

SEMANTIC COLOR CONSTANCY

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

İHSAN EMRE ÜSTÜN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
ELECTRICAL AND ELECTRONICS ENGINEERING

FEBRUARY 2021

Approval of the thesis:

SEMANTIC COLOR CONSTANCY

submitted by **İHSAN EMRE ÜSTÜN** in partial fulfillment of the requirements for the degree of **Master of Science in Electrical and Electronics Engineering Department, Middle East Technical University** by,

Prof. Dr. Halil Kalıpçılar
Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. İlkay Ulusoy
Head of Department, **Electrical and Electronics Engineering**

Prof. Dr. A. Aydın Alatan
Supervisor, **Electrical and Electronics Engineering, METU**

Examining Committee Members:

Prof. Dr. Gözde Bozdağı Akar
Electrical and Electronics Engineering, METU

Prof. Dr. A. Aydın Alatan
Electrical and Electronics Engineering, METU

Assoc. Prof. Dr. Fatih Kamışlı
Electrical and Electronics Engineering, METU

Prof. Dr. Ahmet Oğuz Akyüz
Computer Engineering, METU

Assoc. Prof. Dr. Seniha Esen Yüksel
Electrical and Electronics Engineering, Hacettepe University

Date:

12.02.2021



I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Surname: İhsan Emre Üstün

Signature :

ABSTRACT

SEMANTIC COLOR CONSTANCY

Üstün, İhsan Emre

M.S., Department of Electrical and Electronics Engineering

Supervisor: Prof. Dr. A. Aydın Alatan

FEBRUARY 2021, 92 pages

Color constancy aims to perceive the actual color of an object, disregarding the effect of the light source. Recent works showed that utilizing the semantic information in an image enhances the performance of the computational color constancy methods. Considering the recent success of the segmentation methods and the increased number of labeled images, we propose a color constancy method that combines individual illuminant estimations of detected objects which are computed using the classes of the objects and their associated colors. Then we introduce a weighting system that values the applicability of the object classes to the color constancy problem. Lastly, we introduce another metric expressing the detected object and how well it fits the learned model of its class. Finally, we evaluate our proposed method on a popular color constancy dataset, confirming that each weight addition enhances the performance of the global illuminant estimation. Experimental results show promising results, outperforming the conventional methods while competing with the state of the art methods.

Keywords: Color Constancy, Object Recognition, Color Space, Object Color Char-

acteristics



ÖZ

SEMANTİK RENK DEĞİŞMEZLİĞİ

Üstün, İhsan Emre

Yüksek Lisans, Elektrik ve Elektronik Mühendisliği Bölümü

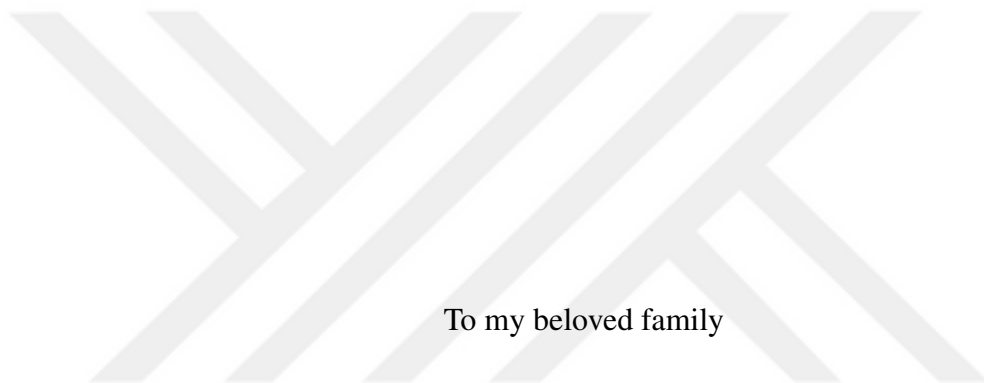
Tez Yöneticisi: Prof. Dr. A. Aydın Alatan

Şubat 2021 , 92 sayfa

Renk değişmezliği bir nesneyi ışık kaynağından bağımsız bir şekilde, sabit bir renkte algılayabilme kabiliyetidir. Son zamanlarda yapılan çalışmalar, görüntülerdeki semantik bilgilerden faydalanmanın renk değişmezliği algoritmalarının performansını arttırabildiğini göstermiştir. Son zamanlarda nesne bölütleme algoritmalarının artan başarısı ve işaretli veri kümesi sayısındaki artış düşünülerek, tespit edilen nesneler ve onlara atanan renklere dayanan, nesne tabanlı bir renk değişmezliği algoritması önerilmiştir. Ardından, bu sisteme her bir nesne sınıfının renk değişmezliği problemine uygunluğu gözetilerek bir ağırlık getirilmiştir. Son olarak, tespit edilen nesnelerin güvenilirliği, nesnenin sınıfının probleme uygunluğuna ilaveten, tespit edilen nesnenin ait olduğu sınıfa benzerliği de kullanılarak güncellenmiştir. Önerilen yöntem popüler bir renk değişmezliği veri kümesinde test edilmiş, yöntemeye yapılan her eklemenin performansta bir artışla sonuçlandığı gözlenmiştir. Deneyler gelecek vadeden sonuçlar göstermiş olup, geleneksel metodlardan daha başarılı sonuçlar elde ederken, güncel başarılı algoritmalarla yarışabilen bir performans göstermiştir.

Anahtar Kelimeler: Renk Doğruluđu, Nesne Tanıma, Renk Uzayı, Nesne Renk Karakteristiđi





To my beloved family

ACKNOWLEDGMENTS

First of all, I would like to thank my supervisor Prof. Dr. A. Aydın Alatan for his great support, continuous encouragement, and guidance during my studies. He was always easy to communicate with and very understanding. In addition to his academic stance, he was also a good role model in terms of how he approaches both human relationships and ethical concerns. He provided a great lab environment which is more like a home than a work environment where I met wonderful people.

I would like to express my thanks to all jury members for their comments and criticism.

I also would like to present special thanks to my family for their unlimited support. The biggest portion of the thank belongs to my lovely mother who is the hidden hero of this thesis. She listened to my daily plans, the obstacles I ran into, my plans for my research with all the details even though the details did not make sense to her and always supported me. I also want to present my thanks to my father for his unlimited support to all my decisions. Even the ones he did not approve of. He taught me the most of the things that now define me. For every decision I made, I felt his support which gave me courage. Thanks to my brother for always doing his best when I ask and laughing at my terrible jokes. I cannot think of a better brother than him. Also, I want to thank my other family members for all their support. Especially during the quarantine, even though we were far apart, I felt their presence and support.

I also would like to thank my friend Özge Cansu for her genuine friendship. She has the kindest soul I've ever known. Thanks to Oğuz Ünlüerler, my childhood friend for always making me laugh. Even though we have been far away from each other for the last ten years, he is still the person to call for me when I needed to talk. I cannot think of a better definition of a friend. Thanks to Buse Alagöz for her friendship. It has always been a pleasure to spend time with her and listen to her interesting opinions. Thanks to Can Çakırgöz for his help and contributions to my thesis. His

assistance to my study was a lifesaver. Thanks to Nazlı Gökarp for her friendship and her artistic touch on my thesis. She always did her best when I was in need. Thanks to Esat Kalfaoğlu for his friendship and. He is a very good friend always eager to help people around him.

Lastly, I would like to thank to Karabaş and his pack for all the joy and comfort they brought to my life.



TABLE OF CONTENTS

ABSTRACT	v
ÖZ	vii
ACKNOWLEDGMENTS	x
TABLE OF CONTENTS	xii
LIST OF TABLES	xv
LIST OF FIGURES	xvi
CHAPTERS	
1 INTRODUCTION	1
1.1 Motivation	1
1.2 Scope of the Thesis	3
1.3 Outline of the Thesis	4
2 RELATED BACKGROUND	5
2.1 Color Constancy Problem	5
2.1.1 Problem Formulation	7
2.1.2 Assumptions	7
2.1.2.1 Single Distant Point Light Source	8
2.1.2.2 Diagonal Transformation	9
2.1.2.3 Body Reflectance Assumption	10

2.1.3	Lambertian Surface	11
2.1.4	Evaluation Metric	12
2.2	Related Literature	13
2.2.1	Statistics-based Methods	14
2.2.1.1	Low-level Statistics-based Methods	15
2.2.1.2	Physics-based Methods	18
2.2.2	Learning-based Methods	19
2.2.2.1	Probabilistic Methods	19
2.2.2.2	Gamut-based Methods	20
2.2.2.3	Methods using Neural Networks	22
2.2.3	Methods Utilizing Semantic Information	28
3	COLOR COHERENCE OF DIFFERENT OBJECT CATEGORIES	45
3.1	Motivation	45
3.2	Proposed Color Similarity Metrics	47
3.2.1	Selection of Color Space	48
3.2.2	Individual Class Measurements	50
3.2.3	Intra-class Average Color Distance	55
3.2.4	Intra-class Color Scatter Distance	58
3.3	Experimental Analysis of Color Coherence with Object Categories	59
3.3.1	Utilized Dataset	59
3.3.2	Comparison of Object Classes Based on Categories	62
3.3.3	Discussion	66
4	PROPOSED SEMANTIC COLOR CONSTANCY MODEL	69

4.1	Proposed Method: Semantic Color Constancy	70
4.1.1	Variants of the Proposed Color Constancy Method	71
4.2	Experimental Results	74
4.2.1	Experimental Setup	74
4.2.2	Individual Semantic Class Performances for Color Constancy .	76
4.2.3	Experimental Results on the Color-COCO Dataset	77
4.2.4	Effect of the Segmentation Mask on Algorithm Performance .	79
4.2.5	Experimental Results on Color Constancy Dataset	79
4.3	Discussion	83
5	SUMMARY & CONCLUSIONS	85
5.1	Summary	85
5.2	Conclusions	86
	REFERENCES	87
	APPENDICES	

LIST OF TABLES

TABLES

Table 2.1	Instantiations of Equation 2.12 with different settings of (n, p, σ)	18
Table 3.1	Top 10 classes based on intra-class average color distances	55
Table 3.2	Bottom 10 classes based on intra-class average color distances	56
Table 4.1	Top 10 classes with the lowest angular error on the Color-COCO dataset	76
Table 4.2	Top 10 classes with the highest angular error on the Color-COCO dataset	77
Table 4.3	Results on the Color-COCO dataset	79
Table 4.4	Results on the Color-COCO dataset with Detectron-2 [60] masks	80
Table 4.5	Results on REC	81

LIST OF FIGURES

FIGURES

Figure 1.1	A simple example of image formation in a scene illuminated by a light source	2
Figure 2.1	Poorly white balanced images and their correctly white balanced versions	6
Figure 2.2	A demonstration of a single distant light source.	9
Figure 2.3	An illustration of reflection of incoming light onto a lambertian surface	12
Figure 2.4	Taxonomy of color constancy methods	14
Figure 2.5	Example scenes where Gray World algorithm fails	16
Figure 2.6	A general overview of application of gamut mapping	21
Figure 2.7	Graphical representation of local estimators in [9] [10]	23
Figure 2.8	Schematic expression of [10]	24
Figure 2.9	Proposed architecture of DS-Net [54]	25
Figure 2.10	Proposed architecture of Bag of Color Features[42]	26
Figure 2.11	Schematic representation of [14]	27
Figure 2.12	Segmentation and evaluation of image regions [31]	28
Figure 2.13	Working scheme of [8]	29

Figure 2.14	Visual representation of the proposed method by Weijer et al. [58]	31
Figure 2.15	Image processing pipeline proposed by [50]	32
Figure 2.16	Network structure of [1]	33
Figure 2.17	Convex hull of face skin pixels in HS plane of HSV color space [11]	34
Figure 2.18	Adaptive thresholding of the skin pixels [11]	35
Figure 2.19	Slightly tinted images and their respective histograms in uv log- chrominance space. [5]	36
Figure 2.20	Visualization of filters of CCC [5]	37
Figure 2.21	An overview of the proposed illuminant estimation process of CCC [5]	38
Figure 2.22	An analogy of the new approach of FFCC [7]	39
Figure 2.23	A demonstration of illuminant estimation using a wrapped his- togram [7]	39
Figure 2.24	The architecture of <i>AlexNet</i> – FC^4 [39]	40
Figure 2.25	A detailed examination of illuminant estimation process of FC^4 [39]	41
Figure 2.26	Test outputs of FC^4 [39] on Reprocessed ColorChecker Dataset	42
Figure 2.27	Visualization of C^4	43
Figure 3.1	Sample images from Microsoft COCO [45]	46
Figure 3.2	Example uv histogram representations of images from COCO dataset [45]	48
Figure 3.3	uv histograms with different parameters	51
Figure 3.4	Histogram representations of some images from REC [35]	52

Figure 3.5	A representation of class measurement calculation	54
Figure 3.6	Histogram of best two and worst two classes based on intra-class average color distances	57
Figure 3.7	Examples images containing Macbeth Color Checker [35]	60
Figure 3.8	Example images from the utilized dataset	61
Figure 3.9	Comparison of different object classes based on their context . .	63
Figure 3.10	Comparison of different object classes based on their nature . . .	64
Figure 3.11	Comparison of different object classes based on their types . . .	65
Figure 4.1	A visual representation of the proposed method	70
Figure 4.2	Example images from COCO [45]	75
Figure 4.3	Example images from REC [35]	76
Figure 4.4	Example images by application of the proposed algorithm on the utilized dataset	78
Figure 4.5	Segmentation results on REC [35] by Detectron-2 [60]	82
Figure 4.6	Example images by application of the proposed algorithm on REC [35]	83

CHAPTER 1

INTRODUCTION

Color constancy is simply the problem of observing an object invariant of the ambient light. Throughout the years, various approaches are proposed varying in terms of underlying assumptions, applicability, and computational complexity. In the scope of this thesis, we focus on intrinsic properties of object classes in real life and how these properties can be used as a clue for illuminant estimation in color constancy problem.

In this thesis, color coherence of object classes are analyzed and a color constancy method is proposed based on the the object classes and their applicability to the problem.

1.1 Motivation

The color of an observed image depends on its surface properties and the light sources existing in the scene. Any change in the spectral power density of the light source results in a change in the observed color of the image. The human visual system is evolved in a way that the color of the object tends to remain stable even under different light sources and perceived as it is observed under a white light source. It has also been argued that familiarity of colors of some object classes might be a clue for the human visual system. This quite complex mechanism is called Color Constancy. However, this mechanism does not exist in imaging devices and the color of the object highly depends on the illuminant source.

Color Constancy is an underdetermined problem in which only the power of the light reflected onto the camera is known, whereas surface reflectance properties and the

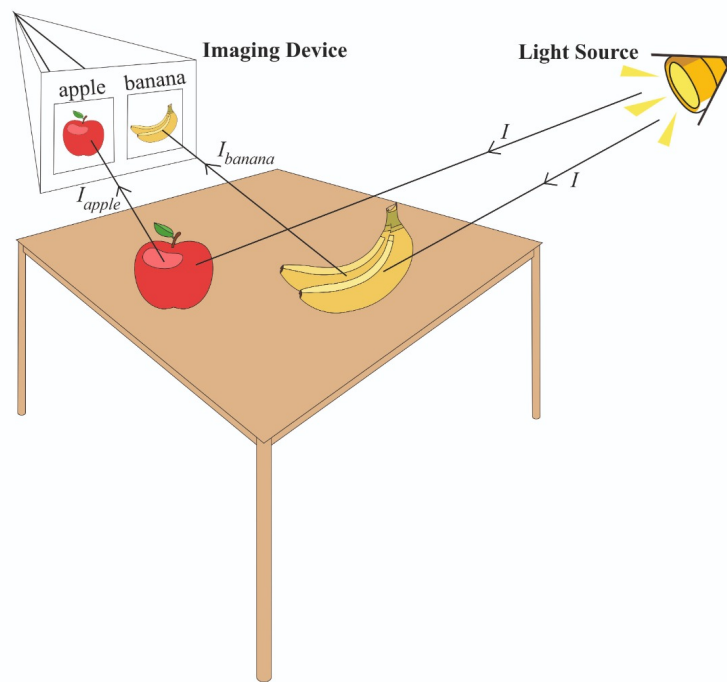


Figure 1.1: A simple example of image formation in a scene illuminated by a light source

color of the light source are unknown. In order to solve this ill-posed problem, some further assumptions, or a prior information, about the scene is strictly necessary. A simple example is an image consisting of only a red wall. The color of the wall might be a result of a white wall under red light or it might be a red wall under white light; both of these cases might be true, unless additional information about the image is known. However, if there is some grass in front of the wall, due to our previous encounters with grass we have some prior reliable information about its color content; something most likely close to the green color. Thus any shift in the observed color of grass pixels towards red chromatic band should be caused by the light source.

There are a few previous efforts [50] [58] trying to make use of prior information about the color of an object class and to use them as prior information for increasing the performance of the light source spectra estimation. Moreover, some other methods fed their algorithm with semantic segmentation results of the image and enhanced illuminant estimation performance. These studies showed that utilization of the semantic information in an image enhances the performance of the estimation.

1.2 Scope of the Thesis

In this thesis, we analyze the common object classes by how they are presented in various images from a popular segmentation dataset COCO [45] that includes labeled segmentation masks for each class. First, we present color space independent of the pixel brightness; then, we propose two similarity metrics to define object classes which are later used to define how reliable the estimation of a detected object. The first metric is the average color of the objects and the second one is their color distribution in a joint histogram. We compare different classes in terms of their applicability in the color constancy problem and proposed a novel illuminant estimation method that combines the weighted assumption of illuminant estimations of individual objects detected in the image. The weighting system compares the individual illuminant estimations in terms of the applicability of the class of the object and how well the object fits its class' learned model.

1.3 Outline of the Thesis

In this thesis, the related background is given in Chapter 2. At first, the color constancy problem is defined with a problem formulation and its common assumptions. Following the problem definition, an overview of the literature is presented. While most of the past work were given with brief explanations, methods related to the scope of the thesis are explained in detail. In Chapter 3, evaluation procedure of the object classes are given. First, the selected color space and its histogram is presented. The histogram is followed by definitions of similarity metrics for object classes. The object classes are then discussed based on their computed similarity metrics. Lastly, object classes are compared in terms of their similarity metrics under generalized categories. in Chapter 4, a method for illuminant estimation based on object classes and the evaluation of the proposed method on two datasets are given. Finally, the thesis is concluded with Chapter 5 which includes the summary of the thesis and its concluding remarks.

CHAPTER 2

RELATED BACKGROUND

This chapter presents an overview of the color constancy problem and analyzes the literature related to the scope of this thesis. The first section includes the problem definition, problem formulation, and commonly accepted assumptions. Following the problem definition, literature related to the problem are presented. A general overview of approaches and methods in the field is given briefly. On the other hand, works that are related to our study are analyzed in more detail.

2.1 Color Constancy Problem

Color is an important feature of computer vision problems and its applications. Colors presented in an image are determined by the light sources in the scene, intrinsic properties of the objects in the image, and the sensor type of the imaging device. Color constancy is the ability to discard the effect of the light sources in the scene so the color of an object can remain stable under the effect of variant light sources.

The human visual system has the ability of maintaining the color constancy thus shows a tendency to perceive an object as it is under a white light source, by filtering out the effect of the light sources in the scene. How this mechanism works is still not fully understood [41]. On the other hand, in vision-based systems stability of the color of an object is still desired to preserve its color under changing ambient light. This mechanism is usually implemented in the pipeline of image signal processor (ISP) of the imaging device, using sensor gains. This process is often referred to as White Balance (WB), which focuses on removing the effect of the light source and forming the image as it is taken under a white light source.



Figure 2.1: Poorly white balanced images and their correctly white balanced versions [35].

In Figure 2.1, raw images without a white balance process and their corrected versions are presented. Images on the first column are the input images taken from ColorChecker Dataset [35] and the images on the second column are the corrected versions using the ground truth illumination which is derived from the Macbeth Chart in the image. The images are gamma corrected for visualization purposes.

A great majority of the computational models for color constancy focus on estimating the effect of the light source. Although there are some methods that achieve color constancy without specifically focusing on the scene illumination. For example, Funt et al. [29] algebraically manipulates the image and derives illumination independent quantities of the image. In the scope of the thesis, estimating the illuminant source is the main focus point considering this study focuses on color constancy's applications on the imaging devices.

2.1.1 Problem Formulation

Klinker et al. [40] propose dichromatic reflection model to explain the relationship between the real world and its corresponding points in an image. The pixel values of the image ($f(x) = [f_R, f_G, f_B]^T$) depends on the light source ($I(x, \lambda)$), surface reflectance characteristic of the object ($S(x, \lambda)$) and the sensor spectral sensitivities ($\rho(\lambda) = (\rho_R, \rho_G, \rho_B)^T$) where x is the spatial coordinate and λ is the wavelength of the light. This formation can be derived as in Equation 2.1 where w is the visible spectrum. Most imaging devices have three distinct classes of sensor thus 3 different responses per pixel. Throughout the thesis, the imaging system will be assumed as trichromatic and its sensor responses will be assumed to be R, G, B as the majority of previous work in the literature has done. The reflected light consists of body and specular reflectances. While body reflectance depends on the surface it is reflected, specular reflectance does not get affected by the surface properties. Their relative scale factors are added as m_b and m_s respectively.

$$f(x) = m_b \int_w I(x, \lambda) S(x, \lambda) \rho(\lambda) + m_s \int_w I(x, \lambda) \rho(\lambda) \quad (2.1)$$

Our goal is to estimate $\mathbf{I}$, the projection of the illumination of the light source on the sensor spectral sensitivities.

$$\mathbf{I}(\mathbf{x}) = \int_w I(x, \lambda) \rho(\lambda) \quad (2.2)$$

2.1.2 Assumptions

Color constancy is an under-determined problem. Only the pixel values of the image are known whereas illumination source, surface intrinsic properties, and the sensor spectral sensitivities are unknown. In order to estimate the illuminant source characteristics, either some prior information or some further assumptions are needed.

Despite the fact that using the dichromatic model for the color constancy problem is more realistic, in practice, it is often simplified further based on various assumptions.

Most of the successful approaches use both the simplified model and some prior information. In this section, a general overview of the common assumptions used in the field is stated and briefly explained.

2.1.2.1 Single Distant Point Light Source

In a scene, objects might be affected by a number of light arrays. These arrays might be coming directly from a light source or might be reflected from another surface. Some areas might not be even getting any light from any of the light sources.

In the scope of the thesis, we assumed that there is only one distant point light source in the scene. This assumption enables us to model the light as it is coming from a single angle from a single light source with constant power onto all of the surfaces. This assumption leads to an important simplification. In terms of color, the effect of the light source on all surfaces in the scene are constant. Although the light source's power might vary depending on the angle between the light source and the surface, it only affects the pixel brightness, and up to some extent it is irrelevant to the color ratios. Consequently, every pixel in the image gets affected equally by the illuminant source from color perspective.

Throughout this study, we also assumed that there is no shadowing effect in the scene and light falling on a surface solely depends on the illuminant source. However, in reality this is rarely the case. Although this assumption is not realistic and invalid in real world cases, it performs well in most cases and accepted by the majority of the state of art methods.

The majority of the state of the art approaches assume there is only one illuminant source and a uniform illumination is presented in the image. However, there are also some studies focus on multiple light sources [34] or extend their methods so that they can adapt on multiple light sources [10].

A preferred solution is to split a single image into patches that are likely to have a stationary illumination on their pixels. Following the segmentation, estimating illumination for every patch via various methods and derive a conclusion on illuminant sources' effect on the image. Gijssen et al. [34] used illuminant estimation meth-

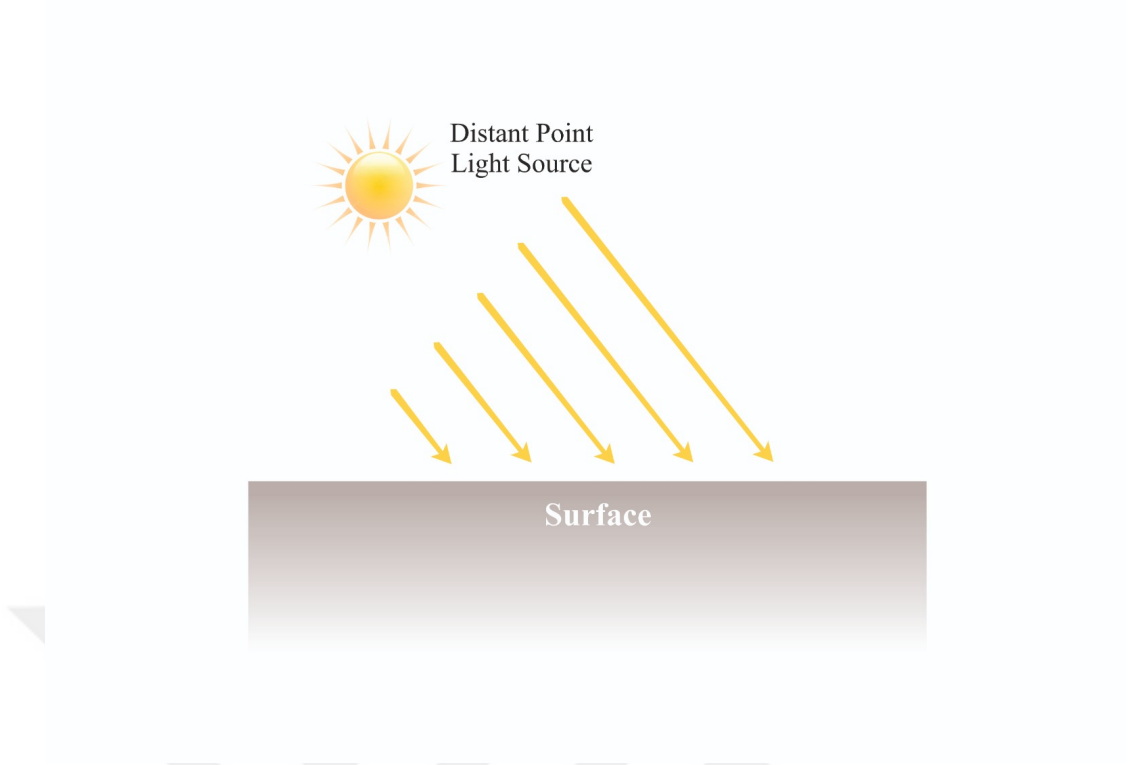


Figure 2.2: A demonstration of a single distant light source.

ods based on a single light source to locally correct an image under multiple light sources. They showed that as the chromatic difference of the light sources increases, their method outperforms the methods based on uniform light source assumption.

On the other hand, Bianco et al. extended their method [9] for multiple light sources. They locally estimated the illumination source's spectral distribution and use these estimations to decide, if there is more than one light source in the image. If they conclude that multiple light sources are affecting the scene, they propose multiple corrections for multiple areas for the input image [10].

2.1.2.2 Diagonal Transformation

Before studying on a color constancy problem, a mapping between the light and camera sensor outputs should be defined. In the scope of this thesis, our model assumes that color constancy can be achieved by simply adjusting the imaging system's distinct sensors. Transformation can be seen in Equation 2.3 where f_a and f_b are the

sensor responses under two different illuminations and $D = \text{diag}([d_1, d_2, d_3])$ is the diagonal transformation matrix.

$$\begin{bmatrix} f_b^1 \\ f_b^2 \\ f_b^3 \end{bmatrix} = \begin{bmatrix} d_1 & 0 & 0 \\ 0 & d_2 & 0 \\ 0 & 0 & d_3 \end{bmatrix} \begin{bmatrix} f_a^1 \\ f_a^2 \\ f_a^3 \end{bmatrix} \quad (2.3)$$

In fact, the above equation is only valid, when a certain number of regulations are presented. The assumption relies on two main conditions. Either the sensors should be responsive only to a narrow-band or the light source should not be broad-band. In reality, usually, both of these conditions barely hold. Additionally, the spectral sensitivities of the sensors might be overlapping. Although the assumption is merely a reflection of the truth, the diagonal adaptation model is an effective and widespread assumption. Throughout this thesis, Von-Kries Transformation [59] model shown in Equation (2.3) is used, resulting in each channel of the image to be adapted individually.

There is a great variety of camera sensors in reality and each of them has a particular spectral sensitivity. Sensors in the imaging device strongly affect the observed image in the case of an illuminant effect. Finlayson et al. [21] showed that when a pre-calculated linear transformation model that depends on the spectral sensitivities of the sensors is applied prior to the color constancy application, the success of an algorithm can be increased. This thesis does not focus on a set of sensors and uses various camera outputs focusing on the general problem. As a result, Von-Kries model (Equation 2.3) is accepted for the rest of our study.

2.1.2.3 Body Reflectance Assumption

The dichromatic model claims that a surface reflectance consists of two parts: Body reflectance and specular reflectance. The former one depends on the power of the light source, arriving angle between the light source and the surface, and the intrinsic properties of the object. The latter one solely relies on the power of the light source and the angle between the surface and the light source. It is independent of the object's

intrinsic properties.

For example, physics-based methods which are briefly explained in Section 2.2.1.2 focus on utilizing the dichromatic model. On the other hand, in the scope of this thesis and a great majority of the methods explained in Section 2.2 ignore the specular reflectance part of the reflection model and assume that only body reflectance is responsible for the reflected light. After this assumption is applied to dichromatic model in (2.1) by using $m_s = 0$, Equation (2.4) is obtained as follows:

$$f_c(x) = m_b \int_w I(x, \lambda) S(x, \lambda) \rho(\lambda) \quad (2.4)$$

The fact that the color of an object depends on the spectral distribution of the light and not its brightness level, m_b loses its significance. If we use the diagonal model and a trichromatic imaging system with three distinct sensor responses ($[R, G, B]$ as previously mentioned, Equation 2.4 can be deduced into Equation 2.5 where $f_c(x)$ is the pixel value for colors $c = R, G, B$ at location x in the image in imaging device coordinate system ($0 \leq x \leq \text{number of pixels}$), $\mathbf{I}_c$ is the illuminant source's projected power (Equation 2.2) for channel c and $S_c(x)$ is the surface reflectance ratio for color c of the surface in real world corresponding location x in camera coordinates.

$$f_c(x) = \mathbf{I}_c S_c(x) \quad \text{where } c = \{R, G, B\} \quad (2.5)$$

After this point, unless otherwise is stated, Equation 2.5 is accepted as the formalization of the color constancy problem.

2.1.3 Lambertian Surface

In the scope of this thesis, the scene surfaces in the image are assumed to have Lambertian reflectance property. If a surface has the Lambertian reflectance property, it reflects a constant amount of the incoming light to all directions (Figure 2.3). As a result, image pixel values ($f(x) = [f_R, f_G, f_B]^T$) are independent from the relative position of the camera and object's surface.

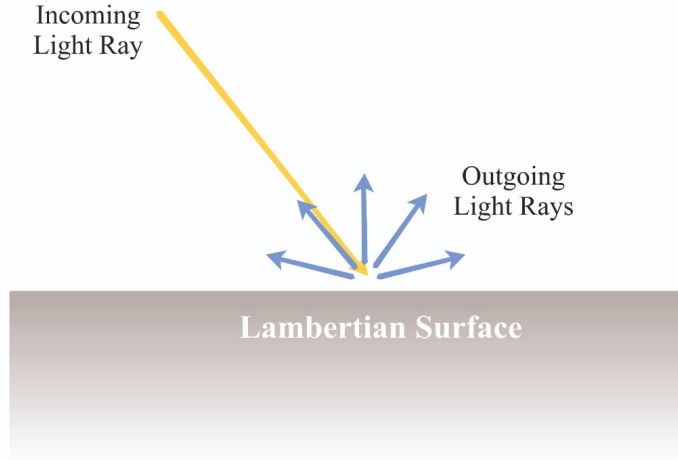


Figure 2.3: An illustration of reflection of incoming light onto a lambertian surface

2.1.4 Evaluation Metric

There is a wide range of approaches to solve the color constancy problem. In order to quantitatively compare these methods, there are two main metrics preferred. The first metric, which is also used more commonly, is the *recovery error*, suggested by Finlayson et al.[38]. This metric measures the angular error e_{rec} between the ground truth illumination array $\mathbf{I} = [I_r, I_g, I_b]^T$ and the estimated illumination $\hat{\mathbf{I}} = [\hat{I}_r, \hat{I}_g, \hat{I}_b]^T$.

$$e_{rec} = \arccos \left(\frac{\mathbf{I} \cdot \hat{\mathbf{I}}}{\|\mathbf{I}\| \|\hat{\mathbf{I}}\|} \right) \quad (2.6)$$

Recently, Finlayson [25] proposed a new evaluation metric, *reproduction error*. Instead of comparing the estimated illumination and the ground truth illumination, they proposed to measure the angle between the reconstructed white surface and the true white surface. This new metric focuses on comparing algorithms considering how

their estimations are used in real life and evaluates how well the image is corrected. Reproduction error metric e_{rep} is shown in Equation 2.8 where S_w is the true white surface and $S_{w,Est}$ is the corrected white surface. Using diagonal model and simply correcting the pixels by dividing them individually, the estimated white surface $S_{w,Est}$ can be written as in Equation (2.7) by using estimated illumination $\hat{\mathbf{I}} = [\hat{I}_r, \hat{I}_g, \hat{I}_b]^T$ and observed white surface S'_w .

$$S_{w,Est} = \left[\frac{S'_{w,R}}{\hat{I}_R}, \frac{S'_{w,G}}{\hat{I}_G}, \frac{S'_{w,B}}{\hat{I}_B} \right] \quad (2.7)$$

$$e_{rep} = \arccos \left(\frac{S_{w,Est} \cdot S_w}{\|S_{w,Est}\|, \|S_w\|} \right) \quad (2.8)$$

Throughout the thesis the recovery error (Equation 2.6) is used as the main evaluation metric. It should be noted that the recovery error is still the most commonly used metric in the literature. Moreover, most of the algorithm parameters are fine-tuned using the recovery error, using the same evaluation metric should give the most accurate comparison.

2.2 Related Literature

In this section, a general overview of the literature is presented. This section mostly concentrates on the methods focusing on estimating the effect of the light source and appropriate to be a part of a digital camera pipeline. The approaches in the literature are explained briefly without detailed explanations, while benchmark algorithms and algorithms that are related to our study are analyzed and explained in more detail.

Illuminant estimation methods can be divided into three main categories: Statistics-based approaches (Section 2.2.1), Learning-based Methods (Section 2.2.2) and Methods Utilizing Semantic Information (Section 2.2.3).

In fact, the illuminant estimation methods are often categorized under Statistics-based and Learning-based categories as in [37] [33]. In order to emphasize the connection between our study and the past work, we also preferred to add a third category, as

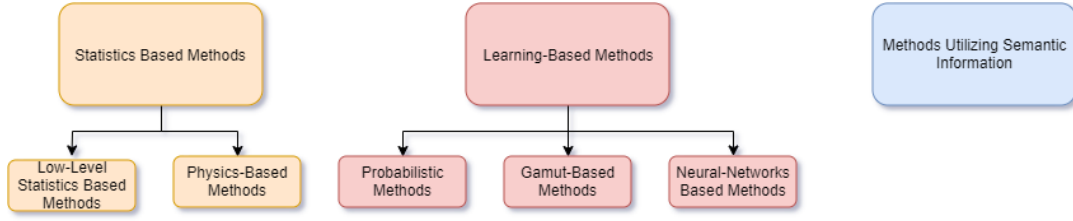


Figure 2.4: Taxonomy of color constancy methods

methods utilizing semantic information.

It is important to note that these categorizations are not divided by strict and disjoint boundaries. For example, to some extent, every method uses underlying physical principles of image formation, not only Physics-based methods. Furthermore, some of the approaches under different subcategories might be strongly related. Hence, there cannot be a single exact taxonomy for the categorization process.

In this section, in addition to the two general categories which are discussed under Section 2.2.1 and Section 2.2.2, we discussed approaches that somehow benefit from the semantic information in the image under the category: Methods Utilizing Semantic Information Section 2.2.3. The underlying purpose of this categorization is to have a better understanding of how semantic information of the image is utilized by various state of art studies.

2.2.1 Statistics-based Methods

In this part of the thesis, statistics-based methods are examined without much detail. Approaches that are examined under this section are built on the assumptions relying on either statistical distribution of the colors in the image or assumptions based on physics underlying image formation. They do not require any training data and their parameters are fixed before application. Usually, these parameters are set on their performance on images with a ground-truth. The parameters and algorithm's success depends on the test images.

Statistics-based methods can be categorized based on their characteristics: *Low-level Statistics-based Methods* and *Physics-based Methods*. The approaches based on any

kind of estimation of the distribution of colors in the images are examined under Low-level Statistics-based category. For example, Gray-World [13] assumes that the average surface reflectance should be equal for all colors under white light and assumes that any color distribution that does not fit this assumption is the result of a color shift of the light source. These approaches are analyzed under Section 2.2.1.1.

By contrast some approaches focus on exploiting the physical properties of objects and based on a more general image formation. They are based on the dichromatic model [40] and rely on assumptions based on the underlying physics of the image. These approaches are examined under Physics-based Methods (Section 2.2.1.2).

2.2.1.1 Low-level Statistics-based Methods

These methods are often built on statistical assumptions based on the distribution of colors of the image pixels. A simple approach would be Gray-World method [13] which is based on the assumption that the average surface reflectance of the image is equal for all color channels. This assumption implies that any inequality of the average color of the image is the direct result of the spectral distribution of the light source. This method is shown at Equation 2.9 where $\mathbf{I} = (\mathbf{I}_R, \mathbf{I}_G, \mathbf{I}_B)$ is the illuminant source, $(f(x) = [f_R(x), f_G(x), f_B(x)]^T)$ is the pixel value array at location x and N is the number of pixels presented in the image.

$$\mathbf{I} = \left[\sum_i^N f_R(x), \sum_i^N f_G(x), \sum_i^N f_B(x) \right] \quad (2.9)$$

A variant of Gray World method uses a fixed reflection ratio [4], instead of assuming the reflection is equal for all channels. In this case, the estimated illuminant is obtained by multiplying the mean of the image channels and fixed coefficients. This method is called as Database Grey-World and its formulation is shown in Equation 2.10 where $[\alpha, \beta, \gamma]$ is the coefficient triplet. These parameters can be set accordingly to the problem.

$$\mathbf{I} = \left[\alpha \sum_i^N f_R(x), \beta \sum_i^N f_G(x), \gamma \sum_i^N f_B(x) \right] \quad (2.10)$$

Another approach based on a similar assumption is White Patch [43] and it is built on the assumption that in the scene the light is reflected perfectly from at least one surface without any loss. Since there is no loss, the corresponding pixel is expected to be the brightest one and to reflect the color distribution of the light source directly. This method is often implemented as MaxRGB algorithm [43] which assumes instead of one pixel with perfect reflection there might be three different perfect reflections for every color separately. MaxRGB method is shown in Equation 2.11

$$\mathbf{I} = \left[\max_x f_R(x), \max_x f_G(x), \max_x f_B(x) \right] \quad (2.11)$$

Gray World and White Patch assumptions are expected to fail for certain cases in real life. For example, a landscape image containing sky and grass, Gray World poorly estimates the illumination trying to make sky gray. The Figure 2.5 shows two examples of such a situation, images taken from REC and gamma correction is applied for visualization purposes.

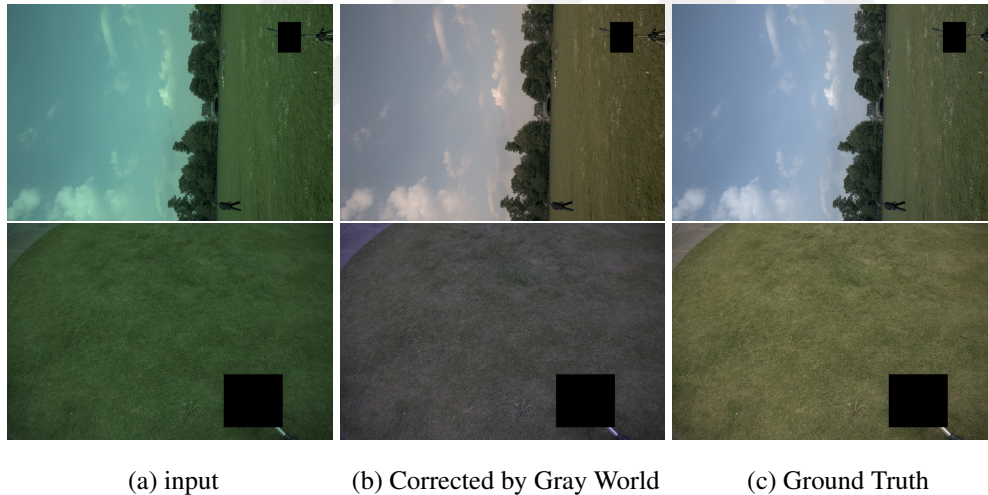


Figure 2.5: Example scenes where Gray World algorithm fails. Images are taken from REC [35].

In practice, these two general methods are implemented with an additional process. For example, Database gray-world uses a fixed reflectance ratio for colors and adapts the ratios according to the problem. Alternatively, [4] segments image into patches and instead of the average color of the image, it uses the average color of the patches.

This process can decrease the effect of uniformly colored surfaces' impact. Another example is to smooth image before applying White Patch, resulting in a decreased effect of noise caused by the imaging device. Clipping over-saturated pixels and using a simple filter to smooth the image are common pre-processing applications for statistics-based approaches.

Another method using low-level statistics is Grey Edge [58]. Their method is based on the assumption of under a white light source average color of the edges in the image should be gray. Therefore any shift in the average edge color is caused by the light source. Gray Edge performs closer to the state-of-art approach and is still used as a benchmark algorithm.

These algorithms can be unified in a more general expression. Weijer et al. [58] unify low-level statistics-based methods under the Equation 2.12 where $(\mathbf{I} = [I_R, I_G, I_B]^T)$ is the estimated illuminant color, n is the order of the derivative, $f_\sigma(x, y) = f(x, y) \otimes G_{\sigma}(x, y)$ is the convolution of the image with a Gaussian filter $G_\sigma(x, y)$ with standard deviation σ , p is the degree of Minkowski norm and (x, y) is the spatial location of the pixel.

$$\mathbf{I}(n, p, \sigma) = \left(\int_x \int_y |\nabla^n f_\sigma(x, y)|^p dx dy \right)^{\frac{1}{p}} \quad (2.12)$$

Parameters of the Equation can be decided depending on the problem, imaging device, or the dataset, resulting in improved performance. For example, [24] compared the effect of the Minkowski norm p and found out that setting $p = 6$ ends up with the best performance for a set of labeled images. It is important to note that the optimal set of parameters may depend on the selected performance measurement method and the test images used during the experiment. They cannot be considered as global settings. In Table 2.1 various algorithms are instantiated from Equation 2.12.

If (n, p, σ) is set to $(0, 0, 0)$, Equation 2.12 becomes Gray World, calculating the average color of the pixels. White Patch also can be obtained if (n, p, σ) is set to $(0, \infty, 0)$. Gray Edge method estimates the illuminant source as the average color of the edges. When $(n, p, \sigma) = (0, 1, 0)$ Gray Edge algorithm is generated.

Low-level Statistics-based Methods are based on simple assumptions. However, they

Table 2.1: Instantiations of Equation 2.12 with different settings of (n, p, σ)

Algorithm	n	p	σ
Gray World	0	0	0
White Patch	0	∞	0
1st order Gray Edge	1	0	0
2nd order Gray Edge	2	0	0
Shades of Gray	0	6	0

often show a considerable performance in real cases. Combining with their computational simplicity, they are still valid approaches and preserve their applicability either as an alone estimation method or within other methods. Moreover, the idea behind these methods are still exploited by various state of the art algorithms such as [7] and [51].

2.2.1.2 Physics-based Methods

Physics-based Methods are based on the physical interaction between the light and the surface. They use a more general image formation model, the dichromatic model [40] which is given in Equation 2.1. A simple approach would be to assume at a point in the image, body reflection is zero ($m_b = 0$) and the surface has purely specular reflectance. In practice, specular reflections are typically brighter than the body reflectance. As a result, if there is reflection consisting only of specular reflection, it is expected to correspond to the brightest pixel of the image. This method is in fact the White Patch algorithm [43]. Unfortunately, this assumption often fails due to two main reasons. The first reason is that fully specular reflections rarely appear in real images. The second reason is that if such a reflection exists, it is likely to be clipped or over-saturated.

More realistic approaches have been proposed as well. A common one is that pixels corresponding to the same surface falls on a plane in RGB color space. This plane is defined by the surface reflection $S(\lambda)$ and spectral distribution of the illuminant source $I(\lambda)$. There is expected to be more than one surface in an image resulting in

multiple planes in RGB color space. [44] and [56] built their methods on the idea that the intersection of these planes must be determined by the illuminant source. Unfortunately, in practice, it did not perform well due to the challenge of reliable segmentation of unique surfaces.

Funt et al. [57] proposed a method that does not require the identification of the surfaces. Instead, they described the colors in an image using a multi-linear model which consists of multiple planes sharing an intersection which is directly the result of the illuminant source. Funt et al. [52] later proposed a method built on the same idea but instead based on two Hough transform voting procedures.

2.2.2 Learning-based Methods

In this section, the algorithms that require training data are discussed. These approaches benefit from a set of images that are either taken under a white light source or comes with the information about spectral power distribution of the light source.

It should be emphasized that learning-based methods strongly depend on the training data. Any mislabeling of the illuminant source decreases the performance of the approaches. A major problem with these methods is the limitation of the number of labeled-image, which results in challenging situations during the training process. Moreover, the trained model is strongly dependent on the camera model on which the images are taken and the variety of the scene types.

2.2.2.1 Probabilistic Methods

In this section, algorithms that use some kind of model are discussed briefly. They generally form a prior distribution over occurrences of colors under possible illuminations.

One of the most simple implementations of this approach is so-called Color By Correlation [22] which can be thought of a summary of its kind. It is based on the probability of a possible scene light for the image, based on the chromaticity distribution of the image. At first, prior information about the possible light sources is encoded

by learning the distribution of the chromaticities for each possible light source; then, if an input image is given, its chromaticity distribution is computed. Finally, the likelihood of the chromaticity distribution for the possible light sources is calculated, followed by the selection of the most likely possible light source.

2.2.2.2 Gamut-based Methods

Gamut mapping algorithm is first proposed by Forsyth [27]. The method is based on the assumption that in the real world, under a selected illuminant source, only a limited number of colors can be observed. Therefore, any color variation which is not in the set of colors observed under the selected illuminant source is the result of a deviation of the color of the light source. The limited set of colors are called canonical gamut C for the selected light source. However, in most cases, the color of the selected light source is white.

The canonical gamut is found by a training phase using the images under the canonical light source. Since the canonical gamut in some way describes the expected colors under the given light, the reliability of the model relies on the number of images and the diversity of environment images are taken or even the camera model images are taken [11].

After the training phase, for any given input image under an unknown light source, its gamut is initially calculated. The objective here is to find a unique mapping that transforms the input gamut so it completely lies within the pre-computed canonical gamut. However, in practice under the assumption of the diagonal model (Section 2.1.2.2) several possible mappings exist. Every individual mapping M_i of the set of feasible mappings M transforms the input gamut I so that it completely lies inside of the canonical gamut (Equation 2.13).

$$M_i I \in C \quad (2.13)$$

Following the calculation of the feasible set of candidate illuminations, the final estimation is decided by using the feasible set. For example, [27] selected the mapping

resulting in the most colorful scene. On the other hand, [2] proposed a simpler alternative, average or weighted average of the feasible set M .

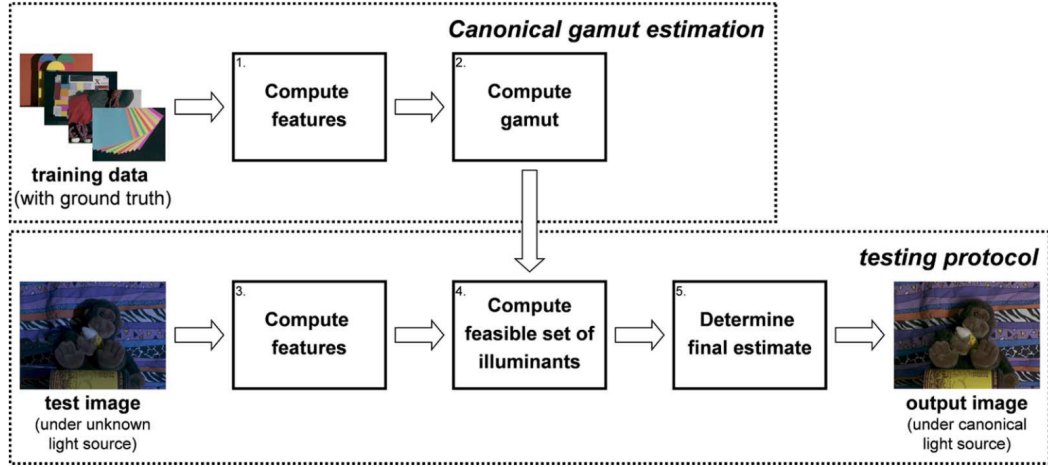


Figure 2.6: Application of gamut mapping. First, a canonical gamut is computed using the training data. Then a set of feasible mappings are calculated. Lastly, one of these mappings is chosen and used to construct the correctly white-balanced image. The image is taken from colorconstancy.com: How to achieve?

This simple but novel approach is extended and improved by several other works. An extension of the gamut mapping which addresses the difficulty of the implementation process by [20] and [19]. They computed the gamuts using chromaticities ($r = \frac{R}{B}, g = \frac{G}{B}$). However, this 2 dimensional (2-D) approach resulted in a slightly worse performance than the 3 dimensional (3-D) approach. This decrease is the result of the distortion of the feasible set caused by the mapping of the images from 3-D to 2-D chromaticity values. Later [19] and [18] overcame the problem by mapping the feasible map from 2-D to 3-D before the selection process. Another simplified version of gamut mapping which uses a cubic representation of the canonical gamut instead of a convex hull is proposed [47].

Some other extensions of the method focus on the dependence of the diagonal model. In reality, the diagonal model might not be fitting to the image perfectly. Therefore, the original method might result in a null feasible set of mappings where there is not any single transformation for the input gamut to map it into the canonical gamut. One intuitive approach used to solve this problem is to augment the input gamut until some

feasible mappings show up [3]. Another solution addressing the problem is to expand the canonical gamut [27] [2] until a set of possible solutions can be found.

Another method proposed by Finlayson [23] is called Gamut Constrained Illuminant Estimation. Instead of estimating the color of the illuminant source, this approach considers only several possible illuminant sources, and based on these possible sets of illuminant sources it classifies the image. For every possible illuminant source, a canonical gamut is learned. Then the unknown light source of the image is estimated by matching the input image gamut to each of the canonical gamuts. The best matching canonical gamut is used as the final estimation.

2.2.2.3 Methods using Neural Networks

The methods discussed in this part are mostly built on neural networks. The underlying idea is that a network can learn the relationship between an image and its illuminant source. In other words, neural networks can be seen as a representation of the statistical structure of the relation between the image and the illuminant effect.

The performance of the network heavily depends on the training data. Usually, the training process results in a neural network that can estimate the illuminant source when it is fed with the input image. Thus, the performance of this estimation is based on the number of training images, training process of the model, and the diversity of the training images. The training process requires an image under a known light source which is used as the ground truth. Datasets dedicated to color constancy usually includes an object which is later used to calculate the color of the illuminant. For example, Recommended Colorchecker Dataset (REC) [35] consists of images taken with a Macbeth Chart inside. Patches of the chart are later used to calculate the ground truth for the image.

Cardei et al. [15] used image chromaticity $c_1 = \frac{R}{R+G+B}$ and $c_2 = \frac{G}{R+G+B}$ information to estimate the chromaticity of the illuminant source $\hat{c}_w$. Given a trained neural network ($network(\cdot)$), illuminant estimation process can be shown as in Equation (2.14) where N is the number of pixels in the image. [28] also uses neural networks

to extract features but follows it with a support vector regression.

$$\hat{c}_w = \text{network}(\{c_i | i = 1, 2, \dots, N\}) \quad (2.14)$$

Recently, deep learning techniques allowed an increase in performance for several computer vision problems. This success might be due to the increased number of training data and the increased performance of the computational tools. Color constancy methods shared the same destiny with other common vision problems and a great majority of recently introduced approaches are based on the neural networks or somehow benefit from them.

Bianco et al. [9] trained a neural networks involving convolutional layers to predict the illuminant color of a small square patch. They segmented the image into several patches with a pre-determined size. These patches are used instead of the image both during training and testing. Every patch is associated with illuminant label of the image it is coming from. This splitting process allowed them to obtain an increased number of training data which is an important requirement for training process.

Their method [9] contained two stages: Local illuminant estimations and combining the local estimations into a global illuminant estimation. They implemented the local estimator with several different hyper-parameters, resulting in a selection of a relatively shallow network. The selected local estimator is shown in Figure 2.7.

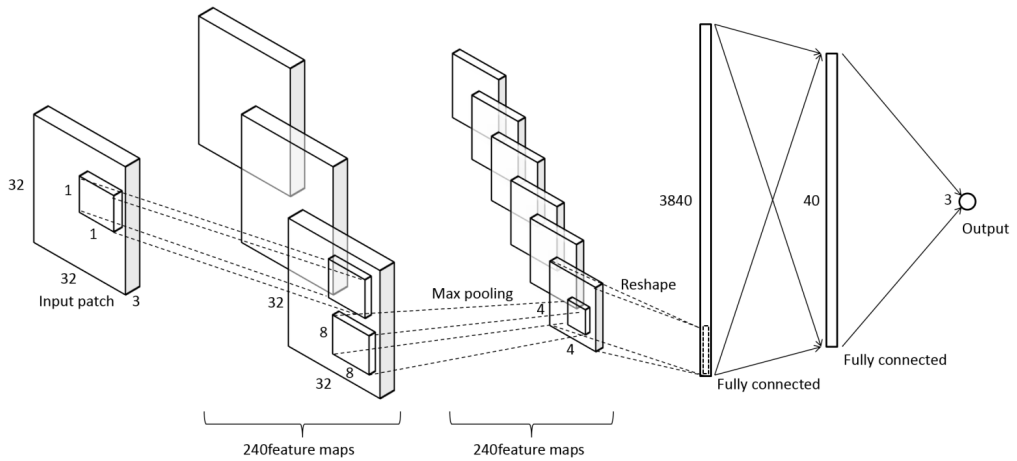


Figure 2.7: Graphical representation of local estimators used in [9] and [10]

Later, the same authors [10] improved their method by introducing a local to global estimator. Figure 2.8 is taken from their study showing the schematic expression of their method. At first, they separated the image into local patches. For every local patch, an illuminant estimation is computed using the pre-trained neural network. Secondly, they used the 2-D kernel density estimation (KDE) to decide the number of illuminant sources in the image. The number of the light sources is found by the number of the peaks of KDE. If there are multiple light sources, local patch estimations are directly used to correct the image. If only a single illuminant source is found, then instead of taking the mean or median of the local estimators as they did in [9], they trained a local to global regressor, which is used to predict the color of the illuminant source.

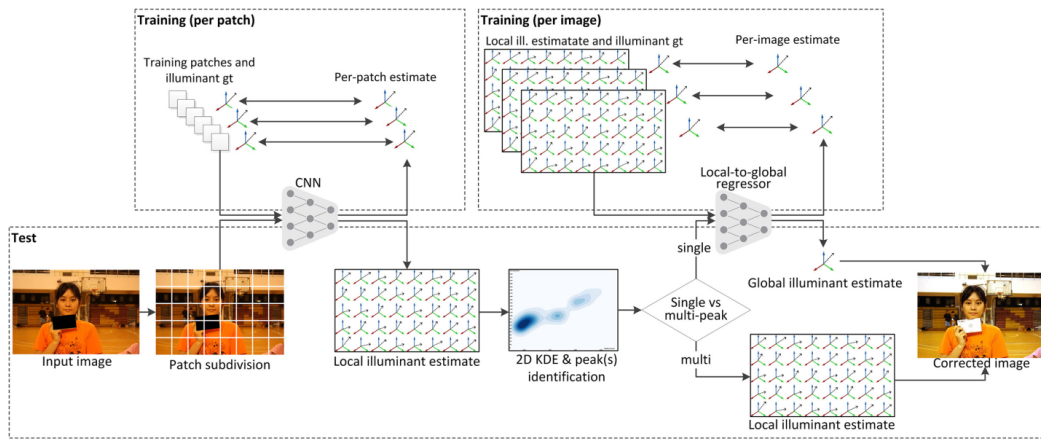


Figure 2.8: Schematic Expression of the proposed algorithm by [10]

A deep architecture is proposed by Shi et al. [54]. They proposed a new convolutional network architecture with two sub-networks. One of them is called Hypotheses Network (HypNet). This sub-network is based on the idea of local regions of an image being able to be modeled differently for different scenarios. For example, some of the regions might be affected by shadow and have textured areas while some of them are smooth and non-shadowed. Instead of making one estimation for every local patch, they made two estimations that both specialized in different situations. They extracted spectral and spatial information via convolutional filters. Then, a regressor is trained to estimate two distinct estimations. The second sub-network is as they call it, Selection Network (SelNet). This network is used to select the appropriate estimation for

every local region. Lastly, a simple median pooling is performed on the selected local illuminant estimates of the image.

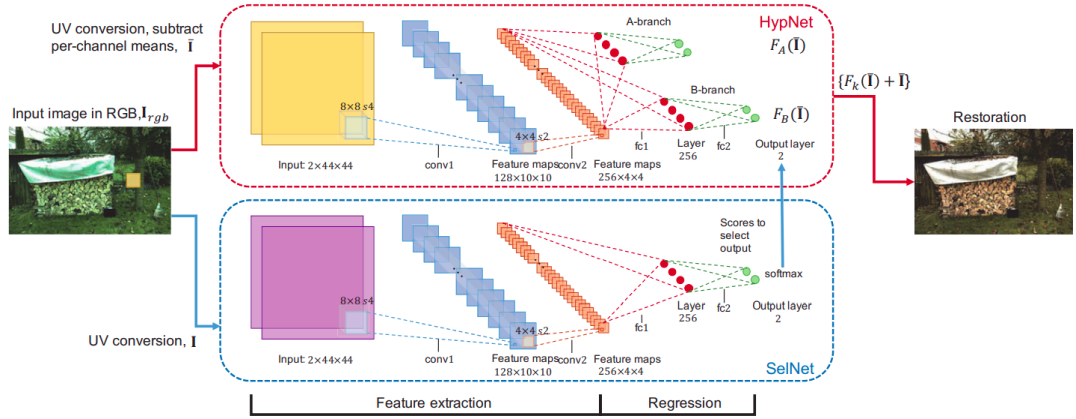


Figure 2.9: Network architecture of Deep Specialized Network for Illuminant Estimation. HypNet generates two distinct illuminant estimations for all local patches. Then, SelNet decides which estimation will be used. Lastly a simple median pooling is used to find the global illuminant estimation [54].

The network architecture of [54] is shown on Figure 2.9. The diversity encouraging multiple illuminant estimations which are specialized in various situations achieved state of art performance. The downside of the approach is that it is computationally costly compared to the other state of art approaches. On the other hand, the idea of their multiple estimations by training distinct regressors is innovative and might be beneficial for other models as well.

Sidorov [55] proposed a novel Generative Adversarial Network (GAN) specially designed for the color constancy problem. The proposed model performed an average performance on single illuminant estimations while outperforming the multi-illuminant estimation methods. They explained that their model treats both tasks equally. However, single illuminant estimation methods oversimplify the problem and work well only within the defined problem. Also, another study [17] discussed the usage of GAN in color constancy problem, possible errors, and recommendations for further usage.

Another novel approach is Bag of Color Features (BoCF) [42]. This method is inspired by the assumption that only the second-order gradient information of the image

is sufficient to extract the illuminant information [16]. The proposed method consists of 3 blocks: Feature extraction block, bag of features block, and estimation block. Later two self-attention mechanisms are included between the blocks. Proposed architecture for image correction is shown in Figure 2.10

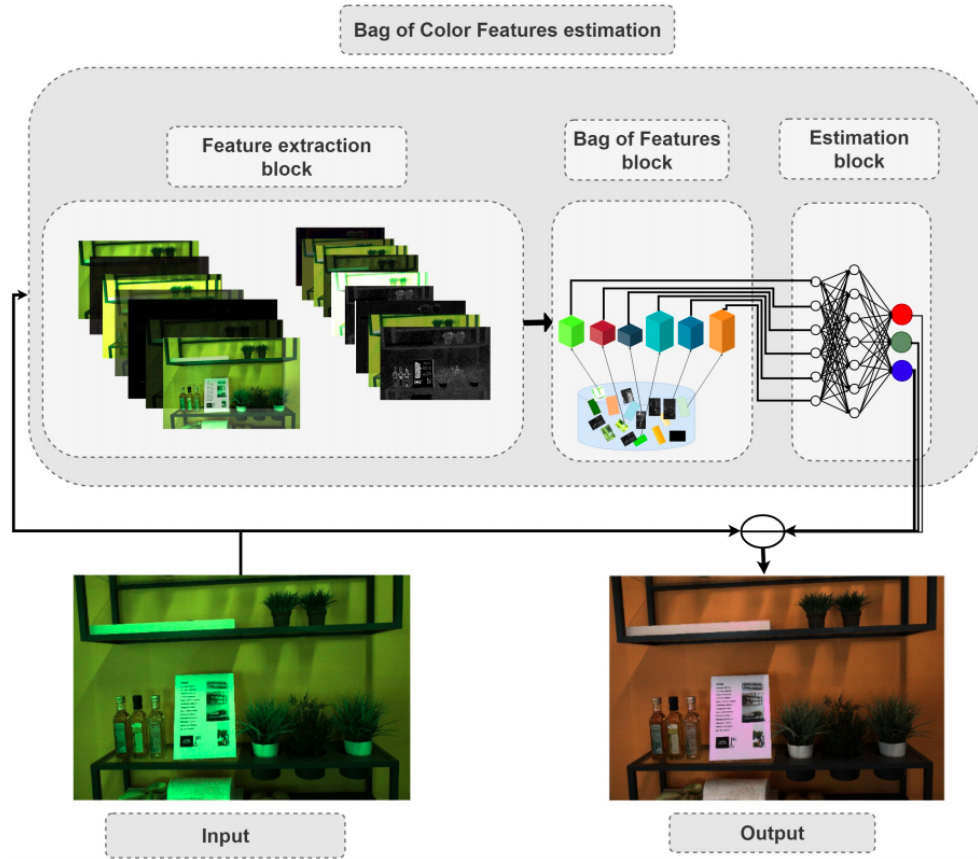


Figure 2.10: Illuminant estimation process of Bag of Color Features approach [42]. Firstly features are extracted. Then the extracted features are mapped into a histogram and lastly the histogram is used to illuminant estimation

Bag of Color Features can be described as a learning-based generalization of many statistical methods and fills the gap between the statistical methods and methods based on neural networks. They introduced the bag of features pooling [48] into the color constancy problem. They deliberately ignored the global spatial information by using a shallow feature extraction block and focused on only the local information.

Bag of Colors has an important advantage over methods with deeper networks. By using only 2 convolutional layers, they reduced the number of parameters substantially.

In addition to that, despite their low computational cost they achieved competitive results with the state of the art approaches.

Another innovative approach for color constancy is Learning Illuminant Estimation from Object Recognition by Buzzelli et al. [14]. They trained an illuminant estimation network without any labels but with the objective of optimizing the predictions of a specifically trained object recognition module. During the training process, back-propagation from a parallel object recognition network is used. They specifically pre-trained the object recognition network to make it overly-sensitive to color variations. For the purpose of the problem a vegetable recognition task is selected.

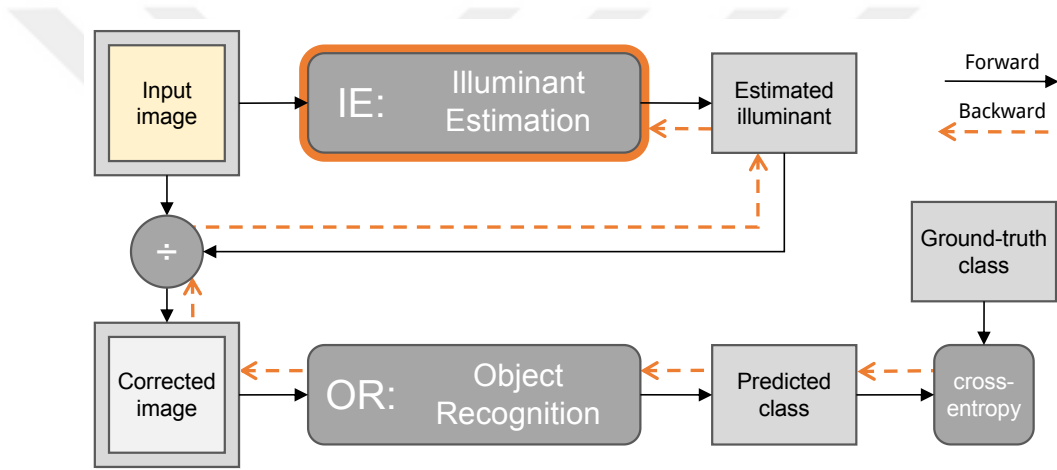


Figure 2.11: Schematic representation of the training process of [14]. At inference time only the illuminant estimation module is used.

In order to make the object recognition module over-sensitive to color as a useful auxiliary for illuminant estimation, [14] followed a two-step strategy during the training phase (Figure 2.11). First, the object recognition network is trained alone without any color change in input images. Secondly, during the training process of the illuminant estimation network, the color of the input images is tinted and weights of the object recognition network are frozen. On the other hand, this time weights of the illuminant estimation network is updated. During the testing phase, only the illuminant estimation network is used.

[14] is a novel and interesting approach both because it used the object recognition task as an auxiliary during the training process and it did not require any label of the illuminant source.

2.2.3 Methods Utilizing Semantic Information

In this section, various methods that somehow benefit from the semantic information are discussed. Although, to some extent, other methods might be utilizing semantic information as well. For example, Shi et al. [54] roughly learns two different sets of parameters for two different situations. Then, with the help of a selective network trained for selecting the better estimation of the two, for every local patch they make a decision for the appropriate estimation, using the information extracted from the patch.

Firstly, approaches that utilizes the semantic information to select the appropriate algorithm [8] [32] or benefit from the extracted semantic information to weight associated illuminant estimations from local patches of the image [39] are discussed. There are also some approaches exploiting the object information explicitly. An example is [50] which uses object classes to improve the performance of common color constancy algorithms.

Another approach [31] uses image regions. They used a simple segmentation method in order to distinguish the reliable image parts of the image for color constancy for a given scene category. The selected categories were open country, street, and indoor. A visualization of the proposed approach is shown in Figure 2.12.

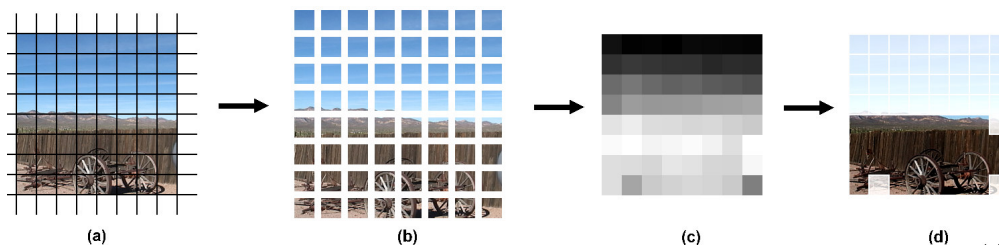


Figure 2.12: Segmentation and evaluation of image regions [31]

[31] is an experiment on the idea that different regions of an image might differ in terms of the provided information for illuminant estimation. If invaluable regions of the image can be located and taken out from the estimation process, the overall performance of a color constancy method can be enhanced. This study shows that depending on the image content, some regions of the image might not be a good information source for the illuminant estimation for the selected color constancy algorithm. They later [32] deepen their proposed method, selection of the best algorithm depending on the image statistics.

Gijssen et al. [32] later proposed a classifier that selects the best performing color constancy algorithm for a specific image. They showed that images from the same scene (i.e. landscape) share similar parameters. Then, using a categorization based on these parameters, evaluated the performance of the color constancy algorithms on different types of images. In the end, they proposed an algorithm selection depending on the scene properties. For instance, if the input image is rich in terms of edges, a higher-order statistics-based method is selected.

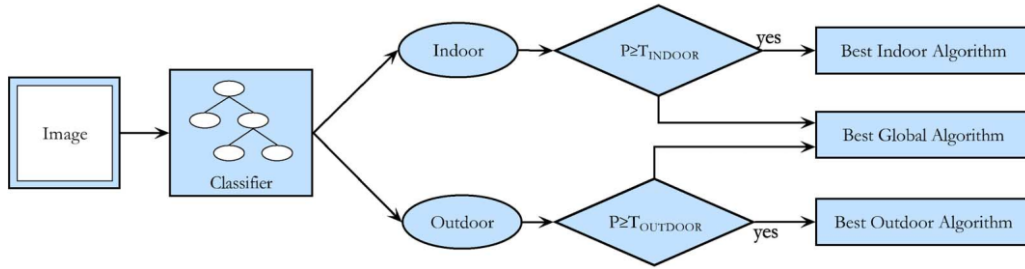


Figure 2.13: Working scheme of [8]

Another method based on the scene classification is proposed by Bianco et al. [8]. They designed different strategies to decide the appropriate estimation method based on the image category. They classified an image using 3 categories: indoor, outdoor and uncertain. Following the classification, algorithms chosen for the category is applied to the input image. If the image category is uncertain then a selected class-independent algorithm is chosen. Their results demonstrated that a classification strategy outperforms an individual strategy on an image set.

Color constancy methods differ in estimation performance on different categories of images. For example, most of the statistical color constancy algorithms under-

perform on landscape images containing a high number of pixel count for the sky. Some basic solutions to this problem are discarding the problematic regions [31] based on the context or if using local patches, reduce their participation in the final illuminant estimation [39]. The categorization processes allow choosing the best performing method on the categorized images thus an overall increase in the performance.

Methods discussed until here took advantage of image categorization, selection of an appropriate color constancy algorithm depending on the image, or some kind of a fusion of several methods. After this point, algorithms that explicitly use a color association for an object class on purpose are discussed. For instance, [1] shows that training a neural network with a given segmentation mask shows improved performance over training the network with only the input image. These approaches tend to achieve a low angular error.

There are not a lot of studies on explicit use of an object category for color constancy problem. With the recent developments in available datasets and computational resources, the performance of detection and segmentation of an object is increased substantially.

Unlike most approaches [58] benefits directly associated information of 9 particular classes: grass, tree, cow, sheep, sky, water, face, road, and building. They proposed segmenting an image into patches described by three modalities: texture, color, and position. The texture is described by SIFT descriptor [46], color is Gaussian averaged color of the patch and the position is the location of the patch in the image. These visual words are then used to compute the semantic likelihood of the image.

Their algorithm combined of two types of approaches: *top-down* approach and *bottom-up* approach. Top-down approach produces candidate corrected images based on the detected object classes. For example, if sky and grass are found in the image, it produces a corrected image based on the sky patches and another corrected image based on the grass patches. On the other hand, bottom-up approach computes images based on six instances of Equation 2.12 with pre-determined parameters. At last, they computed the semantic likelihood for every candidate illuminant correction, and correction with the highest likelihood is selected. A visualization of the process is shown

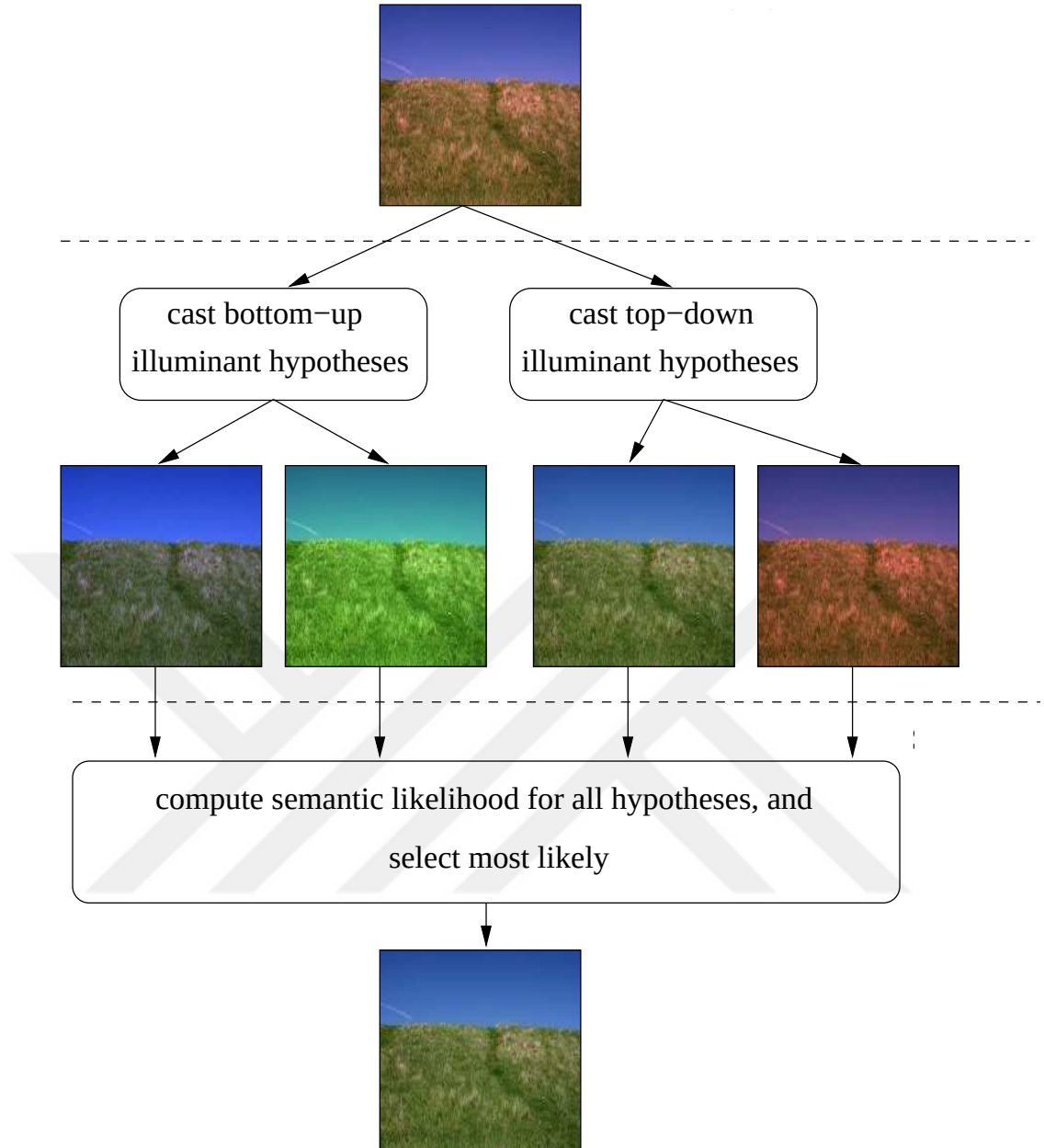


Figure 2.14: Visual representation of the proposed method by Weijer et al. [58]. Possible illuminant estimations are proposed by top-down and bottom-up approaches. Top-down proposes based on the segmented classes while bottom-up uses different instantiations of Equation 2.12. Lastly the most likely estimation is selected.

in Figure 2.14

An important aspect of the study [58] is that they introduced the *Green-Grass Hypothesis*: The average reflectance of a semantic class is equal to the average reflectance of

its instances in the dataset. They also showed that with an accurate classification process illuminant estimation performance can be enhanced. This hypothesis also holds an important role in our study. We defined an expected color for every one of the 133 classes from COCO dataset [45] instead of 9 classes. More detailed explanation is given in Section 3.2.2.

Another method based on the idea of the color of an object class is proposed by [50]. They proposed an image processing pipeline that benefits from recognized object categories. Their method is based on the idea of that some object classes can be associated with a relatively constant color (e.g. the sky).

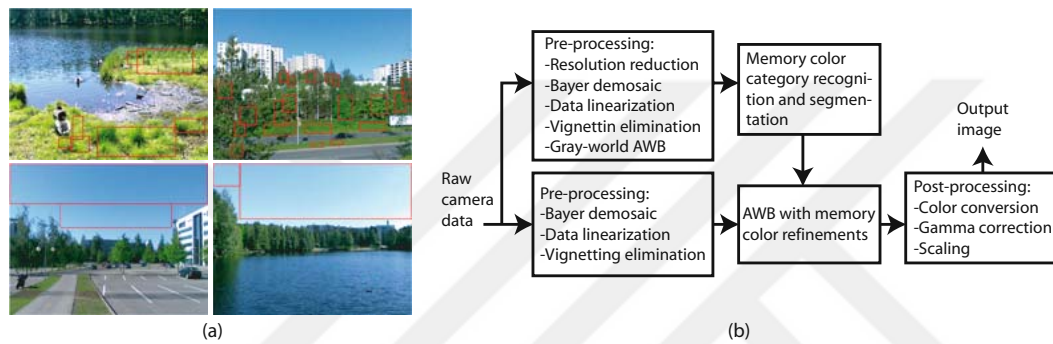


Figure 2.15: Image processing pipeline proposed by [50]. Recognized object categories are used to refine the illuminant estimation

Figure 2.15 shows the proposed image processing pipeline by [50]. First, the image is transformed into a color space which is in most cases linear RGB. Following the transformation, the image is pre-corrected by a simple but fast Gray World [13] algorithm. Gray World both helps the object recognition process and decreases the dependency of the method to the camera sensor properties. Lastly a fine-tuning of the estimation is done by using the recognized object classes and their associated colors.

Afifi [1] proposed a semantic white balance method. He studied the effect of feeding a semantic mask along with the input image into a convolutional neural network. His implementation is based on AlexNet with an adaptation allowing it to take 4 Dimensional (4-D) inputs. The detailed structure of the model is given in Figure 2.16

Afifi [1] compared the proposed method against a regular AlexNet, only feeding the 3-D input image. The results indicated a considerable improvement when the network

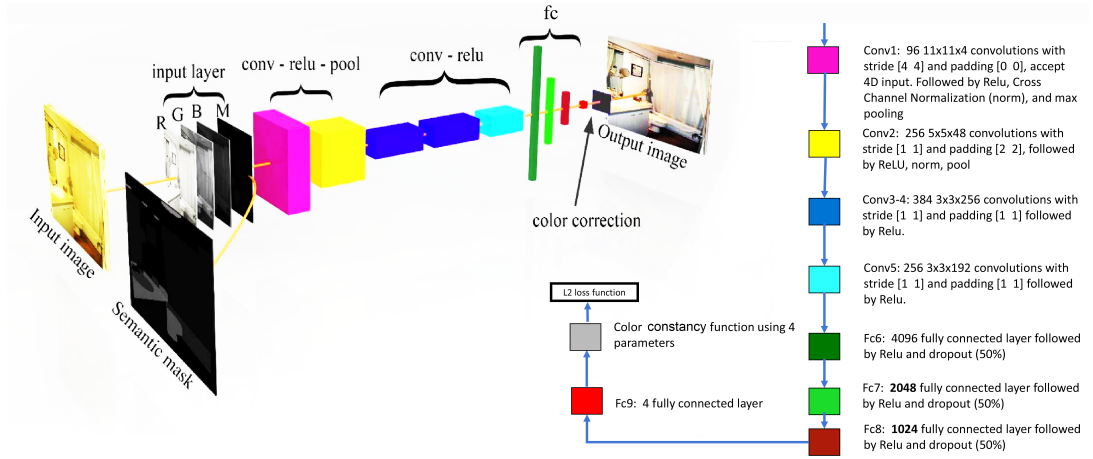


Figure 2.16: Network structure of [1]. Segmentation mask is fed as the fourth channel network along with the input image

utilizes the semantic mask as well. This result shows that specific information of the corresponding object class for a pixel enhances the performance of illuminant estimation.

Bianco et al. [11] designed a color constancy algorithm exploiting the skin regions in the image. Their method is based on detecting faces, extracting pixels, and lastly using the segmented face pixels in order to find a mapping to the pre-computed human skin gamut.

The proposed method is based on the assumption that skin colors form a compact cluster in the color space. This cluster can be used to map skin pixels to make an illuminant estimation. First, they tested the compact cluster hypothesis. They combined the color of the face pixels from different faces individually for six different commercial cameras (Figure 2.17) in the *Hue* and *Saturation* plane. The result showed that skin colors tend to form similar clusters in color space. In other words, it shows that skin gamut might be used as a clue for color constancy applications.

Color Constancy Using Faces [11] consists of three parts. First, a face detection method is used to extract the faces in the image. Following the face extraction, an adaptive thresholding (Figure 2.18) is applied to the pixels in the face boxes. Only the pixels closer to the peak of the Hue histogram of the face pixels are selected.

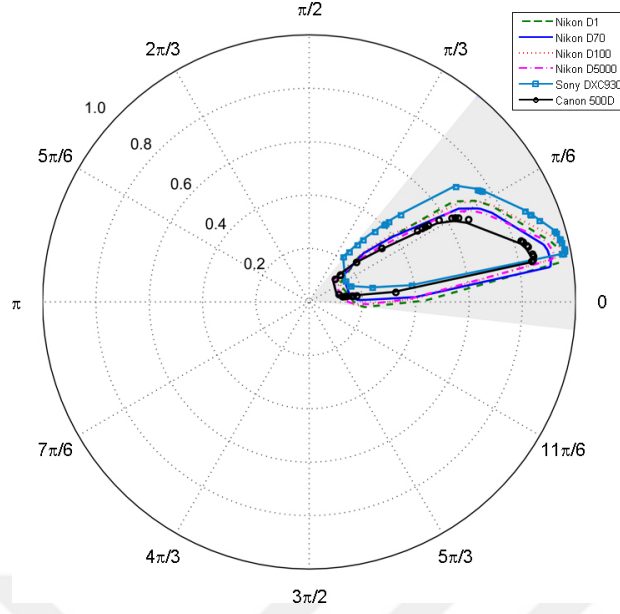


Figure 2.17: Convex hull of face skin pixels in HS plane of HSV color space. The color of skins form a compact cluster which can be used as a clue in illuminant estimation [11]

The remaining pixels are discarded. The first thresholding process allows discarding any outlier pixel. Then second thresholding which is based on the Value (V) of HSV pixels is applied to discard too bright or too dark pixels. If the number of remaining pixels is higher than a predefined threshold, they are added to a pool of pixels. The process repeats itself until there is no face left in the image. In the end, if the pool is empty, a global illuminant estimation is made. Otherwise, the illuminant is estimated based on the computed feasible mappings.

The proposed approach later improved [12] so that it automatically switches between a global illuminant estimation and a local illuminant estimation. Unlike [11] they made estimations for each face independently. If the maximum distance between the estimations is above a predefined threshold, then algorithm considers image as non-uniform in terms of the illuminant source effect and make a spatially varying illuminant estimation. However, if the maximum distance between the estimations is below the predefined threshold, the global estimation is obtained by averaging the individual estimations.

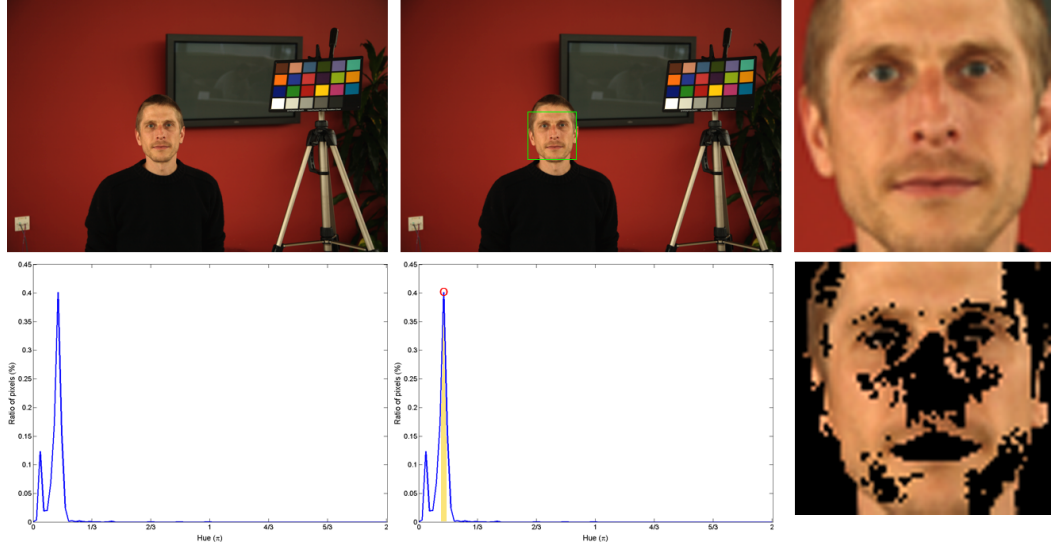


Figure 2.18: Adaptive thresholding of the skin pixels. The peak of hue is found and only pixels with relatively close pixels selected. [11]

The approaches discussed after this point partially utilizes the semantic information in the image. For example, [5] uses different transformations of the image, enhancing the semantic information. [39] makes local estimations throughout the image and for every local patch also produces a confidence metric.

Barron, in his study Convolutional Color Constancy (CCC) [5], reformulated the color constancy problem into a 2-D spatial localization problem in the histogram of the image in log chrominance uv space (Equation 3.1). Proposed log chrominance space transforms multiplicative constraint relating properly white balanced image W and the input image I (Equation 2.5) to an additive constraint. The rewritten problem formulation in log-chrominance space is given in Equation 2.15 where $W_{uv} = [W_u, W_v]^T$ is the properly white balanced image, $L_{uv} = [L_u, L_v]^T$ is the illumination effect and the $I_{uv} = [I_u, I_v]^T$ is the input image in log chrominance space.

$$I_{uv} = W_{uv} + L_{uv} \quad (2.15)$$

The reformulation of the problem and selected color space allowed him to use object detection techniques for illuminant estimation. The preferred color space later inspired us and [54] as well. The underlying concept of this paper is that the histogram

of the log-chrominance space of an image is only shifted under different light sources, preserving its shape. A visualization of the concept is shown in Figure 2.19.

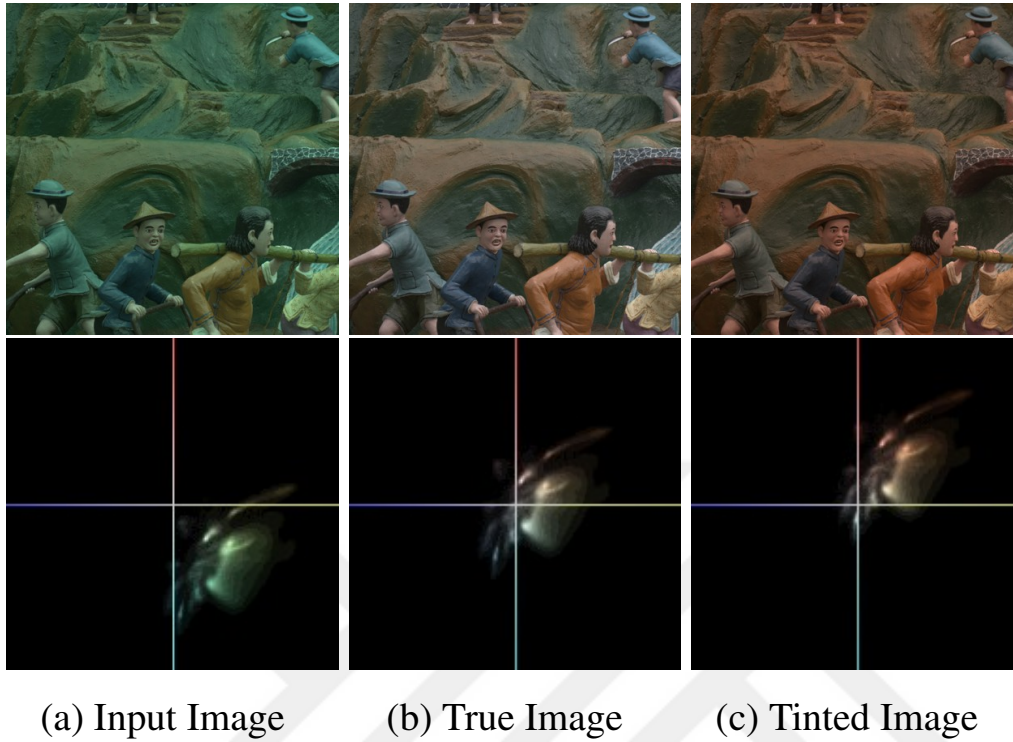


Figure 2.19: Slightly tinted images and their respective histograms in uv log-chrominance space. [5]

Barron [5] used this concept to turn the color constancy problem into a localization task. Since the shifting directly caused by the illuminant source, if how much the image is shifted from its properly white balanced position can be found, the color of the light source can be calculated in log-chrominance space which can be mapped to the desired color space later. The advantages of the log-chrominance color space and properties of its histogram are deeply discussed in Section 3.2.1.

The spatial localization task is done by a learned filter which is later convolved with the input image histogram and the peak bin of the convolved histogram is selected as the estimated illuminant of the light source. Another important touch of the CCC is that during the training process they used a discriminative approach instead of a generative one. In other words, their network is trained to discriminate between a properly white balanced image and a poorly white-balanced image. They showed that

this approach resulted in a filter that increases the weights of blue colors resembling sky while decreases the weight of pale greens that resemble poorly white-balanced images. They also used a generative approach for the training process as well. This approach ended up resembling a gray-world filter where it weights colors similarly.

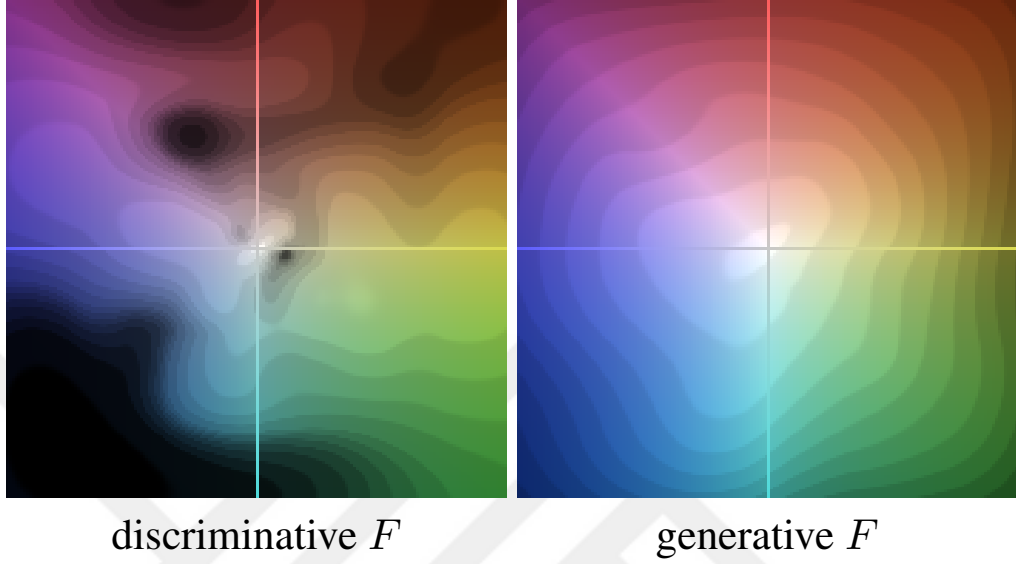


Figure 2.20: Visualization of filters. Both filters are trained on same training data but with different cost functions. Generative filter (right) resembles a gray-world approach where all colors are weighted similarly. Unlike the generative filter, discriminative filter (left) weights common colors presented in a properly white balanced image while subtracting unlikely colors. [5]

CCC also used a pyramid filtering technique [6] in order to avoid over-fitting and expensive evaluation process. Furthermore, they used augmented transformations of the input image. When only the input image itself is used, it is treated like a bag of pixels, discarding the semantic information. To put differently, all of the semantic information of the image is ignored because all operation is done in the histogram of the color space. The goal of the augmentations is to benefit from not only the pixel values but spatial statistics as well. Instead of classifying a single histogram, they trained a set of filters each specially trained for an augmented version of the image and its histogram.

In Figure 2.21 proposed inference process of CCC is shown. First, the given image I

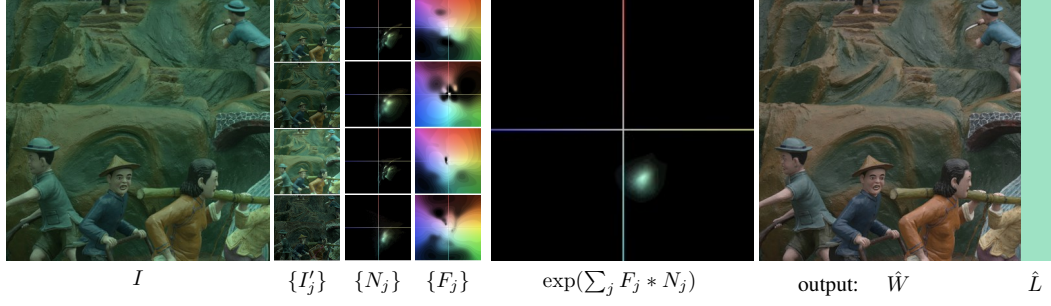
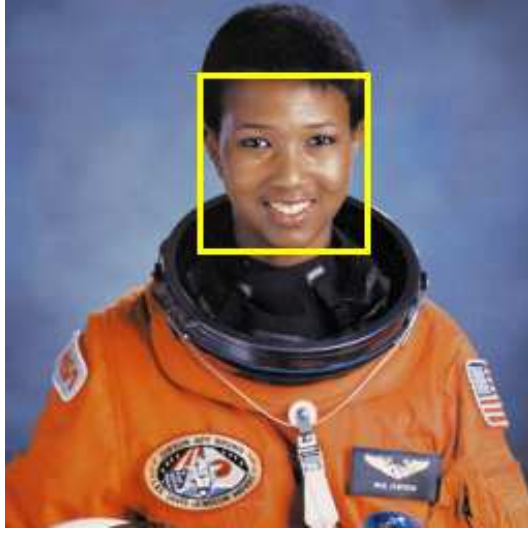


Figure 2.21: An overview of the proposed illuminant estimation process of CCC [5]. The input image I is first transformed into several version of itself I'_j . Then histogram representation N_j in uv log-chrominance space are convolved with the pre-trained filters F_j for each augmentation type respectively. Resulting convolutions are summed up and the histogram bin with highest score is selected as the illuminant estimation L .

is transformed into multiple augmented images I'_j . Then, the histogram of the images N_j in log-chrominance space is computed. All of the histograms are convolved with their respective trained filter F_j and summed up, giving a score for every bin of the histogram. Lastly, histogram bin with the highest score is selected as the color of the illuminant and corrected image W is calculated using Equation 2.15.

Barron and Tsai later presented Fast Fourier Color Constancy (FFCC) [7] which is based on a similar spatial localization task as CCC. FFCC introduces a localization task on a torus. The underlying idea is to repeatedly wrap this localization problem around a torus which allows faster estimation in exchange for new challenges. An analogy of the difference between two approaches [5] [7] in terms of a detection problem is shown in Figure 2.22.

The main difference between [5] and [7] is that FFCC introduces illuminant aliasing. This time the image histogram in uv log-chrominance space is wrapped around itself, meaning that a particular bin of the histogram no longer corresponds to a single (u, v) color but corresponds to an infinite number of (u, v) colors (Figure 2.23c). A localization task is done on this wrapped histogram. Following the localization, *de-aliasing* is applied to select the most likely illuminant estimate (u, v) among the corresponding infinite number of colors (Figure 2.23d). The selection is done by as they call



(a) Image A



(b) Aliased Image B

Figure 2.22: CCC is based on a 2-D localization problem (a). On the other hand FFCC [7] wraps this localization problem around a small torus (b) which later be used to detection. The final estimation is done by de-aliasing the wrapped problem.

gray light de-aliasing where the u, v couple closest to the center of the histogram is selected (Figure 2.23e).

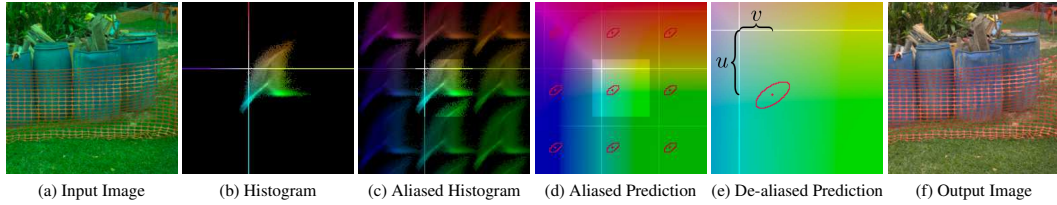


Figure 2.23: A demonstration of illuminant estimation using a wrapped histogram [7]

Hu et al. [39] introduced a Fully Convolutional Color Constancy with Confidence-weighted Pooling (FC^4). The underlying idea is learning to estimate the illuminant affecting the local patch alongside how reliable its estimate is. In other words, the network learns to distinguish semantically valuable and ambiguous regions. Therefore the network weights the estimations based on valuable regions (i.e. regions which are rich in terms of texture) while excludes the estimations of ambiguous regions (i.e.

solid-colored patches). Moreover, using a fully convolutional network with end-to-end training process allows them to handle patches jointly thus relative importance of the patches for estimating the illuminant can be well explored.

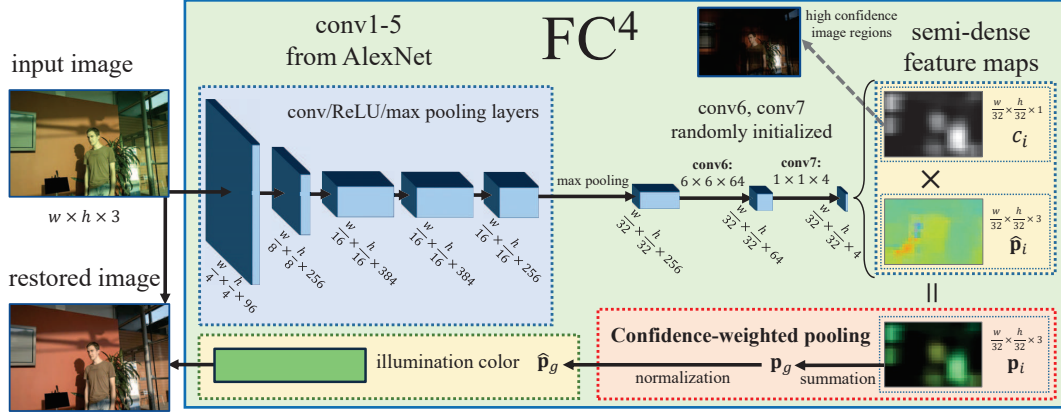


Figure 2.24: The architecture of *AlexNet* – FC^4 [39]

FC^4 computes an array of illuminant estimations and their respective confidences. Parameters for both estimations and the confidences are learned by a single end-to-end training phase. Higher confidence is associated with a lower expected angular error. The confidence maps are not only meaningful for a single image. A detailed example of inference process by (FC^4) is shown in Figure 2.25.

Their conclusions [39] on confidence mappings on several images (Figure 2.26) stated that the network especially weights the bright pixels, faces, surfaces with rich texture, and objects with a restricted range of color (especially achromatic) more. These observations are consistent with previous researches. For example, specular reflection usually results in bright pixels and it is directly related to the color of the light source. Likewise, when the surfaces with rich texture are considered, this observation can be supported by [16] where they argue that the second-order gradient information is sufficient for illuminant estimations.

Another novel approach, Cascading Convolutional Color Constancy (C^4) is proposed by Yu et al. [61]. They used FC^4 [39] as a baseline. The presented architecture ensembles three dependent cascaded stages where each of them generates an illuminant estimation. Each stage corrects the output of the previous stage resulting in a boost in illuminant estimation performance. A visualization of the process is shown in Figure

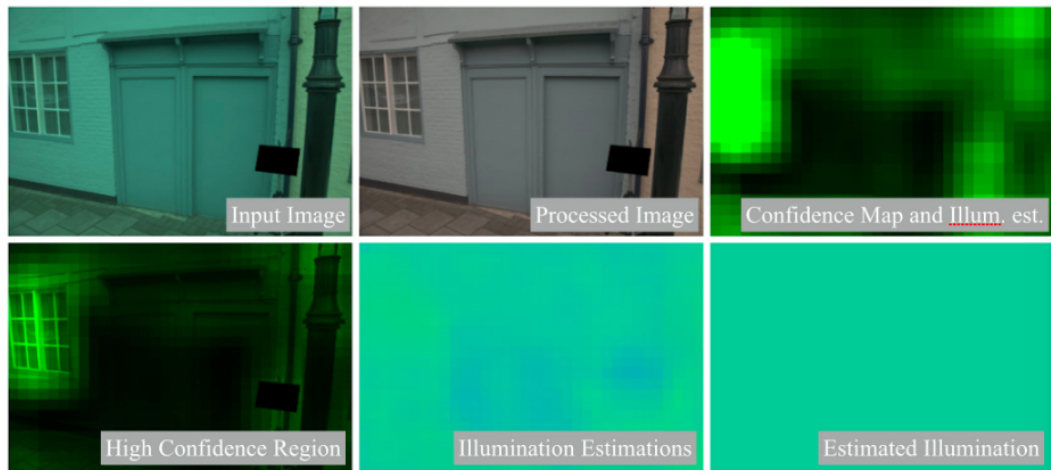


Figure 2.25: A detailed examination of illuminant estimation process of FC^4 . For a given image, a confidence map and illuminant estimation map is computed. Then a global estimation is found by weighted average of each local estimation and their confidences. While surfaces which are rich in terms of texture are assigned with a higher confidence, smooth surfaces are associated with a low confidence [39]. The image is taken from github.com/yuanming-hu/fc4: How to make inference on images based on a trained model?

2.27.

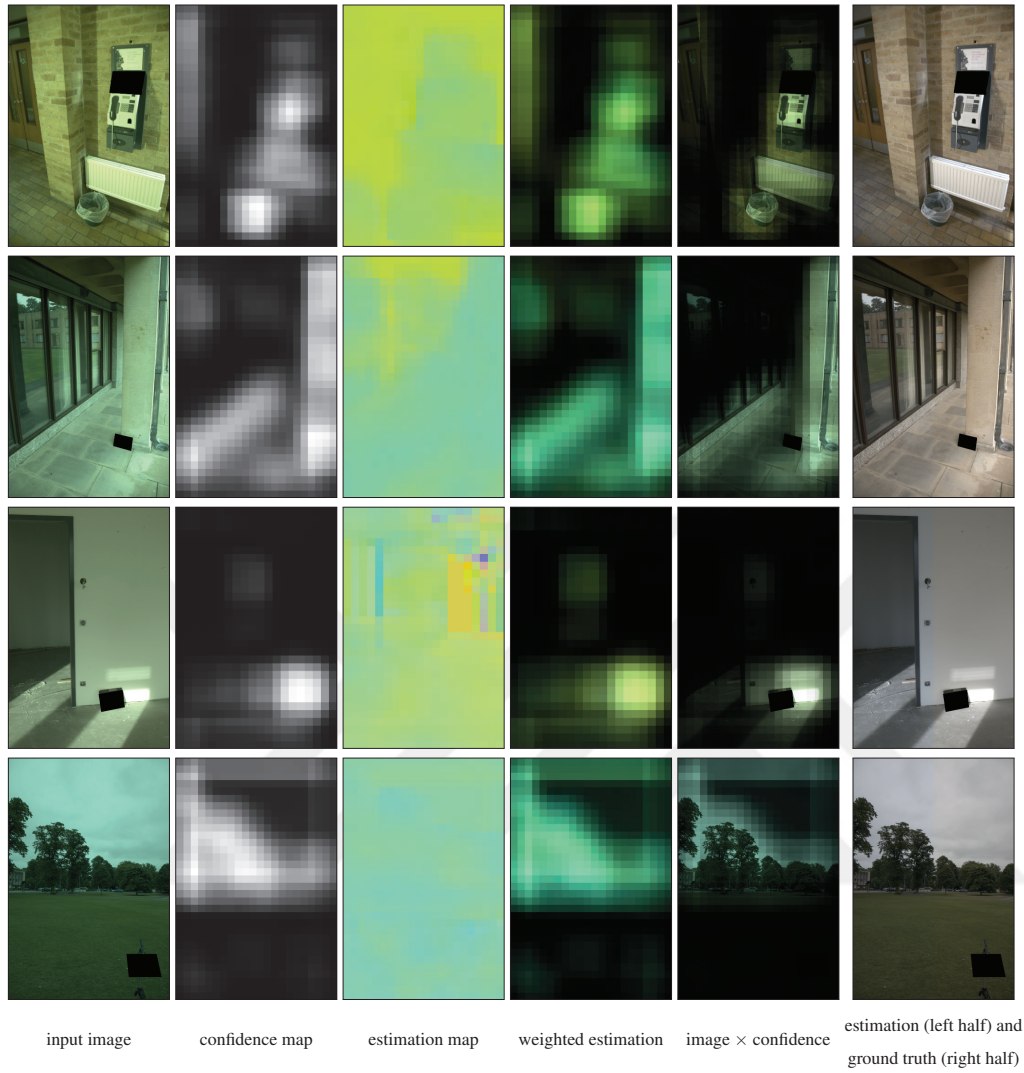


Figure 2.26: Test outputs of FC^4 on Reprocessed ColorChecker Dataset[35]. The image is taken from github.com/yuanming-hu/fc4: [Visual Results: More](#). The first column consists of poorly white balanced input images. Second and third columns present the assigned confidence map and estimation map for each region. Fourth and fifth map are weighted illuminant estimations for each region and image masked by the confidence map respectively. The last column shows the corrected image (by diagonal model) and the ground truth image. The confidence maps for the input images in third column highlight regions rich in terms of texture or regions substantially bright.

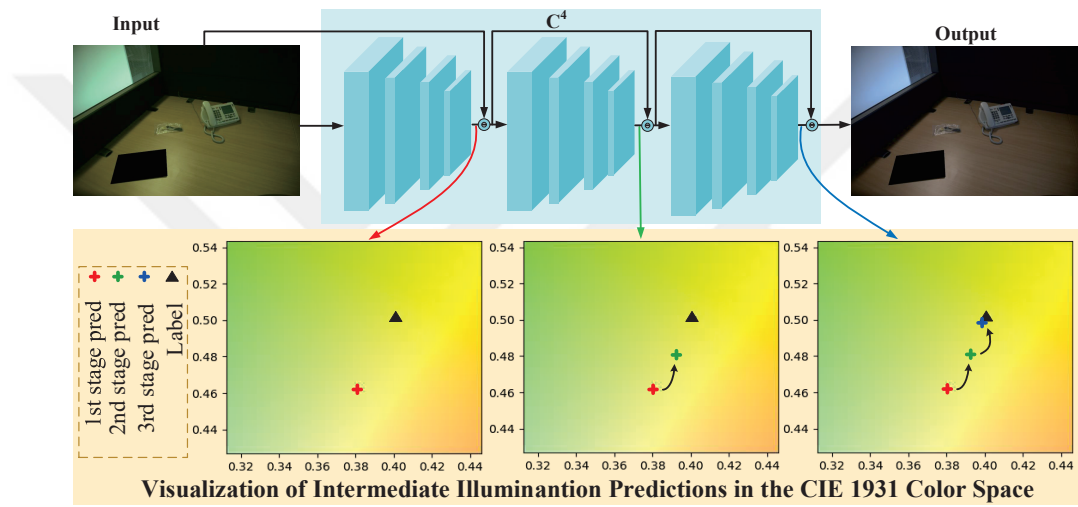


Figure 2.27: Visualization of the proposed method C^4 . The cascaded structure is shown top row. Bottom row shows the intermediate illuminant estimations. The cascaded system refines the estimation thus resulting an increase in performance [61].



CHAPTER 3

COLOR COHERENCE OF DIFFERENT OBJECT CATEGORIES

In this chapter, we analyze the color distributions of common objects. First, the motivation for this study is briefly explained. Following this section, proposed similarity metrics are introduced. Finally, intra-class color properties of object classes are compared by using the proposed similarity metrics. A discussion is also held under some of their common characteristics.

3.1 Motivation

Color constancy is an underdetermined problem where only the pixel values of the image are known. The problem is simplified to Equation 2.5 based on the assumptions explained in detail in Section 2.1.2. However, this simplification is not enough alone to calculate the illumination source's spectral distribution and some further assumptions or some prior information is still obligatory.

We encounter thousands of different objects during our lives resulting in observations that form prior information about our surroundings. For example, a zebra is expected to be observed white with black stripes. Therefore, when we look at a zebra in an image, if it is not black and white we can assume that the color shift is likely to be the result of the illuminant source. Another example is the grass. When we look for grass to sit on, we most likely search for a green color first, since we expect the color of grass to be a shade of green. On the other hand, some objects can be associated with a wider range of colors. For example, hats are produced in a great variety of colors: red, green, blue, white, etc. As a result, two hats can be identical in every aspect except their colors. Thus, if a hat is presented in an image it is hard to predict



Figure 3.1: Images are taken from Microsoft COCO [45]. The first row includes images containing a bus. The second row includes examples of images including bananas. Busses are less suitable for a color constancy application while bananas might give a clue for illuminant source.

its color and the illuminant source.

Consequently, prior information on how an object class is encountered in real life can be used as a clue for illuminant estimation, if such objects are presented in the image. For example, it is quite difficult to distinguish in an image, whether an observed red wall is actually a white wall under red light or a red wall under white light. However, if there is an object in front of the wall that we have prior information about its color such as banana or grass, an unexpected shift in this object's color can give a clue for the illuminant source. If, in the scene, there are more than one objects with prior information are presented, their individual illuminant estimations can also be combined. Furthermore, if an object is more stable in terms of color, its prediction can be weighted more, while forming the global illuminant estimation. For instance, grass might be in any shade of green and sometimes even yellow or brown, while a zebra or sheep are most likely to be achromatic.

Considering the recent developments in object detection and segmentation algorithms that are based on deep networks, nowadays we are much more successful at segment-

ing objects pixel by pixel. Therefore, it is easier to analyze object classes that are commonly focused on by the segmentation algorithms based on their applicability to the color constancy problem. Following a detailed study on the object classes and their associated color properties, white balancing performance for an image can be increased by exploiting the object color characteristics.

Only a small fraction of past works explicitly focused on colors associated with object classes. For instance, a novel approach [58] generates illuminant estimations based on the object classes in the image. Then, for every object class, an illuminant estimation is calculated. Finally, estimation resulting in the most likely correction of image is select. Another technique [50] proposes fine-tuning of white balance by using detected object classes (e.g. sky, grass) in the image. Also, [11] uses a gamut mapping to segmented face pixels in an image. The study showed that skin pixels form a compact region in HSV color space and this characteristic might be used as a clue for illuminant estimation.

It is important to note that throughout the paper, and in common with most previous literature, we assumed that the observed light for all pixels can be modelled as a function of the incoming light and the object surface. The object surfaces are characterised by a spectral distribution ($S(x, \lambda)$) which might not be appropriate for all object classes (e.g. sky, water, window). Classification and utilization of object classes in an image in terms of how well they adapt th model might increase the overall performance of object based approaches.

Alongside with the recent developments in object detection and segmentation tasks, now hundreds of object classes are labeled in semantic segmentation datasets. In this chapter, we present a study that analyzes a wide range of object classes in terms of two proposed color based similarity and argue about their applicability to color constancy problem.

3.2 Proposed Color Similarity Metrics

In this section, we propose two similarity metrics for object classes. The first one specifies the average color of the class, whereas the second one defines the scatter

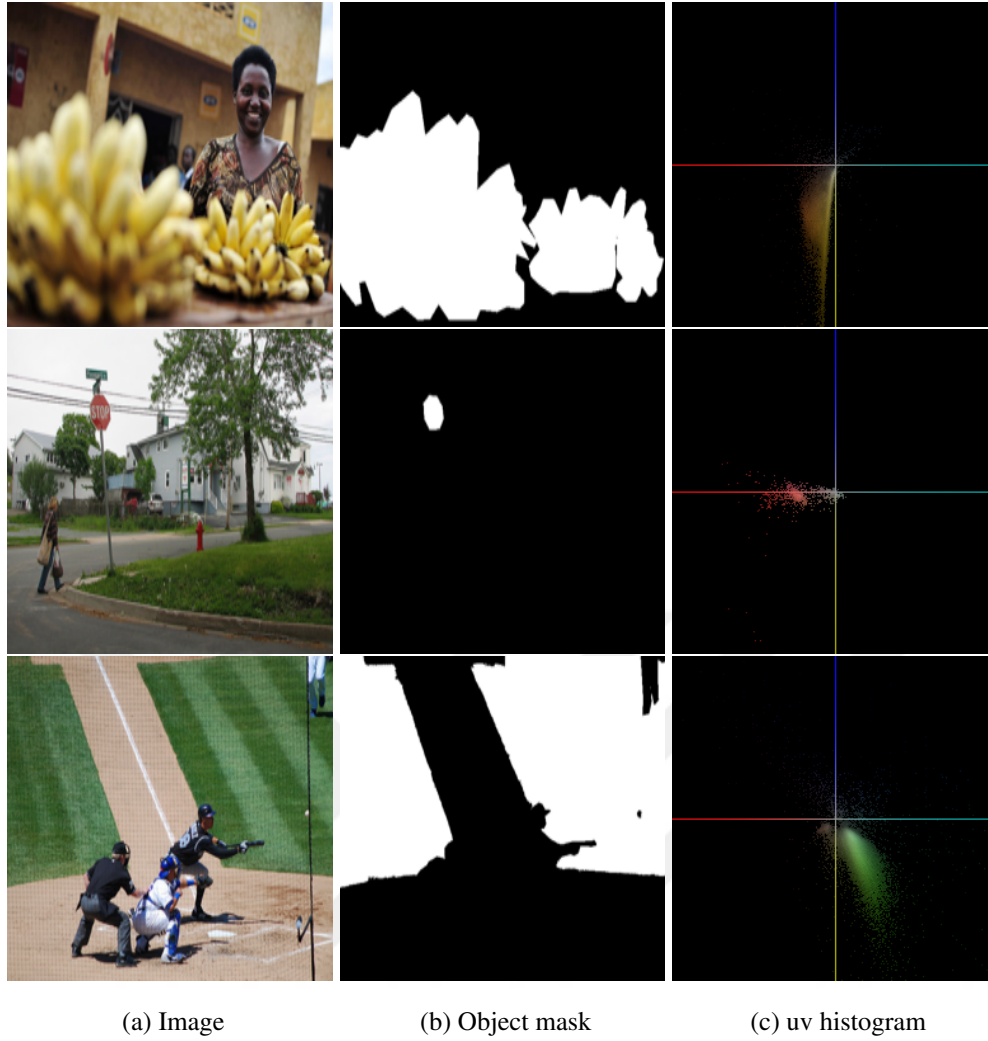


Figure 3.2: Example uv histogram representations of images from COCO dataset [45]. Panoptic segmentation ground truth masks provided by COCO are used.

of the pixel values of the class. More detailed explanations on the proposed metrics are available in Section 3.2.2. These attributes are later used to calculate intra-class properties of the object classes for determining how stable the color of an object is and how stable the distribution of an object’s color, respectively.

3.2.1 Selection of Color Space

In this section, a 2-D color space, inspired by Barron [5] is analyzed. The color constancy problem can be simplified to Equation 2.5 where the surface reflectance

$S_c(x)$ and the projected illuminant source $\mathbf{I}_c$ are unknowns defining the observed pixel values $f_c(x)$.

The observed image is the multiplication of the surface reflectance $S(x)$ and the illuminant source $\mathbf{I}_c$. If we take the logarithm of the equation, one can represent the illuminant effect by simple addition, instead of multiplication. In Equation 3.1, two new chrominance measures: u and v and a luminance measure y are presented in terms of r, g, b pixel values:

$$\begin{aligned} u &= \log \frac{g}{r}, & v &= \log \frac{g}{b} \\ y &= \sqrt{r^2 + g^2 + b^2} \end{aligned} \quad (3.1)$$

At this point, if we rewrite the Equation 2.5 by using u and v , the illuminant effect can be modelled as a simple addition to the reflectance properties of the scene.

$$\begin{aligned} f_u(x) &= \mathbf{I}_u + S_u(x) \\ f_v(x) &= \mathbf{I}_v + S_v(x) \end{aligned} \quad (3.2)$$

Note that if we consider a white light source $\mathbf{I}_{rgb} = [1, 1, 1]^T$, its representation in u - v space is $\mathbf{I}_u v = [0, 0]^T$ and S_u and S_v can be considered as the chromaticity of the image under the white light source. Therefore, under white light, Equation 3.2 can be transformed to the Equation 3.3

$$\begin{aligned} f_u(x) &= S_u(x) \\ f_v(x) &= S_v(x) \end{aligned} \quad (3.3)$$

After estimating $\mathbf{I}_{uv}$, an inverse mapping from uv space to rgb space is required. Although the lost brightness information of RGB color space cannot be restored, it can be assumed to be a unit-form, allowing us to write Equation 3.4

$$\begin{aligned} r &= \frac{\exp(-u)}{z} & g &= \frac{1}{z} & b &= \frac{\exp(-v)}{z} \\ z &= \sqrt{\exp(-u)^2 + \exp(-v)^2 + 1} \end{aligned} \quad (3.4)$$

3.2.2 Individual Class Measurements

In this section, two similarity metrics for the chromatic properties of the semantic classes are defined: Average class color and class color distribution. The first one is the indication of the expected color of a typical object in a particular class. This value represents the color that is expected to be encountered when an object is detected under a white light source. The second metric is an indication of how the color of the objects of a class is expected to scatter.

In order to calculate these two measurements, we first define the normalized joint histogram $M(u, v)$ where $f_u(x)$ and $f_v(x)$ representing the chromaticity values of observed object at x . $M(u, v)$ counts the number of pixels with a chromaticity difference less than $\frac{\epsilon}{2}$ and weights each pixel by its luminance value in Equation 3.5.

$$M(u, v) = \sum_i f_y(x) \left[\left(|f_u(x) - u| \leq \frac{\epsilon}{2} \right) \cap \left(|f_v(x) - v| \leq \frac{\epsilon}{2} \right) \right] \quad (3.5)$$

We also normalized the joint histogram $M(u, v)$. This normalization equalizes the effect of object instances during the calculation of the class properties. The resulting normalized histogram $H(u, v)$ is shown in Equation 3.6

$$H(u, v) = \frac{M(u, v)}{\sum_{u', v'} M(u', v')} \quad (3.6)$$

In Figure 3.3, nine different histogram representations in uv space of an image is given. When insufficient parameters are selected, the histogram cannot represent the colors perfectly. For example, in Figure 3.3a range of histogram is not enough to represent image colors. Therefore both the shape and the mean of the histogram are misrepresented. Our approach strongly depends on the histogram. Therefore, we selected histogram parameters as $\epsilon = 0.025$ and $N_{bin} = 256$ so it can cover a wide range of colors. Moreover, in order to reduce the effect of the outlier images, during our experiments we discarded any object with at least 25 % of its pixels are out of the range of the histogram. CCC [5] also used the same histogram parameters with us to represent images.

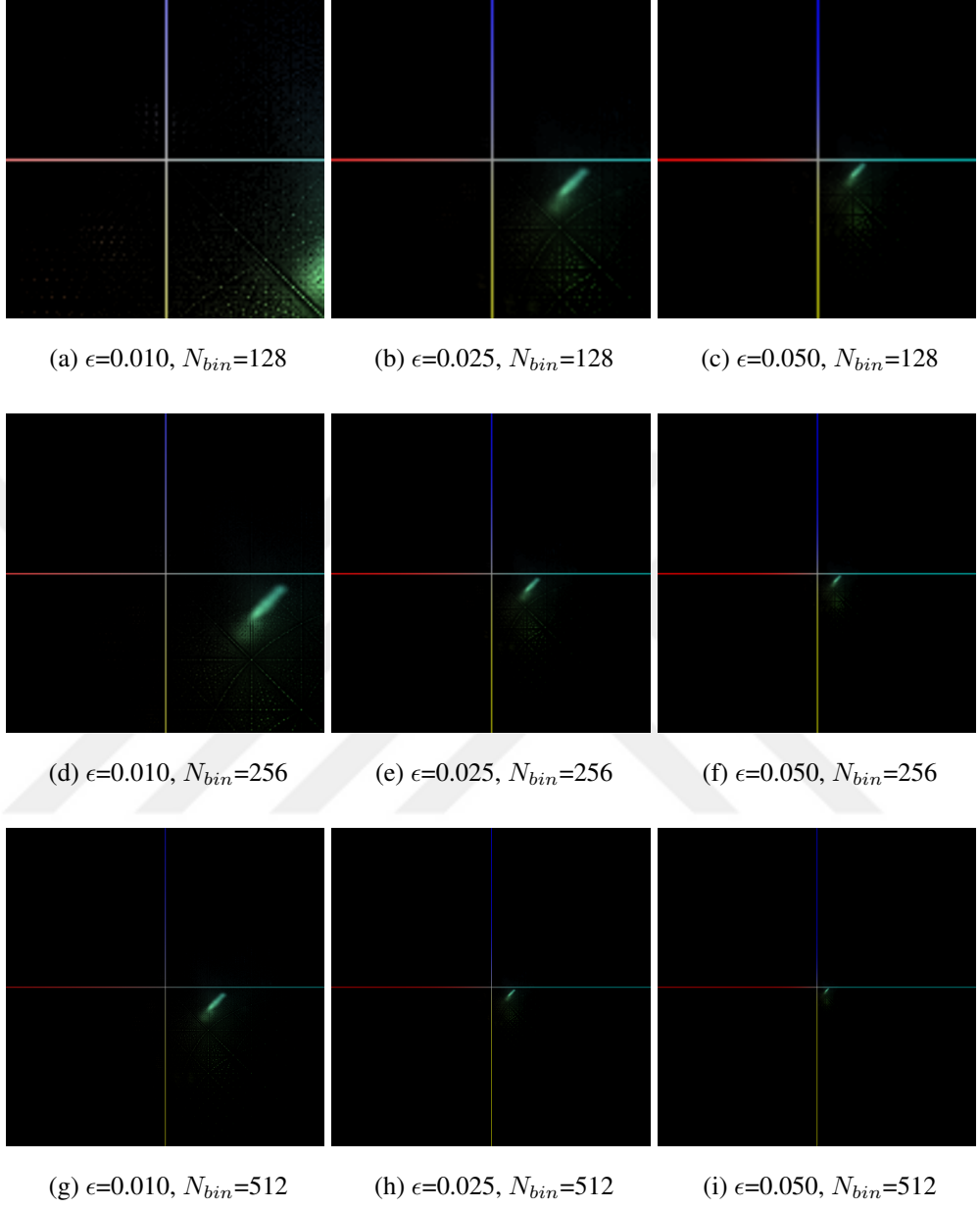


Figure 3.3: Presentation of the same image with different ϵ and N_{bin} parameters. We selected $\epsilon=0.025$, $N_{bin}=256$ because it fully covers the range of corrected REC images.

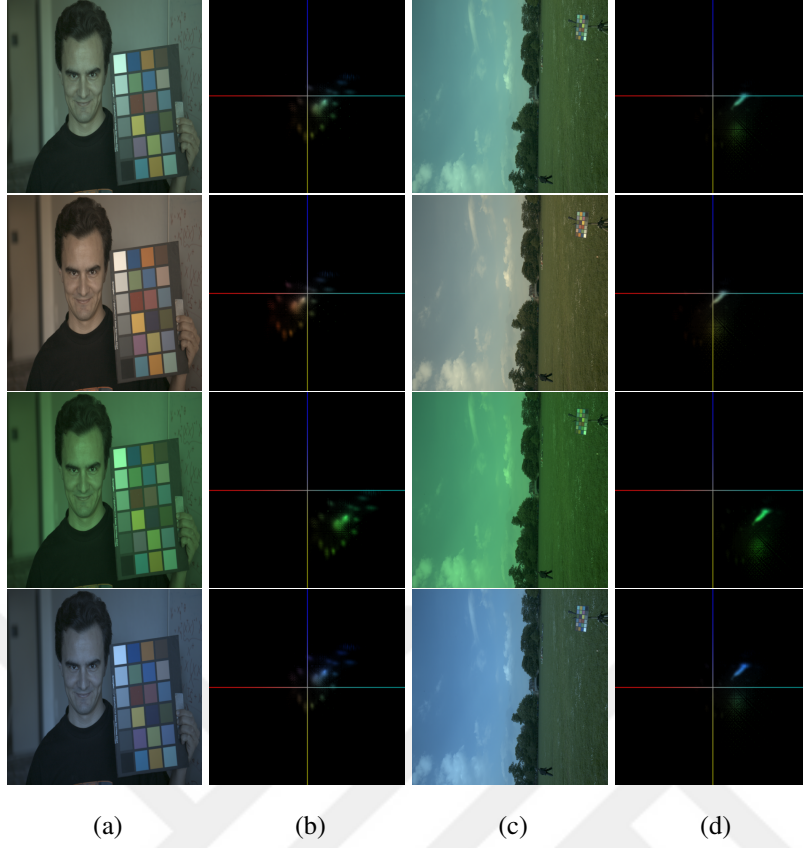


Figure 3.4: Histogram representations of some images from REC [35]. First and third columns show images rendered with different colors while second and fourth columns shows their respective histogram representations. Only gamma correction is applied for visualization purposes.

Expressing the color distribution of an object through the proposed histogram brings an important advantage over the RGB color space. Any change of the illuminant source results in only a shift in the histogram. In theory, the distribution stays the same while its mean is relocated by the illuminant source.

In Figure 3.4, 2 images and their rendered versions under different colors are presented. Each image is given with its histogram representation next to it. The second and fourth columns shows that change in illuminant affects the position of the representation not its shape. In other words, shape of the histogram is illuminant invariant and only its position is affected by a global illuminant change.

We modeled an object instance’s histogram as a Joint Normal Distribution $G(\mu, \Sigma)$.

Mean μ is directly affected by the illuminant source and represents the color of the object. On the other hand, Covariance Matrix Σ represents its distribution and invariant to the light source.

The first metric is the average color of the class. For every object instance, first, the image is masked using a segmentation mask. Then the masked image is converted to uv color space. If the number of remaining pixels P is less than a pre-defined threshold T_p , the object is neglected to increase the reliability of the metrics. The number of pixels an important parameter, especially when computing the sample covariance matrix Σ .

$$\begin{aligned}\mu_u^i &= \frac{\sum_n^P y_n u_n}{\sum_n^P y_n} \\ \mu_v^i &= \frac{\sum_n^P y_n v_n}{\sum_n^P y_n}\end{aligned}\tag{3.7}$$

The average color of object instance i ($\mu^i = [\mu_u^i, \mu_v^i]^T$) is calculated via weighted summation which is shown in Equation 3.7. Luminance values of the pixels (Equation 3.1) are used as weights. This decreases the effect of the unbright pixels and information loss due to color space transformation.

$$\begin{aligned}cov^i(u, v) &= \frac{\sum_n^P y_n (u_n - \mu_u^i)(v_n - \mu_v^i)}{\sum_n^P y_n} \\ var^i(u) &= \frac{\sum_n^P y_n (u_n - \mu_u^i)(u_n - \mu_u^i)}{\sum_n^P y_n} \\ var^i(v) &= \frac{\sum_n^P y_n (v_n - \mu_v^i)(v_n - \mu_v^i)}{\sum_n^P y_n}\end{aligned}\tag{3.8}$$

The second metric is the covariance matrix Σ^i of the object instance i . Sample covariance matrix calculation from [49] is used. Luminance values of pixels are used as weights again. Equation 3.8 shows the detailed derivation and Equation 3.9 shows the formation of Σ^i .

$$\Sigma^i = \begin{bmatrix} var^i(u) & cov^i(u, v) \\ cov^i(u, v) & var^i(v) \end{bmatrix}\tag{3.9}$$

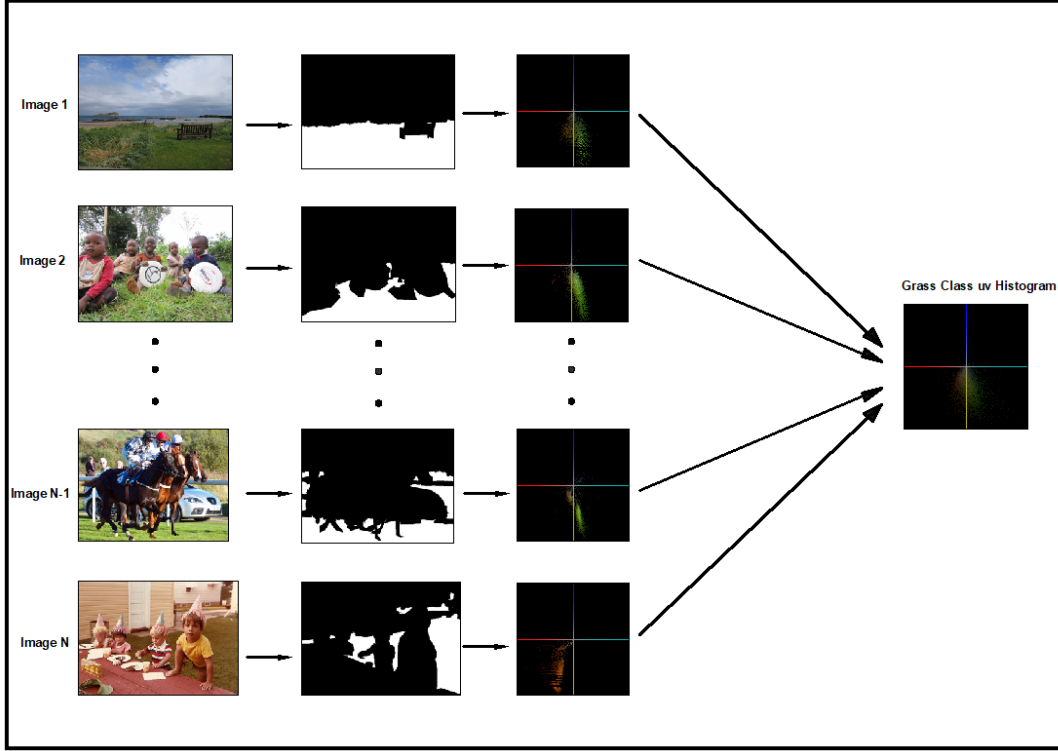


Figure 3.5: A representation of class measurement calculation. Images and object masks are taken from COCO [45]. This process is used to assign the proposed metrics for every class.

Until now, how proposed properties are calculated for a single object is showed. In order to calculate these measurements for the class, we simply summed all of object instance histograms for class c and calculated class mean μ_c and class covariance matrix Σ_c from the histogram summation. Normalization process (Equation 3.6) allows every object instance to contribute equally when calculating the class properties, invariant of their size.

$$H_c(u, v) = \sum_i^N H^i(u, v) \quad (3.10)$$

In Figure 3.5, class property calculation is demonstrated. Every image meeting the necessary conditions (3 channels, non-gray, having more than 100 pixels) are masked with ground truth. Then all of the histograms are summed up to a class histogram. Finally the class properties μ_c and Σ_c are computed using the class' histogram.

3.2.3 Intra-class Average Color Distance

In order to compare classes in terms of their color stability, we proposed intra-class average color distance d_μ^c where c is the interested class.

$$d_\mu^c = \frac{1}{N} \sum_i^N d_{L2}(\mu_c, \mu_i) \quad (3.11)$$

Intra-class average color distance is calculated by averaging the distance between every object instance's average color μ_i and class average color μ_c (Equation 3.11). The distance between the two average color is calculated by the l2 distance (Equation 3.12).

$$d_{L2}(p, q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2} \quad (3.12)$$

The measurement indicates how the average color of an object is scattered. A class with a higher intra-class average color distance d_μ^c is expected to be found in a wider range of colors. Classes with lower d_μ^c are found in more compact colors thus more appropriate to use in a color constancy method.

Table 3.1: Top 10 classes based on intra-class average color distances

Rank	Class Name	d_μ^c	Rank	Class Name	d_μ^c
1	snow	0.137	6	car	0.207
2	motorcycle	0.140	7	bicycle	0.229
3	zebra	0.166	8	sheep	0.237
4	skis	0.186	9	road	0.243
5	airplane	0.207	10	window-other	0.250

In Table 3.1, top ten classes with lowest intra-class average color distances are given. The most important common property of the classes is that they are achromatic. When we looked at the average colors assigned for the classes, we saw that all of them are close to the center of the histogram. In other words, they do not show a tendency to a color like orange. Snow, sheep, and zebra are colorless due to their nature and

only differ in brightness. Moreover, airplanes are also white and roads are gray. Both motorcycle and bicycle show an unexpected result for us. However, when we go back and look at the dataset, we observed that segmentation of bicycle and motorcycle includes road pixels as well. These object classes are found mostly on roads or pavements. Another interesting class on the table is the car. At first, car class seems to be some kind of a miss-calculation. However, although we expect the average color of car instances to scatter all around the color space, after an inspection of the dataset we saw that most of the cars are white, black, or gray.

Table 3.2: Bottom 10 classes based on intra-class average color distances

Rank	Class Name	d_{μ}^c	Rank	Class Name	d_{μ}^c
132	orange	0.946	127	playingfield	0.716
131	carrot	0.873	126	fire hydrant	0.728
130	floor-wood	0.855	125	stop sign	0.730
129	apple	0.842	124	fruit	0.756
128	wall-wood	0.783	123	cabinet-merged	0.783

In Table 3.2, ten classes with highest intra-class average color distances are given. Four of the classes are fruits. Although their average colors are what we expected to see (i.e. average color of the apple is still reddish and carrot is orangish), average colors of class instances scatter in the color space. We expect this to be related to two main reasons. The first one is the color space. Except for the playing-field, all object classes have extreme colors. By extreme color, we mean objects with much higher values in one channel and lower values in another. This kind of color structure easily results in fluctuations in uv color space. Also, the colors of the object are not equally susceptible to the noise due to the darker channels being more sensitive to noise than the brighter ones. On the other hand, colorless objects' channels are affected by the noise equally. A visualization of scattering of average colors for orange class and carrot class is given in Figure 3.6.

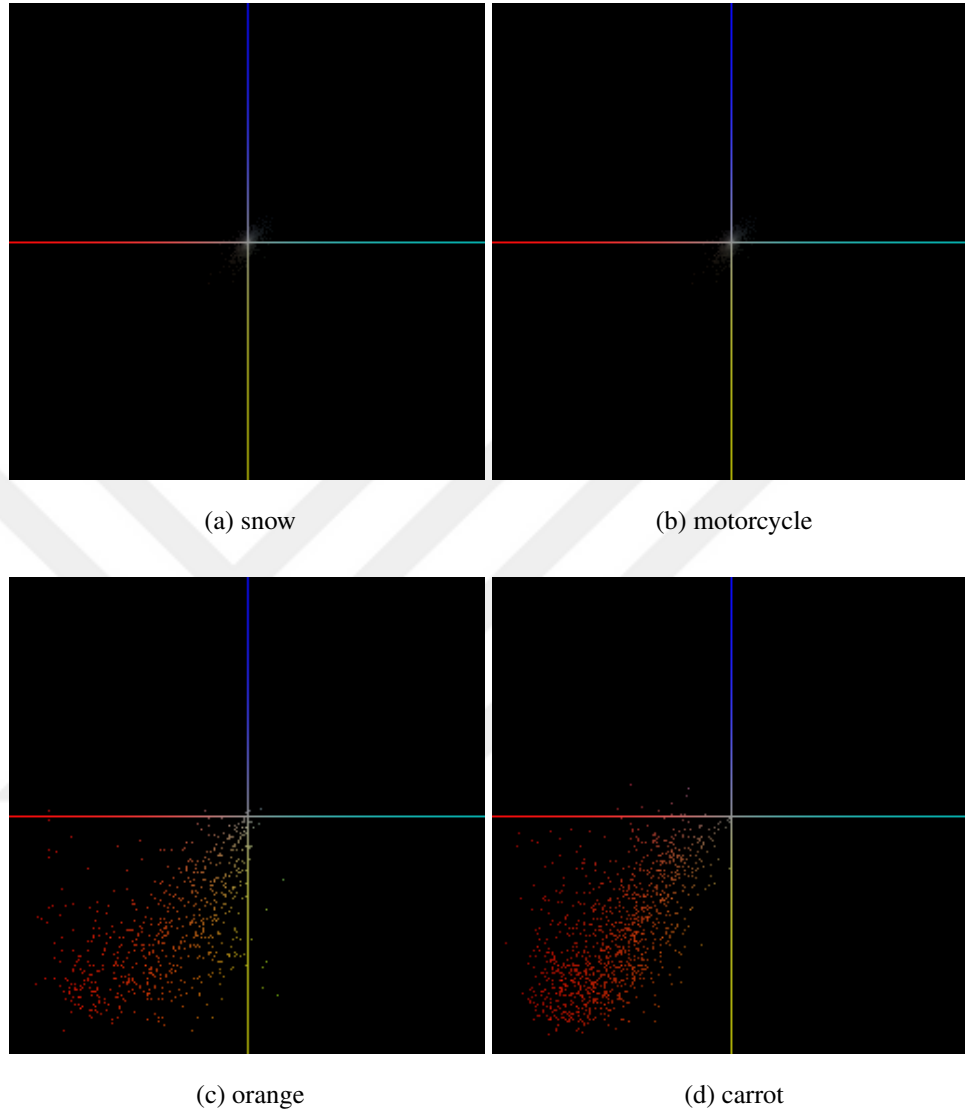


Figure 3.6: Histogram of average color of top two classes and bottom two classes based on intra-class average color distance. While scattering of average colors of apples and carrots can be seen easily, in snow and motorcycle histograms it is almost impossible to see any scattering.

3.2.4 Intra-class Color Scatter Distance

The covariance matrix is used to model the distribution of pixel values of an object. The most important advantage of this measurement is that for an object it is stable even under different light sources. In other words, this measurement is irrelevant to the illuminant source.

In this section, we define Intra-class Color Scatter Distance 3.13 as a measurement for the stability of distribution of a class among its instances. The metric is calculated by taking the average of the distance between the covariance matrix of the class and the covariance matrix of an object instance belonging to the class.

$$d_{\Sigma}^c = \frac{1}{N} \sum_i^N d(\Sigma_c, \Sigma_i) \quad (3.13)$$

It is hard to decide on the distance measurement method between covariance matrices Σ_1 and Σ_2 . In our experiments, We started with three different metrics for distance measurement and then compared their results. The first distance measurement is what we will call Herdin Distance, inspired [36]. Herdin Distance 3.14 can be considered as a measurement of orthogonality of matrices and can be formulated as the inner product between the vectorized Σ_1 and Σ_2 .

$$d_{herdin}(\Sigma_1, \Sigma_2) = 1 - \frac{\langle vec(\Sigma_1), vec(\Sigma_2) \rangle}{\|vec(\Sigma_1)\|_2 \|vec(\Sigma_2)\|_2} \quad (3.14)$$

The second distance measurement method we used is proposed by Wolfgang et al. [26] and for the rest of the thesis it will be called as wolf distance (Equation 3.15). $\lambda_i(\Sigma_1, \Sigma_2)$ are the eigenvalues from $|\lambda \Sigma_1 - \Sigma_2| = 0$

$$d_{wolfgang}(\Sigma_1, \Sigma_2) = \sqrt{\sum_{i=1}^n \ln^2 \lambda_i(\Sigma_1, \Sigma_2)} \quad (3.15)$$

The last measurement method is the Frobenius Norm as in Equation 3.16.

$$d_{frob} = \sqrt{\text{trace}(D^T \cdot D)} \quad (3.16)$$

$$D = \Sigma_1 - \Sigma_2$$

How to correctly measure the distance between covariance matrices is a hard question and strongly depends on the problem. They focus on the different properties of the matrices. We are interested in the measurement method which fits best to our problem.

3.3 Experimental Analysis of Color Coherence with Object Categories

3.3.1 Utilized Dataset

In this section, the Color-COCO dataset is introduced. We used COCO panoptic dataset [45]. COCO dataset has two available sets. The first one is the training set with 118000 images and the second one is the validation set consisting of 5000 images. All images are segmented and annotated using 133 distinct object classes.

We used the ground truth segmentation masks to select the objects from an image and calculate their individual class properties: Average class color and class color distribution. Then, these metrics are used to calculate intra-class average color distance (Section 3.2.3) and intra-class color scatter distance (Section 3.2.4). Lastly, object classes are compared based on their proposed similarity metrics.

COCO dataset consists of images taken under different conditions by different cameras. There is no information on the spectral distribution of the light source. We assumed that while taking pictures imaging devices have an active white balance module in their Image Signal Processing (ISP) unit. After discarding the gray images and outlier images in terms of average color, the remaining dataset is assumed properly white balanced. COCO is a broad dataset and a color shift in an image may be canceled by a color shift in inverse direction in another image. We assumed that the imaging devices' inner white balance methods do not shift towards a particular color and cancel each other's errors.



Figure 3.7: Examples images containing Macbeth Color Checker [35]

We used ColorChecker Dataset [35] to decide the range of properly white balanced images. ColorChecker dataset consists of 568 images containing a Macbeth Color Checker (Figure 3.7) containing 24 squares with six achromatic patches, six primary colors, and approximations of natural colors such as the blue sky or human skin. Ground truth illumination color for every image is calculated using the median of the RGB digital counts of the brightest achromatic patch without clipping. The original dataset is published by Gehler et al. [30]. The images were automatically produced and contained some artifacts (e.g. clipped pixels, non-linearity). Later Shi and Funt re-processed the original RAW images to avoid such problems [53]. Lately it is noticed that various ground truths for the Color Checker dataset exist. The ground truth of the dataset is revisited and a single ground is provided by Hemrit et al. [35].

We corrected every image of ColorChecker dataset and calculated the minimum chromaticities (u_{min}, v_{min}) and maximum chromaticities (u_{max}, v_{max}) of images. Following the calculation of the minimum and maximum chromaticity of the ColorChecker dataset, we discarded all images that are not in the range.

Another similar discard process is used after masking the objects with the ground truth provided by COCO. We used a weighted sample covariance matrix to model the distribution of colors of an object. We only used objects have higher than 100 pixels and did not count any object that does not meet the condition.

In reality images provided by COCO are not RAW images as most of the color constancy algorithms work on. The images are the results of a non-linear camera response function which consists of several steps such as tone manipulation, color en-

hancement, and a final gamma encoding. COCO dataset consists of images coming from various imaging devices. To make a generalization, we assumed that they can be linearized by a generalized gamma function (Equation 3.17) by using $\gamma = 2.2$. Although this is merely a reflection of truth, this approach is widely used. Moreover, a linearization step is crucial for our approach to reflect the true ratio of colors of the object classes.

$$I_{non-linear} = I_{lin}^{\frac{1}{\gamma}} \quad (3.17)$$

Lastly we used the ground truth illuminations provided by ColorChecker dataset and apply an artificial illumination on the remaining images. These intentionally improperly balanced images are used to compare our cost functions in Chapter 4.

After the utilization process, the Color-COCO dataset contains 111093 images where all images with an average color outside of the range of ColorChecker’s corrected images’ average color are discarded. All of the images are assigned with artificial illumination in a range of the ground truth illuminations of ColorChecker [35] set and then rendered into improperly white balanced images.



Figure 3.8: First row shows images tinted with a random color. This images used as inputs during experiments. The second row shows their ground truths directly taken from COCO dataset [45]. Average color of these ground truth images are between the average color of the corrected Colorchecker images [35].

3.3.2 Comparison of Object Classes Based on Categories

In this section, we examined the relationship between object classes in terms of their categories. For example [8], used an indoor/outdoor categorization for images. Following the categorization, they chose the best performing algorithm to estimate illuminate. Our motivation was to find if there are any common properties for object classes that are appropriate to color constancy in terms of our proposed metrics. For instance, we expected a more stable color for natural objects than human crafted objects. We expected to see different ties varying in color but we did not expect to see such an extreme deviation for an elephant.

In this section, three properties are assigned for each object: Indoor/Outdoor, Natural/Artifact, and Thing/Stuff. The first inspected category is Indoor/Outdoor, inspired by the work of [8]. The second category is Natural/Artificial. The underlying reason for this categorization is to investigate if there is a color characteristic (we modeled with Σ_c) that exist more in natural objects compared to human crafted objects. If it is, can it be a clue for illuminant estimation? Thing/stuff category is the last category that is compared. For every class, their category is assigned by hand for the first two categories. However, Thing/Stuff categorization is directly taken from COCO [45].

The object categories are compared using two main metrics. The first one is the expected distance between a random instance's average color and its class's average color. This metric measures how reliable the class average color for color constancy. For example snow class is expected to be white and if we know there is snow in the scene, it is probably white. On the other hand, if we consider the bus class, it is rather hard to make an assumption on the color of the objects. This metric indicates how stable the class' average color is. The second metric is the expected distance between a random instance's histogram covariance matrix and the assigned histogram covariance matrix for its class. We showed that change in illuminant source does not affect the distribution of class pixel histogram and only affects its mean. The metric is an indication of how well the color scattering of objects fit the class' model.

In Figure 3.9, a comparison of object classes in terms of their indoor/outdoor categorization is given. We also assigned *InOut* category for some object classes (i.e.

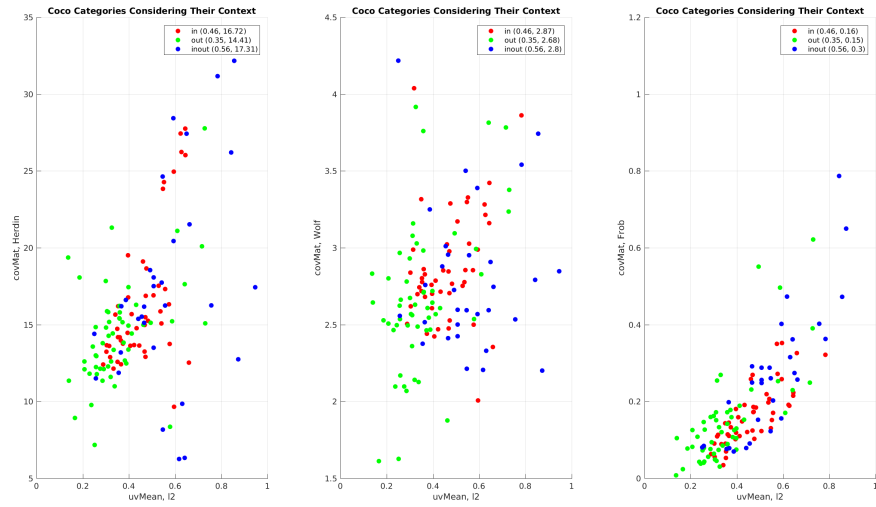


Figure 3.9: Comparison of different object classes based on their context. Horizontal x axis shows the intra-class average color distance while vertical axis indicates the intra-class average color scatter. Three figures are differ in terms of covariance matrix distance measurement methods. The methods shown are Herdin (Equation 3.14, Wolf 3.15 and Frobenius distance 3.16, respectively.

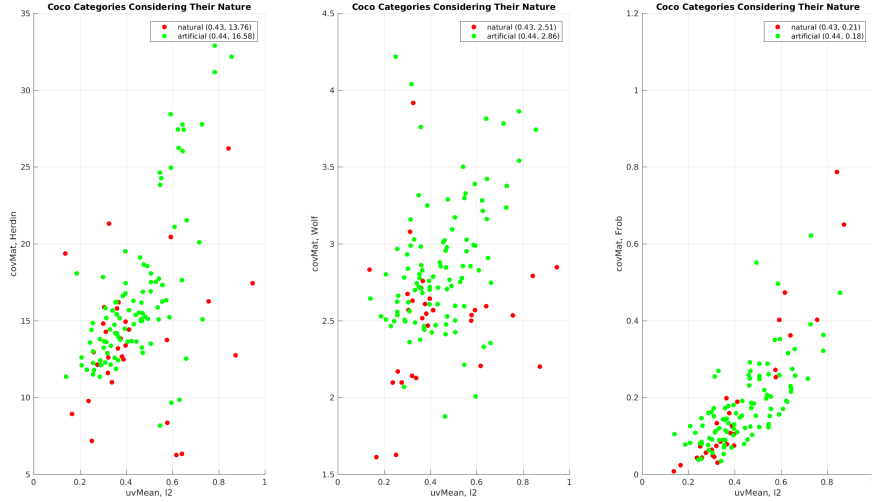


Figure 3.10: Comparison of different object classes based on their nature. Horizontal axis shows the intra-class average color distance while vertical axis indicates the intra-class average color scatter. Three figures are different in terms of covariance matrix distance measurement methods. The methods shown are Herdin (Equation 3.14, Wolf 3.15 and Frobenius distance 3.16, respectively).

banana, tie, window, or person) that can be observed in both indoor and outdoor. We saw that outdoor objects have the lowest intra-class average distance among them. This is in fact an expected result because outdoor scenes are expected to be both varying in terms of the color of the illuminant source and the reflective properties in the scene.

A general assumption for the color constancy problem is that all scene is only affected by a single light source. Moreover, the reflected light from an object to another one is neglected which is a situation quite likely to occur in indoor scenes. Also, multiple light sources might be affecting the scene as well. Consequently, outdoor scenes are more likely to be observed under common color constancy assumptions than indoor images, resulting in the images of objects taken outdoor being more consistent.

We also compared the object classes by their nature. We expected to see a more consistent covariance matrix for natural objects while a more random distribution in the histogram for human-crafted object classes. As an example, we expected to see

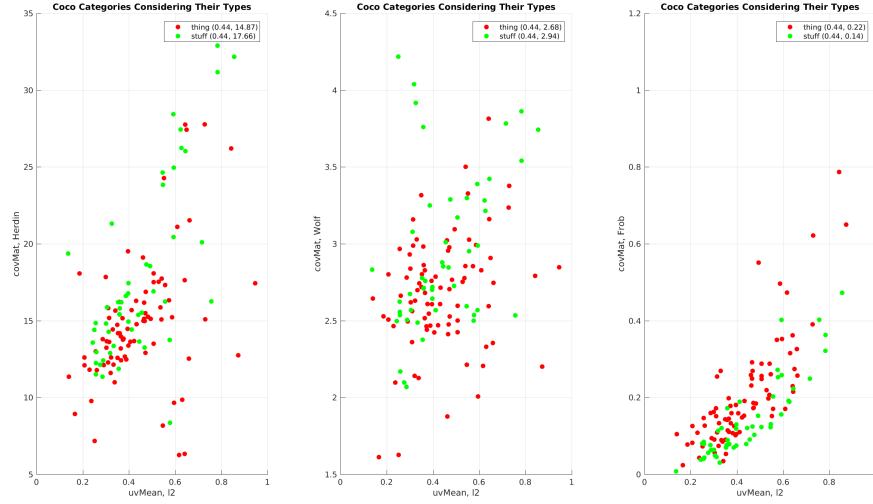


Figure 3.11: Comparison of different object classes based on their types. Horizontal axis shows the intra-class average color distance while vertical axis indicates the intra-class average color scatter. Three figures are different in terms of covariance matrix distance measurement methods. The methods shown are Herdin (Equation 3.14, Wolf 3.15 and Frobenius distance 3.16, respectively).

similar variations of the surface reflective properties through the surface of natural objects, resulting in a specific color distribution for the object class. However, the results (Figure 3.10) showed that there is no significant difference between the two categories.

The last comparison (Figure 3.11) was done by using the direct categorizations from COCO dataset. There are two categories: Thing and stuff. The former one consists of object classes that are often the target for object detection algorithms. Stuff type is introduced later, with the underlying purpose of being able to create a ground truth class for every pixel in an image. Road, pavement, river, and sand are some examples from stuff categories. There is not any distinct difference between thing and stuff categories. Their average intra-class color distance is almost the same. However, Frobenius norm usage showed a more apparent distinction of categories in terms of their intra-class average color scatters.

It is important to note that we used the selected images from COCO dataset. These

images are assumed to be perfectly white balanced by a WB algorithm embedded in the camera. Also, categorizations of the object classes are not strict. Moreover, objects are labeled in a variety of forms. For instance, both peeled and unpeeled apples are included in the same class.

3.3.3 Discussion

This chapter focuses on the object the classes and their color coherence. First, a color space is presented and its properties are discussed. Then, two similarity metrics based on the new color space are proposed, namely intra-class average color distance and intra-class average color scatter distance. Following the metrics, object classes from a frequently used segmentation dataset (COCO [45]) are compared in terms of their similarity metrics. Lastly, the object classes are categorized under three different categories to investigate if any common behavior exists.

It should be noted that the dataset we worked on was not perfect. First of all, the images in it are not perfectly white balanced. Original COCO consists of images taken under varying conditions. To decrease the effect of the imperfectly white-balanced images, we excluded all images whose average color is not in a pre-defined range. This range is computed by the average color of REC [35] images which are corrected by a ground truth illumination by the Macbeth ColorChecker in the image. The rest of the images are assumed to be white balanced good enough by the auto white balance module of the camera they are taken. Another imperfection of the dataset is it includes a different version of the object under its class. For example, peeled, immature, or over-matured bananas are labeled under the same class. Moreover, the labeling of the classes is not perfect all the time. A clear example of this problem can be seen in the segmentation of bicycles. Most of the time road which is seen from the gap of the tires are labeled as a bicycle as well.

The first comparison is made by using the intra-class average color distances of individual classes. A lower value corresponds to a more compact color space for random instances of the class. An obvious observation is that classes with the lowest intra-class average color distances tend to be achromatic. This is due to two reasons. Firstly, as the chromaticity of an object increases, the color ratio becomes more sus-

ceptible to be affected by the noise. Secondly, achromatic objects are more stable in terms of color. When we look at the grass class or banana, we see a range of green colors during their life period. On the other hand, it is hard to talk about a color change of an achromatic object class like zebra, snow, or grass. This observation is also consistent with what the network of FC^4 [39] tends to learn. The network tended to value the patches that include objects with a restricted color range, especially the achromatic ones.

The second comparison is based on the intra-class color scatter distances of object classes. We intuitively expected objects in the same class to demonstrate a similar distribution in uv histogram. Unfortunately, most of the time it is hard to model a consistent covariance matrix due to the low number of pixels of the segmented object. Also a comparison based this metric is not fair due to the number of objects with a small color distribution. For instance, when we talk about an object class with a small compact color distribution, covariance matrices of its instances are almost same as well. The most important characteristic of the metric is that it is independent from the illuminant source. Even though this metric might not be meaningful for all object classes, it might be useful for some classes to measure the similarity between a particular object and its class.

The last comparison is done by categorizing objects. The most prominent result was that the outdoor objects have significantly lower intra-class average colors than outdoor objects. This is an expected result due to outdoor scenes obeying more to the accepted assumptions better and a possible light source of the scene is more consistent (the sun). Usually, the scene includes a single distant light source while it is possible to have more light sources indoors. Moreover, for an indoor object to be affected by reflected light from another object is more realistic, considering the small and tangled structure of rooms.



CHAPTER 4

PROPOSED SEMANTIC COLOR CONSTANCY MODEL

In this chapter, we propose a novel cost function to determine the spectral content of the illumination. The proposed method is tested on two distinct data sets: Color-COCO dataset, Color Checker Dataset. The results are shared and compared with state of art approaches. Also the relation between the segmentation accuracy and the performance of the proposed method is discussed.

In Section 4.1, we present a cost function to utilize semantic object information in the image and the generalized version of our proposed method. Then, in Chapter 4.1.1, we explain three different variations of the proposed method, namely Vanilla, v-1, and v-2. The first one is the baseline version that only benefits from the objects and their associated colors. The second one includes pre-calculated weights based on the class' intra-class average color distance. In the last version, the method utilizes a similarity metric which measures the similarity between the detected object and its class.

In Section 4.2, we share our experimental results. First, we give the implementation and test details in Section 4.2.1. Following the explanation of our setup, object classes' individual performances on Color-COCO dataset are given in 4.2.2. After that, we evaluate the three variants of the proposed method on Color-COCO dataset which is the customized version of the segmentation dataset COCO [45] in Section 4.2.3. Then, we compared the effect of the accuracy of the segmentation mask in Section 4.2.4. Finally, in Section 4.2.5 we present our results on the latest version of popular color constancy dataset REC [35], originally published by [30].

4.1 Proposed Method: Semantic Color Constancy

In this section, we define a cost function with an aim to minimize the weighted contribution of each class's estimation to the global estimation. All of the estimations are done in the proposed color space (Section 3.2.1). This allowed us to use the simplified model (Equation 3.2) where the illuminant source's effect can be modelled as an addition constraint. The log chromaticity space is in fact used before with small additional processes by other state-of-art methods as well [5] [7] [54].

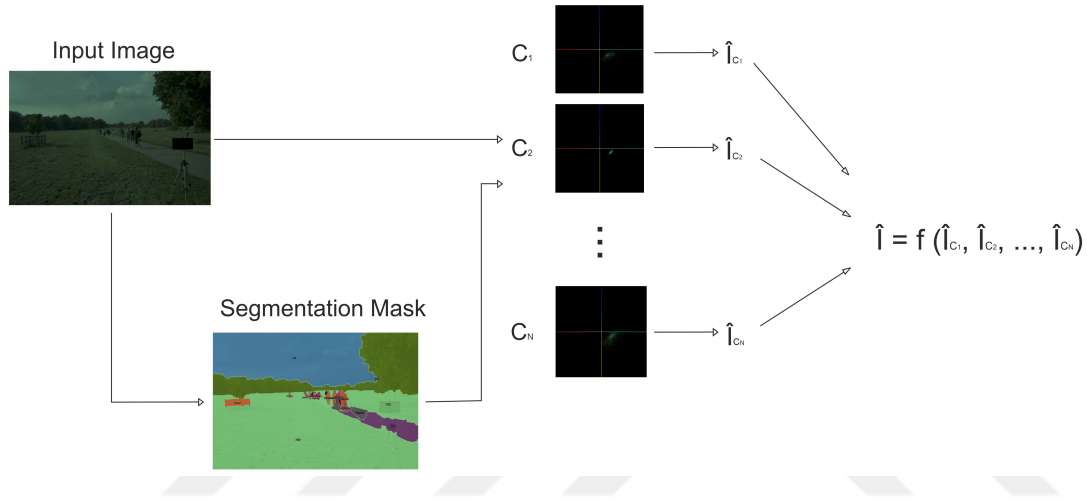


Figure 4.1: A visual representation of the proposed method.

The goal of the proposed approach is to benefit from the colors associated to object classes in Chapter 3 and the detected objects in the image. For every detected object belonging to the class c in the image, color of the light source can be individually estimated ($\hat{I}_c$). Then, these estimations can be combined to make a global illuminant estimation ($\hat{I}$) assuming that the effect of the illuminant source is global and uniform throughout the image (Equation 4.1).

$$\hat{I} = f(\hat{I}_c \mid c = \{grass, sky, apple, banana...\}) \quad (4.1)$$

We proposed a cost function of the global illuminant estimation $\hat{I}$, in Equation 4.2 where $\mu(I^c)$ is the average of the segmented object's pixels, μ_c is the class' learned

average color, and w_c is the class weight assigned to class c .

$$J(\hat{\mathbf{I}}) = \sum_{c=1}^N w_c \left\| \mu(I^c) - \hat{\mathbf{I}} - \mu_c \right\|_2^2 \quad (4.2)$$

After solving the function for the global illuminant estimation $\hat{\mathbf{I}}$, we get our generalized method (Equation 4.3), a weighted summation of the illuminant estimation of each detected object.

$$\hat{\mathbf{I}} = \frac{\sum_{c=1}^N w_c (\mu(I^c) - \mu_c)}{\sum_{c=1}^N w_c} \quad (4.3)$$

To gain a clear understanding of the proposed function, and the underlying idea under it, we defined the individual illuminant estimation for class c : $\hat{\mathbf{I}}_c$.

$$\hat{\mathbf{I}} = \frac{\sum_{c=1}^N w_c \hat{\mathbf{I}}_c}{\sum_{c=1}^N w_c}, \quad \hat{\mathbf{I}}_c = \mu(I^c) - \mu_c \quad (4.4)$$

4.1.1 Variants of the Proposed Color Constancy Method

In this section, we explained the three variants of the proposed method: Vanilla, v-1, and v-2. The first one is the most simple version which only utilizes the object class's color information without any discrimination between the classes. The second method (v-1) is the weighted version of the Vanilla implementation. This method weights object classes based on their applicability to the color constancy problem. The last version is v-2 which also benefit from the similarity of the detected object and how well it fits to its class' model on top of v-1.

We broke up the class weight w_c into three different weights: w_c^μ , w_c^Σ and w_c^{sim} . The first two are added to introduce the effect of intra-class metrics of the class, average the intraclass color distance and the average intra-class color scatter distance

respectively. Lastly, w_c^{sim} is used as a function of color scatter of the detected object and the class object belongs to (Equation 4.5).

$$w_c = w_c^\mu \cdot w_c^\Sigma \cdot w_c^{sim} \quad (4.5)$$

As a baseline implementation of the function, we treated every class equally. In other words, instead of taking the class's applicability to an illuminant estimation problem into account, we treated them as equal ($w_c = 1$). Consequently, this method takes a straight average of every individual object's illuminant estimation. We called this approach the vanilla implementation. The resulting function (Equation 4.6) becomes a simple averaging of individual object illuminant estimations.

$$\hat{\mathbf{I}} = \frac{1}{N} \sum_{c=1}^N \hat{\mathbf{I}}_c \quad (4.6)$$

Following the baseline implementation, we added a weight based on the intra-class average color distance. We valued the contributions of classes with low intra-class average color distance higher than the classes with high intra-class average color distance. In other words, the weight of a class scales inversely proportional to its intra-class average color distance. This variant (v-1) of our method (Equation 4.9) introduces pre-defined weights for classes, based on their applicability to the color constancy.

$$d_\mu^{max} = \max_c(\bar{d}_\mu^c), \quad \text{where } \bar{d}_\mu^c = \{d_\mu^c | c = 1, 2, 3, \dots, N\} \quad (4.7)$$

$$w_c^\mu = \frac{d_\mu^{max} - d_\mu^c}{d_\mu^{max}} \quad (4.8)$$

d_μ^{max} is the maximum intra-class average color distance of all classes (Equation 4.7). This is used to inversely scale the average color partition of class weight (Equation

4.8).

$$\hat{\mathbf{I}} = \frac{w_c^\mu \sum_{c=1}^N \hat{\mathbf{I}}_c}{\sum_{c=1}^N} \quad (4.9)$$

In the last variant (v-2), our goal was to include the similarity between the detected object and its class. To do this, we used the invariant proposed metric: covariance matrix of the object's histogram. We defined a similarity weight w_c^{sim} (Equation 4.10). This weight partition compares our object and its class' learned model. If the distance between the covariance matrix of the detected object and the covariance matrix of its class is higher than the expected covariance matrix distance between the class and its instances, we neglected the illuminant estimation completely. The more the color distribution of the object fits its class's color distribution, the more its estimation is taken into account.

$$w_c^{sim} = clip \left(\frac{d_\Sigma^c - d_{frob}(\Sigma^{obj}, \Sigma^c)}{d_\Sigma^c}, [0, 1] \right) \quad (4.10)$$

We also utilized the relationship between the relation between object-class average color distance and object-class covariance matrix distance. This weight partition w_c^Σ adds a weight based on how these two metrics are related, and thus how reliable the similarity weight w_c^{sim} is.

$$w_c^\Sigma = corr(d(\Sigma_c, \Sigma_{instance}), d(\mu_c, \mu_{instance})) \quad (4.11)$$

The resulting v-2 variant is shown in Equation 4.12.

$$\hat{\mathbf{I}} = \frac{w_c^\mu \cdot w_c^\Sigma \cdot w_c^{sim} \sum_{c=1}^N \hat{\mathbf{I}}_c}{\sum_{c=1}^N} \quad (4.12)$$

4.2 Experimental Results

In this section, we first share the implementation details of our experiments. Then, in Section 4.2.3 we evaluated the three variants of the proposed method on the Color-COCO dataset. Following the results on the Color-COCO dataset, we shared the experimental results on the Color-COCO dataset with the imperfect segmentation masks by Detectron-2 [60]. Finally, we tested our approach on a popular color constancy dataset, REC [35], in Section 4.2.5.

4.2.1 Experimental Setup

We used two different datasets during our experiments. One of them is an artificial color constancy set that we utilized to compute object color properties. The other one is a common color constancy dataset which is a standard in the literature. We also selected recovery error (Equation 2.6) as the error metric during evaluation.

The Color-COCO Dataset (Section 3.3.1) is an artificial dataset that we used to decide the similarity metrics of object classes. We modified the original COCO in two ways. First, we discarded all the images which are not in range of the REC’s corrected images’ color range. Thresholding is applied to eliminate any outlier image which is probably taken by a manual WB mode of the camera. The remaining images are used as ground truth images which are assumed to be taken under a white light source or perfectly white balanced by the camera’s inner white balance module. Secondly, we applied an artificial illuminant to all images where the random illuminant effects are selected from a uniform distribution between the extreme illuminant colors of REC. Moreover, during tests, we used the validation set that we have never used during the object property calculation process. The most important deficiency of the Color-COCO dataset is that its images might not be perfectly white balanced as we assumed them to be. All images in the dataset are assumed to be gamma corrected ($\gamma = 2.2$) and used their linearized versions in our experiments. Some examples of the dataset are given in Figure 4.2.

The second dataset used during experiments is REC [35]. For a color constancy algorithm, it is almost standard to measure its performance on it. The images in



Figure 4.2: Example images from COCO [45]. They are used as ground truth in experiments, assuming they are perfectly white-balanced.

the dataset include a Macbeth Chart. The ground truth illuminant of every image is computed using the gray patches of the chart. It consists of 568 images from two different cameras. [45]. Images in REC are not exposed to gamma correction and are used directly. In figures, we used their gamma-corrected versions. We also excluded the Macbeth Charts from the images during experiments. Some example images from REC are given in Figure 4.3.

Our implementation strongly depends on the accuracy of the segmentation masks. When we used the Color-COCO dataset, we used the ground truth masks from COCO. We can consider these masks close to perfect. On the other hand, REC does not have a ground truth for segmentation. When we experiment on REC, we used Detectron 2 [60] with a model trained on panoptic COCO.

We also used two additional methods: Gray World and Do Nothing to see the improvement of our method.



Figure 4.3: Example images from REC [35]. All images include a Macbeth ColorChecker Chart which is used to compute that color of the illuminant source.

Table 4.1: Top 10 classes with the lowest angular error on the Color-COCO dataset

Rank	Class Name	Mean Error	Rank	Class Name	Mean Error
1	snow	3.69	6	airplane	4.67
2	zebra	4.10	7	giraffe	5.00
3	motorcycle	4.33	8	bicycle	5.02
4	skis	4.36	9	sheep	5.19
5	car	4.51	10	window-other	5.29

4.2.2 Individual Semantic Class Performances for Color Constancy

In this section, we compared the performance of each class individually on the Color-COCO dataset. Only the images containing the class are taken into account. Moreover, objects with a lower number of pixels than a threshold are discarded. Best performing and worst performing object classes are shared in Table 4.1 and Table 4.2 respectively.

If we compare Table 3.1 and Table 4.1, we can see that top performing classes also have the lowest intra-class average color. When we also compare the Table 3.2 and Table 4.2, we again see the same object classes under-performing. These results confirms our expectations from the similarity metric: intra-class average color distance.

Table 4.2: Top 10 classes with the highest angular error on the Color-COCO dataset

Rank	Class Name	Mean Error	Rank	Class Name	Mean Error
123	playingfield	15.04	128	fire hydrant	16.68
124	wall-wood	15.13	129	apple	17.13
125	floor-wood	15.51	130	stop sign	17.13
126	fruit	15.80	131	orange	18.76
127	tie	15.84	132	carrot	18.93

4.2.3 Experimental Results on the Color-COCO Dataset

In this chapter, we evaluated the three variants of our method. Each method introduces another aspect of our approach to the model. We also included a simple Gray-World algorithm’s results and a Do Nothing which is as its name suggest represents the error of the dataset without any correction. In Table 4.3 the results can be seen.

To begin with, we used vanilla implementation where every detected object contributes equally to the final estimation. We can see a huge improvement over both Gray World and Do Nothing. Following the base implementation, we used the second variant of our estimation method (v-1), with class weights based on the class metrics, calculated beforehand. The resulting function weights all classes based on their intra-class average color distances, resulting in increased performance over the vanilla implementation. The implementation’s result confirms that applying a weight for each class based on their applicability to the color constancy problem increases the global estimation performance.

The third variant of our method (v2) tries to make a connection between the similarity between the color scatter of the detected object and its class’ color scatter. It also benefits from the pre-calculated color constancy applicability of the class. The underlying idea is to increase the contribution of the illuminant estimation of the object that fits their class’s model well. Moreover v2 variant completely neglects the object with a higher covariance distance between the object and its class than its class’s expected covariance distance.



Figure 4.4: Example images from the Color-COCO dataset. The first row shows the artificially illuminated images. The second row shows the ground truth images. Corrected images by our method’s estimation is given in the row column. The angular error between the ground truth and the estimated illuminant is given in top right corner of the predicted images.

In our final implementation, we optimized the v-2 by simply using different exponents for each class weight component. We achieved the best result when we use the value "7" as the exponent of w_c^μ .

It is important to note that we evaluated our model using an artificial color constancy dataset with almost perfect segmentation masks and artificial illumination effects. We also assumed that all the images in the dataset before the illumination effect is placed are perfectly white balanced. These experiments are valuable in terms of testing our initial assumptions. The vanilla implementation shows that an assigned color for each object class makes a meaningful assumption for the illuminant. Then, v-1 shows that a weighting system based on an object class’ metrics enhances the algorithm performance. This improvement shows that some object classes carry more reliable information about the illuminant source than others. The last variant v-2 showed the

Table 4.3: Results on the Color-COCO dataset

Algorithm	Mean	Median	Best25	Worst25
Do Nothing	17.43	18.20	8.06	25.54
Gray World [13]	9.36	7.58	3.54	18.04
Proposed Method-Vanilla	7.02	5.47	1.89	14.83
Proposed Method-v1	6.66	5.14	1.75	14.23
Proposed Method-v2	5.94	4.34	1.46	13.24
Proposed Method-v2 (optimized)	5.76	4.12	1.30	13.17

best performance. This method introduces a weighting based on the detected object and its similarity to its class model.

4.2.4 Effect of the Segmentation Mask on Algorithm Performance

In this section, we tested our methods on the Color-COCO dataset with segmentation masks generated by Detectron-2 [60]. The results are given in Table 4.4. We saw a similar pattern between the results with a ground truth segmentation mask and an automatically generated imperfect segmentation mask. Moreover, we saw a slight increase in terms of error in Table 4.4 when we compared it with Table 4.3, showing that the success of the segmentation process is an important factor affecting the performance of the algorithm. However, the performance increase is still maintained with the proposed method for every variant. Consequently, Table 4.4 confirmed our previous readings and demonstrated the importance of the segmentation mask.

4.2.5 Experimental Results on Color Constancy Dataset

Lastly, we tested our algorithm with one of the most commonly used datasets for color constancy. Because it is almost a standard for an illuminant estimation approach to testing itself on the ColorChecker dataset, we selected its latest version REC [35]. The dataset contains images from two different cameras, unprocessed and linear. There are also ground truth illumination values for each image. However, unlike COCO,

Table 4.4: Results on the Color-COCO dataset with Detectron-2 [60] masks

Algorithm	Mean	Median	Best25	Worst25
Do Nothing	17.43	18.20	8.06	25.54
Gray World [13]	9.36	7.58	3.54	18.04
Proposed Method-Vanilla	7.37	5.57	1.92	15.90
Proposed Method-v1	7.00	5.18	20.00	15.38
Proposed Method-v2	6.61	4.67	1.58	14.92
Proposed Method-v2 (optimized)	6.45	4.49	1.43	14.86

REC does not have a ground truth mask for object classes in it. We produced the segmentation masks by using Detectron 2 [60] with a pre-trained model on panoptic COCO dataset [45]. Some examples of the segmentation on REC by Detectron-2 can be seen in Figure 4.5.

The test results of the proposed method on REC is given in Table 4.5. We have taken the updated results from [35] except the algorithms which are shown with an asterisk sign. These algorithms’ results are taken directly from their paper where some of them are evaluated with the previous version of the ColorChecker dataset.

The table showed our method over-performs the statistical methods while achieving competing results when compared to the the state of the art learning-based methods. Besides, it should be noted that learning-based methods are trained on the REC itself while we learned all class metrics from an unrelated dataset without any ground truth illuminant information. The overall results confirm the contribution of object-class color association (vanilla method), a weighting system based on class’s applicability to the color constancy problem (method v-1), and a similarity metric between the detected object and its class (method v-2).

When we compared our results with a conventional method such as Gray World, we saw that especially for outdoor images our method can handle challenging scenes. For instance, in Figure 4.6 a comparison of two methods for two challenging images are given. The last column shows a good example of how the proposed methods utilizes the object classes and their color associations in challenging scenarios.

Table 4.5: Results on REC

Algorithm	Mean	Median	Best25	Worst25
Gray World [13]	9.7	10.0	5.0	13.7
White Patch [43]	9.1	6.7	2.2	18.9
Shades of Gray [24]	7.3	6.8	2.3	12.8
Pixel-Based Gamut [19]	6.0	4.4	1.7	12.9
Bottom-up+Top-down [58]	5.6	4.5	-	-
Natural Image Statics [32]	5.5	4.7	-	-
Bianco CNN [12]	2.4	1.4	-	-
BoCF* [42]	2.0	1.2	0.3	4.8
CCC (disc+ext)* [5]	2.0	1.2	0.4	4.8
FFCC (Model Q)* [7]	2.0	1.1	0.3	5.1
DS-Net(HypeNet + SelNet)* [54]	1.9	1.1	0.3	4.8
FC4 (Squeeze-Net)* [39]	1.7	1.2	0.4	3.8
Proposed Method-Vanilla	4.3	3.4	1.2	9.1
Proposed Method-v1	4.0	3.2	1.1	8.4
Proposed Method-v2	4.0	3.0	1.0	8.6
Proposed Method-v2 (optimized)	3.9	3.0	0.9	8.5

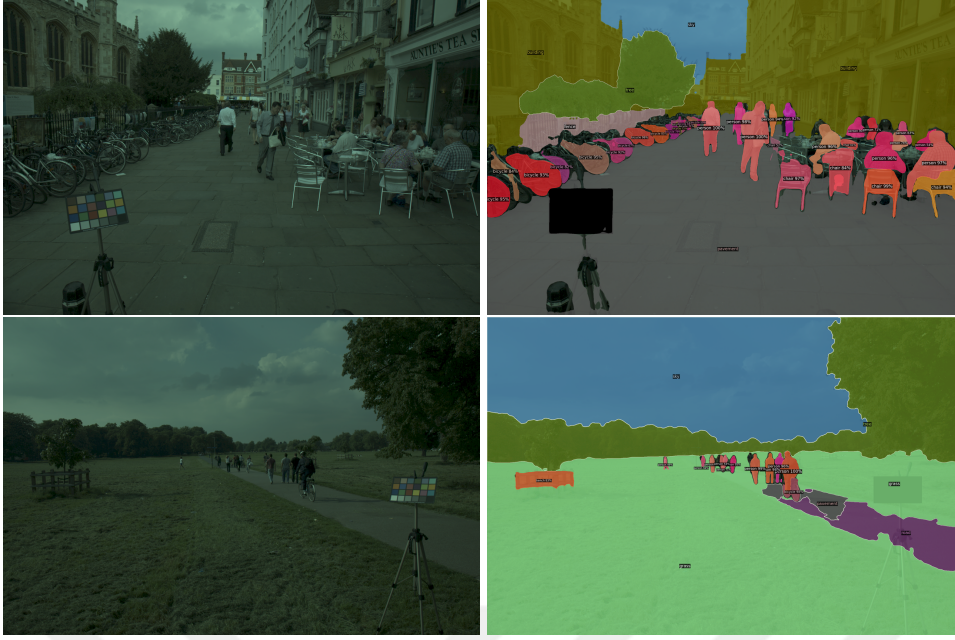


Figure 4.5: 2 example images from REC dataset [35] are shown in the left column and their respective segmented versions by Detectron-2 [60] are given in the right column.

Another observation was to see that the proposed method works much better on outdoor scenes than indoor scenes. When we checked the results, we noted that 16 of 20 best-predicted images were outdoor and 15 of the images with the lowest score were indoor. We expect this to be the result of two possible reasons. The first one is that our object class metrics are more consistent in outdoor scenes because the illuminant source is the sun most of the time. and it obeys our assumptions better. However indoor scenes are more not as consistent as outdoor scenes. Another reason is that the more reliable achromatic classes (i.e. road, pavement) are actually can be found outdoor more frequently.

In conclusion, our method shows better performance than the conventional methods and competes with the state of the art learning-based methods. An important detail here is that the learning-based methods' results are in-dataset results, using REC during training. On contrary, our method does not include any information from REC thus our performance should be considered as a cross-dataset performance. The results also showed that color association for object classes can be a clue for the color constancy problem. Additionally, we showed that inclusion of a weighting system



Figure 4.6: The figure shows two examples of how a prior information about an object class is able to decrease the angular error in a commonly used color constancy dataset [35]. The angular errors for the predictions are given in the top right corner of the images. Third fourth columns shows the predicted images for gray world and our method, respectively.

based on the similarity metrics in Chapter 3 resulted in a performance increase as well. Moreover, a similarity metric for the object and its class also enhances the overall performance. Another important feature of our method is that it is able to show a good performance in challenging scenes including reliable objects. Our method also performs better in outdoor scenes than indoor scenes which are consistent with our observation from Section 3.3.2.

4.3 Discussion

In this chapter, an object-based semantic color constancy method is proposed. Then, three variants of the proposed method are discussed. Following the discussion, individual performances of object classes are shared. The results confirmed the observation in section 3.2.3, object classes with low intra-class average color distance are more reliable in illuminant estimation.

In Section 4.2.3, proposed methods tested on the Color-COCO dataset. The results showed that our method shows a better performance than a conventional method using

individual information from objects. Moreover, adding a pre-computed class weight, reflecting the color constancy reliability of a class enhances the performance of the method. In addition to a pre-determined weight, adding another parameter representing the detected object and how well it fits its learned color model also increases the success of the weighting system thus the overall performance of the method.

In Section 4.2.4, the correlation between the success of the segmentation algorithm and the performance of the proposed color constancy algorithm is shown.

Lastly we evaluated our method for a popular color constancy dataset as well, in Section 4.2.5. The results are consistent with the results on the Color-COCO dataset. Furthermore, the proposed method over-performed the conventional statics-based methods while competing with the state of the art even though the object metrics are learned from a dataset with images lacking the ground truth illuminants.

CHAPTER 5

SUMMARY & CONCLUSIONS

5.1 Summary

This thesis introduces an object-based weighted color constancy algorithm.

In Chapter 2, first the color constancy problem is introduced. Then, the problem formulation is given in Section 2.1.1. This formulation is simplified with assumptions which are explained in detail in Section 2.1.2. Lastly, in Section 2.2, the past work relating to the color constancy problem is given. The past studies are explained briefly while methods that are more related to this thesis and methods with unique approaches are given in detail.

In Chapter 3, first our motivation for this study is shared. Then, the color space we have worked on is explained with details in Section 3.2.1. Two proposed class properties are defined for object classes and given in Section 3.2.2. Later, two intra-class metrics that can be used in a color constancy method are proposed. The metrics are followed by the detailed information on the utilized dataset. Lastly, based on the proposed metrics, object classes are compared based on three categories.

In Chapter 4.2, a generalized illuminant estimation method based on the detected objects in an image is proposed. Following the generalized method, three variants of the method are discussed in Section 4.1.1 where each variant includes another aspect of our approach. At first, individual performance of the object classes are compared in Section 4.2.2. Following the individual results, the proposed methods are evaluated on the utilized dataset and compared with the result of a conventional algorithm in Section 4.2.3. Then, the effect of the success of segmentation on illuminant estimation

is showed in Section 4.2.4. Lastly, the three variants of the proposed methods are also tested on a popular color constancy dataset in Section 4.2.5.

5.2 Conclusions

To begin with, the experimental results both on the Color-COCO dataset and REC indicate that the vanilla implementation of the proposed method successfully overperforms the conventional statistics-based methods. As an important conclusion of this result, it can be stated that estimating illuminant individually for each detected object by a learned color for their respective classes shows prominent results.

Moreover, a weighting system based on the similarity metrics defined in Chapter 3 is introduced. The proposed weighting system values each class in terms of their applicability to the color constancy problem. A variant of the proposed method (v-1) is tested and it showed an enhancement on the performance of the vanilla implementation. This enhancement validates that the reliability of the illuminant estimation of an object is related to its semantic class.

Following the proposal of the learned constant weights for each class, we further improve the weighting system by adding an adaptive class weight parameter to the cost function. This time we weighted an object's illuminant estimation by not only the reliability of its class, but also how well the object fits to its class' model. With this latest enhancement, our method increased its performance on both of the test sets.

Finally, we present the experimental results of the same dataset with different automatically extracted segmentation masks. The results showed a clear relationship between the performance of the object segmentation to the success of the illuminant estimation.

REFERENCES

- [1] M. Afifi. Semantic White Balance: Semantic Color Constancy Using Convolutional Neural Network, 2018.
- [2] K. Barnard. Improvements to gamut mapping colour constancy algorithms. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2000.
- [3] K. Barnard, V. Cardei, and B. Funt. A comparison of computational color constancy algorithms - Part I: Methodology and experiments with synthesized data. *IEEE Transactions on Image Processing*, 2002.
- [4] 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*, 2002.
- [5] J. T. Barron. Convolutional color constancy. In *Proceedings of the IEEE International Conference on Computer Vision*, 2015.
- [6] J. T. Barron, M. D. Biggin, P. Arbelaez, D. W. Knowles, S. V. Keranen, and J. Malik. Volumetric semantic segmentation using pyramid context features. In *Proceedings of the IEEE International Conference on Computer Vision*, 2013.
- [7] J. T. Barron and Y. T. Tsai. Fast fourier color constancy. In *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017.
- [8] S. Bianco, G. Ciocca, C. Cusano, and R. Schettini. Improving color constancy using indoor-outdoor image classification. *IEEE Transactions on Image Processing*, 2008.
- [9] S. Bianco, C. Cusano, and R. Schettini. Color constancy using CNNs. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2015.

- [10] S. Bianco, C. Cusano, and R. Schettini. Single and Multiple Illuminant Estimation Using Convolutional Neural Networks. *IEEE Transactions on Image Processing*, 2017.
- [11] S. Bianco and R. Schettini. Color constancy using faces. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2012.
- [12] S. Bianco and R. Schettini. Adaptive color constancy using faces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2014.
- [13] G. Buchsbaum. A spatial processor model for object colour perception. *Journal of the Franklin Institute*, 1980.
- [14] M. Buzzelli, J. van de Weijer, and R. Schettini. Learning Illuminant Estimation from Object Recognition. In *2018 25th IEEE International Conference on Image Processing (ICIP)*, pages 3234–3238, 2018.
- [15] V. C. Cardei, B. Funt, and K. Barnard. Estimating the scene illumination chromaticity by using a neural network. *Journal of the Optical Society of America A*, 2002.
- [16] D. Cheng, D. K. Prasad, and M. S. Brown. Illuminant estimation for color constancy: why spatial-domain methods work and the role of the color distribution. *Journal of the Optical Society of America A*, 2014.
- [17] P. Das, A. S. Baslamisli, Y. Liu, S. Karaoglu, and T. Gevers. Color constancy by GANs: An experimental survey, 2018.
- [18] G. Finlayson and S. Hordley. Selection for gamut mapping colour constancy. *Image and Vision Computing*, 1999.
- [19] G. Finlayson and S. Hordley. Improving gamut mapping color constancy. *IEEE Transactions on Image Processing*, 2000.
- [20] G. D. Finlayson. Color in perspective. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1996.

- [21] G. D. Finlayson, M. S. Drew, and B. V. Funt. Spectral sharpening: sensor transformations for improved color constancy. *Journal of the Optical Society of America A*, 1994.
- [22] G. D. Finlayson, S. D. Hordley, and P. M. Hubel. Color by correlation: A simple, unifying framework for color constancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2001.
- [23] G. D. Finlayson, S. D. Hordley, and I. Tasl. Gamut constrained illuminant estimation. *International Journal of Computer Vision*, 2006.
- [24] G. D. Finlayson and E. Trezzi. Shades of gray and colour constancy. In *Final Program and Proceedings - IS and T/SID Color Imaging Conference*, 2004.
- [25] G. D. Finlayson, R. Zakizadeh, and A. Gijsenij. The Reproduction Angular Error for Evaluating the Performance of Illuminant Estimation Algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.
- [26] W. Förstner and B. Moonen. A Metric for Covariance Matrices. In *Geodesy-The Challenge of the 3rd Millennium*. 2003.
- [27] D. A. Forsyth. A novel algorithm for color constancy. *International Journal of Computer Vision*, 1990.
- [28] B. Funt and W. Xiong. Estimating illumination chromaticity via support vector regression. In *Final Program and Proceedings - IS and T/SID Color Imaging Conference*, 2004.
- [29] B. V. Funt and G. D. Finlayson. Color Constant Color Indexing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1995.
- [30] P. V. Gehler, C. Rother, A. Blake, T. Minka, and T. Sharp. Bayesian color constancy revisited. In *26th IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, 2008.
- [31] A. Gijsenij and T. Gevers. Color constancy using image regions. In *Proceedings - International Conference on Image Processing, ICIP*, 2006.

- [32] A. Gijsenij and T. Gevers. Color constancy using natural image statistics and scene semantics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2011.
- [33] A. Gijsenij, T. Gevers, and J. Van De Weijer. Computational color constancy: Survey and experiments. *IEEE Transactions on Image Processing*, 2011.
- [34] A. Gijsenij, R. Lu, and T. Gevers. Color constancy for multiple light sources. *IEEE Transactions on Image Processing*, 2012.
- [35] G. Hemrit, G. D. Finlayson, A. Gijsenij, P. Gehler, S. Bianco, M. S. Drew, B. Funt, and L. Shi. Providing a Single Ground-Truth for Illuminant Estimation for the ColorChecker Dataset, 2020.
- [36] M. Herdin, N. Czink, H. Özcelik, and E. Bonek. Correlation matrix distance, a meaningful measure for evaluation of non-stationary MIMO channels. In *IEEE Vehicular Technology Conference*, 2005.
- [37] S. D. Hordley. Scene illuminant estimation: Past, present, and future. *Color Research and Application*, 31(4):303–314, 2006.
- [38] S. D. Hordley and G. D. Finlayson. Re-evaluating colour constancy algorithms. In *Proceedings - International Conference on Pattern Recognition*, 2004.
- [39] Y. Hu, B. Wang, and S. Lin. FC4: Fully convolutional color constancy with confidence-weighted pooling. In *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017.
- [40] G. J. Klinker, S. A. Shafer, and T. Kanade. A physical approach to color image understanding. *International Journal of Computer Vision*, 1990.
- [41] J. M. Kraft and D. H. Brainard. Mechanisms of color constancy under nearly natural viewing. *Proceedings of the National Academy of Sciences of the United States of America*, 1999.
- [42] F. Laakom, N. Passalis, J. Raitoharju, J. Nikkanen, A. Tefas, A. Iosifidis, and M. Gabbouj. Bag of Color Features for Color Constancy. *IEEE Transactions on Image Processing*, 2020.

- [43] E. H. Land and J. J. McCann. Lightness and retinex theory. *Journal of the Optical Society of America*, 1971.
- [44] H.-C. Lee. Method for computing the scene-illuminant chromaticity from specular highlights. *Journal of the Optical Society of America A*, 1986.
- [45] T. Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft COCO: Common objects in context. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2014.
- [46] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 2004.
- [47] M. Mosny and B. Funt. Cubical Gamut Mapping colour constancy. In *5th European Conference on Colour in Graphics, Imaging, and Vision and 12th International Symposium on Multispectral Colour Science 2010, CGIV 2010/MCS'10*, 2010.
- [48] N. Passalis and A. Tefas. Learning Bag-of-Features Pooling for Deep Convolutional Neural Networks. In *Proceedings of the IEEE International Conference on Computer Vision*, 2017.
- [49] G. R. PRICE. Extension of covariance selection mathematics. *Annals of Human Genetics*, 1972.
- [50] E. Rahtu, J. Nikkanen, J. Kannala, L. Lepistö, and J. Heikkilä. Applying visual object categorization and memory colors for automatic color constancy. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2009.
- [51] A. Savchik, E. Ershov, and S. Karpenko. Color Cerberus. In *International Symposium on Image and Signal Processing and Analysis, ISPA*, 2019.
- [52] L. Shi and B. Funt. Dichromatic illumination estimation via hough transforms in 3D. In *Society for Imaging Science and Technology - 4th European Conference on Colour in Graphics, Imaging, and Vision and 10th International Symposium on Multispectral Colour Science, CGIV 2008/MCS'08*, 2008.

- [53] L. Shi and B. Funt. Re-processed version of the gehler color constancy dataset of 568 images. accessed from <http://www.cs.sfu.ca/~colour/data/>, 2010.
- [54] W. Shi, C. C. Loy, and X. Tang. Deep specialized network for illuminant estimation. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2016.
- [55] O. Sidorov. Conditional GANs for multi-illuminant color constancy: Revolution or yet another approach? In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2019.
- [56] S. Tominaga and B. A. Wandell. Standard surface-reflectance model and illuminant estimation. *Journal of the Optical Society of America A*, 1989.
- [57] J. Toro and B. Funt. A multilinear constraint on dichromatic planes for illumination estimation. *IEEE Transactions on Image Processing*, 2007.
- [58] J. Van De Weijer, C. Schmid, and J. Verbeek. Using high-level visual information for color constancy. In *Proceedings of the IEEE International Conference on Computer Vision*, 2007.
- [59] J. von Kries. Chromatic Adaptation. In *Festschrift der AlbrechtLudwigs-Universitat*, pages 145–158, 1902.
- [60] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick. Detectron2. <https://github.com/facebookresearch/detectron2>, 2019.
- [61] H. Yu, K. Chen, K. Wang, Y. Qian, Z. Zhang, and K. Jia. Cascading convolutional color constancy, 2019.