions on Image Processing*, pages 985–996, 2002.
  - [7] J. T. Barron. Fast fourier color constancy. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.
  - [8] Shida Beigpour, C. Riess, Joost van de Weijer, and E. Angelopoulou. Multi-illuminant estimation with conditional random fields. *IEEE Transactions on Image Processing*, 23:83–96, 2014.
  - [9] S. Bianco, G. Ciocca, C. Cusano, and R. Schettini. Improving color constancy using indoor-outdoor image classification. *IEEE Transactions on Image Processing*, pages 1057–1149, 2008.
  - [10] S. Bianco, G. Ciocca, C. Cusano, and R. Schettini. Automatic color constancy algorithm selection and combination. *Pattern Recognition*, pages 695–705, 2010.
  - [11] S. Bianco, C. Cusano, and R. Schettini. Color constancy using cnns. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2015.
  - [12] G. Buchsbaum. A spatial processor model for object colour perception. *Sources of Color Science*, pages 1–26, 1980.
  - [13] A. Chakrabarti, K. Hirakawa, and T. Zickler. Color constancy with spatio-spectral statistics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1509–1519, 2012.
  - [14] Dongliang Cheng, Brian Price, Scott Cohen, and Michael S. Brown. Effective learning-based illuminant estimation using simple features. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1000–1008, 2015.
  - [15] Y. Choi, M. Choi, M. Kim, J. W. Ha, S. Kim, and J. Choo. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.
  - [16] F. Ciurea and B. Funt. A large image database for color constancy research. In *Color and Imaging Conference*, 2003.
  - [17] Partha Das, Anil S. Baslamisli, Yang Liu, Sezer Karaoglu, and Theo Gevers. Color constancy by gans: An experimental survey. *CoRR*, abs/1812.03085, 2018.
  - [18] M. D. Fairchild. *Color Appearance Models*. Wiley, Chichester, U.K., 2013.
  - [19] G. D. Finlayson, G. Hemrit, A. Gijsenij, and P. Gehler. A curious problem with using the colour checker dataset for illuminant estimation. In *Color and Imaging Conference*, 2017.
  - [20] G. D. Finlayson and E. Trezzi. Shades of gray and colour constancy. In *Color and Imaging Conference*, 2004.
  - [21] B. Funt and Weihua Xiong. Estimating illumination chromaticity via support vector regression. In *Color Imaging Conference*, 2004.
  - [22] R. Gershon, Allan Jepson, and John Tsotsos. From  $[r, g, b]$  to surface reflectance: Computing color constant descriptors in images. In *International Joint Conferences on Artificial Intelligence*, pages 755–758, 01 1987.
  - [23] A. Gijsenij and T. Gevers. Color constancy using natural image statistics and scene semantics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 687–698, 2010.
  - [24] A. Gijsenij, T. Gevers, and J. van de Weijer. Improving color constancy by photometric edge weighting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(5):918–929, May 2012.
  - [25] A. Gijsenij, R. Lu, and T. Gevers. Color constancy for multiple light sources. *IEEE Transactions on Image Processing*, 21(2):697–707, 2012.
  - [26] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, 2014.
  - [27] Y. Hu, B. Wang, and S. Lin. Fc4: Fully convolutional color constancy with confidence-weighted pooling. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.
  - [28] M. A. Hussain and A. S. Akbari. Color constancy algorithm for mixed-illuminant scene images. *IEEE Access*, 6:8964–8976, 2018.
  - [29] P. Isola, J. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial nets. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.
  - [30] H. R. V. Joze and M. S. Drew. Exemplar-based color constancy and multiple illumination. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 860–873, 2014.
  - [31] E. H. Land. The retinex theory of color vision. *Scientific American*, pages 108–129, 1977.
  - [32] E. H. Land and J. J. McCann. Lightness and retinex theory. *Journal of the Optical Society of America*, pages 1–11, 1971.
  - [33] Z. Lou, T. Gevers, N. Hu, and M. P. Lucassen. Color constancy by deep learning. In *British Machine Vision Conference*, 2015.
  - [34] X. Mao, C. Shen, and Y. Yang. Image restoration using very deep fully convolutional encoder-decoder networks with symmetric skip connections. In *Advances in Neural Information Processing Systems*, 2016.
  - [35] S. W. Oh and S. J. Kim. Approaching the computational color constancy as a classification problem through deep learning. *Pattern Recognition*, pages 405–416, 2017.

- [36] Y. Qian, K. Chen, J. Nikkanen, J. K. Kamarainen, and J. Matas. Recurrent color constancy. In *IEEE International Conference on Computer Vision*, 2017.
- [37] Yanlin Qian, Jarno Nikkanen, Joni-Kristian Kämäräinen, and Jiri Matas. On finding gray pixels. *CoRR*, abs/1901.03198, 2019.
- [38] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference On Medical Image Computing & Computer Assisted Intervention*, 2015.
- [39] C. Rosenberg, T. Minka, and A. Ladsariya. Bayesian color constancy with non-gaussian models. In *Advances in Neural Information Processing Systems*, 2004.
- [40] Oleksii Sidorov. Conditional gans for multi-illuminant color constancy: Revolution or yet another approach? *CoRR*, abs/1811.06604, 2018.
- [41] S. Tan, F. Peng, H. Tan, S. Lai, and M. Zhang. Multiple illumination estimation with end-to-end network. In *IEEE International Conference on Image, Vision and Computing*, 2018.
- [42] J. van De Weijer, T. Gevers, and A. Gijsenij. Edge-based color constancy. *IEEE Transactions on Image Processing*, pages 2207–2214, 2007.
- [43] J. van de Weijer, C. Schmid, and J. Verbeek. Using high-level visual information for color constancy. In *IEEE International Conference on Computer Vision*, 2007.
- [44] J. von Kries. Influence of adaptation on the effects produced by luminous stimuli. *Sources of Color Science*, pages 109–119, 1970.
- [45] J. Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *IEEE International Conference on Computer Vision*, 2017.