Illumination estimation and segmentation in multi illuminant scenes

Vršnak, Donik

Sveučilište u Zagrebu

FACULTY OF ELECTRICAL ENGINEERING AND COMPUTING

Donik Vršnak

ILLUMINATION ESTIMATION AND SEGMENTATION IN MULTI ILLUMINANT SCENES

DOCTORAL THESIS

Zagreb, 2023

University of Zagreb

FACULTY OF ELECTRICAL ENGINEERING AND COMPUTING

Donik Vršnak

ILLUMINATION ESTIMATION AND SEGMENTATION IN MULTI ILLUMINANT SCENES

DOCTORAL THESIS

Supervisor: Professor Sven Lončarić, F.C.A

Zagreb, 2023

Sveučilište u Zagrebu

FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

Donik Vršnak

PROCJENA I SEGMENTACIJA OSVJETLJENJA U SCENAMA S VIŠE OSVJETLJENJA

DOKTORSKI RAD

Mentor: Akademik prof. dr. sc. Sven Lončarić

Zagreb, 2023.

This doctoral thesis was completed at the University of Zagreb Faculty of Electrical Engineering and Computing, on Department of Electronic Systems and Information Processing

Supervisor: Professor Sven Lončarić, F.C.A

The thesis has: 100 pages

Thesis No.: _____

O mentoru

Sven Lončarić diplomirao je i magistrirao u polju elektrotehnike na Fakultetu elektrotehnike i računarstva, 1985. i 1989. godine. Doktorirao je u polju elektrotehnike na Sveučilištu u Cincinnatiju, SAD, 1994. godine. U zvanje redoviti profesor u trajnom zvanju u polju elektrotehnike i polju računarstva na FER-u izabran je 2011. godine. Bio je suradnik ili voditelj na brojnim istraživačkim i razvojnim projektima u području razvoja metoda za obradu slika i računalnog vida. Od 2001. do 2003. bio je Assistant Professor na Sveučilištu New Jersey Institute of Technology, SAD. Voditelj je istraživačkog laboratorija za obradu slike na FER-u. Osnivač je i voditelj Centra izvrsnosti za računalni vid na Sveučilištu u Zagrebu. Suvoditelj je nacionalnog Znanstvenog centra izvrsnosti za znanost o podatcima i kooperativne sustave i voditelj Centra za umjetnu inteligenciju FER-a. Sa svojim studentima i suradnicima publicirao je više od 250 znanstvenih i stručnih radova. Prof. Lončarić redoviti je član Hrvatske akademije znanosti i umjetnosti. Prema studiji Sveučilišta Stanford objavljenoj 2022. godine rangiran je u 2% najutjecajnijih svjetskih znanstvenika u kategoriji umjetna inteligencija i obrada slike. Za svoj znanstveni i stručni rad dobio je više nagrada uključujući Državnu nagradu za znanost.

About the Supervisor

Sven Lončarić received Diploma of Engineering and Master of Science degrees in electrical engineering from the Faculty of Electrical Engineering and Computing in 1985 and 1989, respectively. He received Ph.D. degree in electrical engineering from University of Cincinnati, USA, in 1994. Since 2011, he has been a tenured full professor in electrical engineering and computer science at FER. He was a project leader on a number of research projects in the area of image processing and computer vision. From 2001-2003, he was an assistant professor at New Jersey Institute of Technology, USA. He founded the Image Processing Laboratory at FER and the Center for Computer Vision at University of Zagreb. Prof. Lončarić has been a co-director of the national Center of Research Excellence in Data Science and Cooperative Systems and the director of the Center for Artificial Intelligence at FER. With his students and collaborators he published more than 250 scientific papers. Prof. Lončarić is a full member of the Croatian Academy of Sciences and Arts. According to a Stanford University study published in 2022 he was ranked in the top 2% of the most cited world scientists in the category artificial intelligence – image processing. For his scientific work he received several awards including the National Science Award.

Preface

I would like to express my sincere gratitude to my thesis supervisor, prof. Sven Lončarić for his guidance and continuous support during my work on this thesis, as well as for his help and contribution to the scientific papers included in this thesis. Many thanks as well to prof. Marko Subašić for his guidance and support in my research.

I also wish to express special thanks to my parents Dunja and Bojan, family, and friends for their support and encouragement during my doctoral study and years of education, and for being an excellent support through it all. Without all of you, my PhD research (and life) would have been so much more difficult, and for that, thank you.

Abstract

Color constancy can be defined as the property of the human visual system (HVS) that allows humans to perceive the colors of objects independent of the color of the illumination that is illuminating them. However, the main mechanism behind it is still unknown and much research is being conducted in this area. While this property is sometimes challenged, the HVS can adapt and react quickly to a wide variety of changes in the illumination conditions. This means that objects remain perceived mostly the same in the changing illumination conditions. For example, a banana would look yellow under both sunlight, led or fluorescent light sources.

Computational color constancy, refers to a subset of computer vision problems, where the goal is to reproduce this property in digital image processing. Unlike the HVS, digital camera sensors do not possess this ability inherently. Thus, it is important to be able to recreate it using image processing. The process of computational color constancy can be done in two steps, illumination estimation and color correction (also referred to as white balancing). Out of those two problems, illumination estimation is the more difficult of the two. In the above-mentioned example with a banana, without any other knowledge, it is impossible to tell, just by looking at the RGB values, whether the banana is yellow and illuminated by white light, white illuminated by yellow light, or some other combination of those properties. This means that this problem is ill-posed. Furthermore, this problem can be split into two main categories, based on the number of illuminants present in the scene. Those categories are single-illumination and multi-illumination estimation methods.

The focus of this thesis is on multi-illumination methods. These can be separated into multi-illumination estimation methods and image segmentation methods based on the illumination. The goal of segmentation methods is to create a mask indicating the areas where only one illuminant is present. They are designed to work in concert with more well researched single-illumination estimation methods. Multi-illumination estimation methods, on the other hand, ideally produce an estimation for each pixel in the scene. In this work, the focus is on the implementation and analysis of methods for both multi-illuminant estimation and segmentation. They are implemented using deep-learning based techniques, and achieve state-of-the-art performance on their respective problems. Deep-learning models based on convolutional layers are selected because they show great performance and generalization capabilities on image processing tasks. Additionally, for deep-learning methods, the question of the training dataset is important. For the proposed task, a real-world dataset with 2500 images was used for scenes with two sources of illumination. For scenes with three or more sources, an artificial dataset was created.

Keywords: Color Constancy, Illumination Estimation, Image Segmentation, Deep Learning

Proširenisažetak

Postojanost boja definira se kao svojstvo ljudskog vizualnog sustava koje nam omogućava da vidimo boje objekata neovisno o boji svjetla koje ih osvjetljava. No, glavni mehanizmi koji upravljaju tim svojstvom su i dalje samo djelomično poznati, te se mnogo istraživanja provodi u ovom području. Također, premda ljudski vizualni sistem ponekad ima poteškoća s postojanosti boja, on je sposoban relativno brzo se prilagoditi raznim uvjetima osvjetljenja. Dakle, percepcija boje objekata ostat će konstantna kroz razna osvjetljenja. Tako će, na primjer, banana će izgledati žuto pod sunčevim svijetlom, LED rasvjetom ili fluorescentnim osvjetljenjem.

Računalna postojanost boja se odnosi na podskup problema računalnog vida, kojima je cilj reproducirati svojstvo postojanosti boja ljudskog vizualnog sustava. Potreba za time se javlja jer, za razliku od ljudskog vizualnog sustava, digitalni senzori ne posjeduju to svojstvo inherentno. Time, to svojstvo se mora reproducirati koristeći metode obrade slike. Pojam računalne postojanosti boja se obično sastoji od dva osnovna koraka. To su procjena osvjetljenja te korekcija boje (također poznato u literaturi kao kromatska adaptacija). Od ta dva koraka, prvi korak je složeniji te će na njemu biti naglasak u ovoj disertaciji. Glavni razlog za složenost je taj da je problem poddefiniran. U primjeru s bananom, ako gledamo samo na RGB vrijednosti, nije moguće znati radi li se o žutoj banani na bijelom svijetlu, bijeloj banani osvijetljenoj žutim svjetlom, ili nekoj trećoj kombinaciji koja izgleda žuto. Dakle, prvi korak svakog algoritma postojanosti boje je procijeniti vektor osvjetljenja scene u određenom dijelu slike. Granularnost tih regija može se razlikovati od jednog slikovnog elementa, preko dijela slike (obično kvadratnog oblika), pa sve do cijele slike. Ta granularnost procjene definira vrstu metode za procjenu osvjetljenja koja je potrebna. Metode koje procjenjuju osvjetljenje za kvadratne regije scene ili na razini slikovnog elementa spadaju pod metode za procjenu više od jednog osvjetljenja.

Glavni fokus ove disertacije je procjena osvjetljenja u scenama s više izvora. One se mogu podijeliti na metode za procjenu više osvjetljenja te na metode za segmentaciju slike u ovisnosti na boju osvjetljenja. Metode za segmentaciju su dizajnirane tako da budu kompatibilne s metodama za globalnu procjenu osvjetljenja, koje su mnogo više istraživane nego metode za procjenu više osvjetljenja. S druge strane, metode za procjenu više od jednog osvjetljenja idealno proizvode je razviti set metoda baziranih na dubokom učenju koje su sposobne segmentirati sliku u ovisnosti o osvjetljenju te procjenjivati osvjetljenje scene na razini slikovnog elementa u scenama s više osvjetljenja. Ove metode implementirane su modelima dubokog učenja, poglavito konvolucijskih neuronskih mreža, jer trenutna istraživanja pokazuju prednost takvih metoda nad ostalim metodama obrade slika. Metode su implementirane, trenirane i testirane na novo razvijenom skupu slika, posebno pripremljenom za ovaj problem. Skup slika sadrži 2500 tisuće sirovih slika dobivenih s 5 kamera, scena s osvijetljenih s dva izvora svjetla. Za metode koje rade s varijabilnim brojem osvjetljenja, posebno su izrađene umjetno obojene

slike bazirane na Cube+ bazi podataka [1]. Ta umjetna baza služi kao baza za treniranje te testiranje generalizacijskih mogućnosti predloženih metoda jer se metode trenirane na njoj ispituju na drugim skupovima podataka koji sadrže samo prirodne slike. Cilj tog ispitivanja bio je otkriti koje metode dobro generaliziraju procjenu i segmentaciju osvjetljenja neovisno o broju izvora svijetla.

Doprinosi ove disertacije su:

- Metoda za segmentaciju osvjetljenja za scene s dva osvjetljenja s najviše jednim poznatim osvjetljenjem bazirana na konvolucijskim i transformerskim mrežama
- Metoda za usporednu procjenu osvjetljenja na razini slikovnog elementa te segmentaciju u slikama s jednim ili više osvjetljenja koristeći duboku autoenkodersku mrežu

U ovom radu prikazane su metoda za segmentaciju scena u ovisnosti o osvjetljenju, metoda za procjenu dva osvjetljenja, te metoda za procjenu nepoznatog broja osvjetljenja. Također, prikazana je i metoda za detekciju pogrešne korekcije, koja proizlazi iz korištenja globalne korekcije. Disertacija je podjeljenija u 5 glavnih dijelova. U prvom dijelu su prikazani uvodni podaci potrebni za razumijevanje problematike te metoda koje su prikazane. Zatim, u drugom dijelu opisana je metoda za detekciju pogrešne korekcije. U trećem dijelu prikazana je metoda za segmentaciju scene u ovisnosti o poznatim osvjetljenjima. Nakon toga, u četvrtom dijelu prikazana je metoda za procjenu nepoznatog broja osvjetljenja trenirana na autoenkoderskoj mreži, dok je u petom dijelu opisana metoda za estimaciju osvjetljenja u scenama gdje se nalaze dva izvora. Ove metode predstavljaju razvoj od metoda za procjenu jednog osvjetljenja do metoda koje mogu predvidjeti te segmentirati osvjetljenja u kompleksnim scenama s više osvjetljenja.

U prvom dijelu disertacije opisana je metoda za segmentaciju dijelova scene u kojima je procjena, te samim time i korekcija, bila pogrešna. Ova metoda zasnovana je na transformerskoj arhitekturi. Originalna sirova slika korigirana je globalnom procjenom, te je zatim podjeljenija je na preklapajuće kvadratne regije koje čine jedno susjedstvo. Te kvadratne regije su zatim opet podijeljene u još manje, nepreklapajuće kvadratne regije. Svaka regija unutar susjedstva kodirana je korištenjem matrice kodiranja, te su ti kodovi predani u transformerski model. Zatim transformerski model, korištenjem samo mehanizma pažnje, za svaku malu regiju predvidi radi li se o ispravno korigiranoj regiji ili regija nije dobro korigirana. Finalna segmentacijska maska se zatim dobije spajanjem svih predviđenih regija nazad u sliku. Eksperimenti koji su provedeni s ovom metodom pokazuju da ona postiže odlične rezultate na stvarnim slikama iz predložene baze. Uz odlične rezultate, prednost ove metode naspram drugih metoda dubokog učenja je relativno mali broj parametara, što pozitivno utječe na brzinu izvođenja te memorijsko zauzeće. Nadalje, ova metoda direktno se nadovezuje na metode za procjenu jednog osvjetljenja, jer omogućuje da se detektiraju regije gdje takve metode griješe u scenama s više osvjetljenja. Prednost ovog pristupa je u tome što omogućuje korištenje vrlo dobro istraženih metoda za

procjenu jednog osvjetljenja bez ikakve modifikacije u scenama s više osvjetljenja.

Drugi dio disertacije prikazuje metodu za segmentaciju scena u ovisnosti o osvjetljenju uz pretpostavku da je jedno osvjetljenje scene poznato. Metoda se temelji da ideji da je moguće relativno precizno napraviti relativno točnu globalnu procjenu osvjetljenja čak i u scenama s više osvjetljenja, te time procijeniti dominantno osvjetljenje u sceni. Samim time pretpostavku da je jedno osvjetljenje poznato je moguće zadovoljiti korištenjem neke od dobro istraženih metoda za procjenu jednog osvjetljenja. Tada zadatak metode postaje lokalizirati utjecaj tog poznatog osvjetljenja na određene dijelove scene. Metoda to predviđa na temelju originalne sirove slike te poznatog dominantnog vektora osvjetljenja, korištenjem metoda dubokog učenja. Sama metoda sastoji se od dva dijela. Prvi dio je duboka mreža čiji je cilj napraviti kodiranje originalne slike na razini slikovnih elemenata. Kroz eksperimente se pokazalo da je najbolje koristi kodiranje čiji je cilj proizvesti procjenu osvjetljenja na razini slikovnog elementa. Drugi dio metode je još jedna duboka mreža koja na ulazu prima originalnu sliku, kodiranu sliku (izlaz prvog dijela), te poznati vektor osvjetljenja. Zadatak ovog dijela je proizvesti binarnu segmentacijsku masku koja pokazuje u kojem dijelu scene je poznato osvjetljenje dominantno. Ovako implementirana metoda mogla bi raditi samo sa scenama s dva izvora osvjetljenja. Kako bi se zaobišlo to ograničenje, predložena je iterativna metoda koja više puta primjenjuje opisani model, te time svaki prolaz segmentira regiju osvijetljenu određenim izvorom svjetla. Nadalje, kako bi se u svakom koraku moglo predvidjeti dominantno osvjetljenje, za globalnu procjenu sve prethodne regije su maskirane te se za procjenu koristi samo regije za koje osvjetljenje još nije poznato. Eksperimentalni rezultati pokazuju da ovakva metoda za segmentaciju postiže najbolje rezultate na skupu slika s dva ili više osvjetljenja. Nadalje, rezultati pokazuju da je pretpostavka da je globalna procjena jednako dobra kao i unaprijed poznato osvjetljenje ispravna, jer je preciznost modela jednaka neovisno o izboru.

U trećem dijelu disertacije opisana je metoda koja procjenjuje osvjetljenje svakog slikovnog elementa scene. Ova metoda je također temeljena na dubokom učenju. Problem procjene svakog osvjetljenja za svaki slikovni element je u tome što jedan slikovni element nosi veoma malo informacija o svojoj okolini te samim time i o osvjetljenju. S druge strane, metode koje rade globalnu procjenu, mogu koristiti sve informacije prisutne u sceni kako bi napravile procjenu osvjetljenja. No njihov problem je što mogu predvidjeti samo jedno osvjetljenje po sceni. Opisana metoda bazira se na arhitekturi koja kombinira globalna i lokalna svojstva, kao što su U-Net i mreža s piramidom značajki (FPN). Takve arhitekture omogućuju da se globalne i lokalne informacije uzmu u obzir prilikom procjene za svaki slikovni element. No, korištenje samo takve arhitekture rezultira u nedovoljno točnoj procjeni, zbog poddefiniranosti problema procjene osvjetljenja. Zbog toga, predložena je i posebna metoda treniranja takvih arhitektura koja omogućava točniju procjenu te smanjuje količinu šuma na izlazu modela. Trening se bazira na autoenkoderskim mrežama, gdje je cilj rekonstruirati ulaznu sliku. Prije rekonstrukcije,

metode moraju procijeniti mapu osvjetljenja te korigiranu sliku, te se konačna rekonstrukcija ostvaruje kao umnožak tih izlaza. Ovakva rekonstrukcija tjera mrežu da nauči svojstva objekata neovisno o svojstvima osvjetljenja. Nadalje, treniranje se odvija tako da se sva tri izlaza (mapa osvjetljenja, korigirana slika i rekonstrukcija), predaju složenoj funkciji gubitka. Ona se sastoji od tri kvadratna gubitka za svaka od tri izlaza, te regularizacijskog člana koji je zadužen za smanjivanje lokalne varijabilnosti mape procjene. Regularizacija se koristi zato što je obično promjena osvjetljenja unutar scene spora, dok promjena objekata u sceni nije. Na kraju, za korištenje modela, zadržava se samo procjena osvjetljenja na razini slikovnog elementa, dok se korigirana slika i rekonstrukcija odbacuju. Eksperimentalni rezultati pokazuju da ovakva vrsta treninga poboljšava rezultate procjene osvjetljenja na razini slikovnog elementa. Također, prednost ovakvog modela i treninga je taj da je model neovisan o broju osvjetljenja u sceni.

U konačnici, četvrti dio disertacije posvećen je metodi za procjenu osvjetljenja u scenama gdje postoje samo dva izvora osvjetljenja. Glavni razlog za razvoj specijalne metoda s dva izvora je taj da su prirodne scene najčešće osvjetljenje s maksimalno dva izvora. Na primjer, vanjske scene su osvjetljenje sunčevim svijetlom te nebom, dok su scene u zatvorenom prostoru osvjetljenje umjetnim izvorom te dnevnim svijetlom koje dolazi kroz prozor. Ova metoda bazirana je na metodi predstavljenoj u drugom dijelu disertacije, odnosno na metodi za segmentaciju s poznatim izvorom. Ideja metode je da se kombiniraju dobro istražene metode za procjenu jednog osvjetljenja, koje je moguće trenirati na puno većim skupovima nego one za više osvjetljenja, s metodom za segmentaciju osvjetljenja. Tijek metode je takav da prvo jedna metoda za procjenu osvjetljenja procjeni globalno osvjetljenje u sceni Tako metoda za segmentaciju slična onoj opisanoj u drugom dijelu podjeli ulaznu sliku u regije gdje je samo jedno svijetlo dominantno. Zatim se originalna slika maskira tako da na njoj ostanu samo regije osvjetljenje jedim svijetlom, te se maskirane slike predaju metodama za procjenu jednog osvjetljenja. Konačna procjena na razini slikovnog elementa za cijelu scenu dobije se linearnom kombinacijom osvjetljenja pomoću segmentacijske maske. Maska određuje koeficijente za linearnu kombinaciju za svaki slikovni element. Duboke mreže koje su glavne sastavnice ove metode prvo se treniraju posebno, svaka za svoj problem, te se konačno mreže spajaju te se cijela metoda još trenira zajedno. Rezultati pokazuju da ovakvo treniranje poboljšava ukupnu točnost mreže naspram metoda koje su samo trenirane zasebno. Nadalje, rezultati pokazuju da ovakva metoda postiže najbolje rezultate na skupu s dva izvora osvjetljenja, te da postiže usporedive rezultate na slikama sa samo jednim izvorom osvjetljenja.

U sklopu ove doktorske disertacije razvijene su četiri metode koje omogućuju adaptiranje jednostavnijih i bolje istraženih metoda za globalnu procjenu jednog osvjetljenja na scene s više izvora osvjetljenja. Predložene metode za segmentaciju osvjetljenja omogućuju direktnu primjenu metoda za procjenu jednog osvjetljenja na regije scene koje sadrže samo jedno osvjetljenje te se time poboljšava kvaliteta reprodukcije slike. S druge strane, metode za procjenu

omogućavaju direktnu korekciju slike s procijenjenom maskom osvjetljenja te one pokazuju odlična svojstva generalizacije na scenama s drugačijim brojem izvora osvjetljenja. Nadalje, eksperimentalni rezultati za svaku od metoda pokazuju da postižu odlične rezultate na svojem problemu. Vizualna usporedba rezultata također pokazuje da su dobivene slike realna reprodukcija stvarnih uvjeta.

Ova doktorska disertacija sastoji se od četiri radova objavljenih u časopisima velikog faktora odjeka te međunarodnim konferencijama. Priloženi radovi predstavljaju izvorni znanstveni doprinos disertacije. Na početku disertacije nalaze se pregled metodologije procjene i segmentacije osvjetljenja i pregled postojeće literature. Nakon toga je predstavljen izvorni doprinos popraćen znanstvenim radovima.

Ključne riječi: Postojanost boja, Procjena Osvjetljenja, Segmentacija Osvjetljenja, Duboko Učenje

Contents

1. Introduction 1
1.1. Computational Color Constancy 1
1.2. Problem Statement 2
1.2.1. Illumination Estimation 2
1.2.2. Segmentation Based on Illumination 3
1.3. Scientific Contributions 3
1.4. Thesis Structure 3

2. Overview 4
2.1. Color Constancy 4
2.1.1. Image Creation and Color Spaces 6
2.1.2. Chromatic Adaptation 8
2.1.3. Illumination Estimation 9
2.2. Deep Learning 10
2.2.1. Artificial Neuron 11
2.2.2. Fully Connected Networks 11
2.2.3. Convolutional Networks 13
2.2.4. Transformer Networks 15
2.2.5. Deep Learning Training 16
2.2.6. Deep Learning Application to Computer Vision 18

3. Related Work 20
3.1. Single Illuminant Methods 20
3.1.1. Statistics-Based Methods 20
3.1.2. Physics-Based Methods 21
3.1.3. Learning-Based Methods 22
3.2. Multi-Illuminant Methods 24
3.3. Datasets 25
3.4. Performance Metrics 27

4. Multi-illuminant Dataset 30
4.1. Two Illuminants Dataset 30
4.2. Three Illuminants Artificial Dataset 32

5. The Main Scientific Contributions of the Thesis 34
5.1. Transformer-Based Illumination Estimation Error Detection 34
5.2. Deep Learning Illumination-Based Segmentation 35
5.3. Autoencoder Based Training for Illumination Estimation 35
5.4. Framework for Multi-Illumination Estimation and Segmentation 36

6. Conclusion and Future Work 37
6.1. The Main Conclusions of the Thesis 37
6.2. Future Work 38

7. List of publications 40

8. Author's contribution to the publications 41

Bibliography 43

Publications 51
Pub 1: Illuminant estimation error detection for outdoor scenes using transformers . . 52
Pub 2: Illuminant segmentation for multi-illuminant scenes using latent illumination encoding 59
Pub 3: Autoencoder-based training for multi-illuminant color constancy 77
Pub 4: Framework for Illumination Estimation and Segmentation in Multi-Illuminant Scenes 87

Biography 98

Životopis 100

Chapter 1

Introduction

Color constancy is the property of the human visual system that allows humans to see the color of objects independent of the color of the illumination that is illuminating them. On the other hand, computational color constancy refers to the image processing problem, where the goal is to remove the influence of the illumination from images. This property is important for image processing in digital cameras, as correct illumination estimation and removal produces realistic reproductions of scenes. Furthermore, it has been shown that incorrect color constancy can result in lower performance of down-stream image processing and analysis tasks [2]. For this reason, research and development of deep learning-based models for color constancy in various conditions is the topic of this thesis. In this chapter, a brief overview of color constancy is given, followed by the description of the problem of illumination estimation and segmentation. Thereafter, the main scientific contributions are listed, and finally, the description of the thesis structure is given.

1.1 Computational Color Constancy

As was stated earlier, computational color constancy is the process of estimating the illumination of the scene and correcting for it to produce an image that looks as it was taken under white light. Modern digital cameras have this ability integrated into it in the form of automatic white balancing. This property is usually based on some predefined illumination categories, such as *Daylight, Shade, Tungsten, Cloudy, etc.* However, if the image is taken under more challenging conditions, such as some uncommon illuminants or multiple illuminants per scene, these simpler algorithms will fail and produce incorrectly white balanced images. Incorrectly white balanced images are those where white surfaces are not reproduced as white, i.e., $p_c = [R,G,B]$, where $R \neq G \neq B$. Because of this reason, a need for more complex methods for illumination estimation is required. However, this problem is ill posed, as many combinations of illumination and surface color can produce the same R G B response on the camera sensor. So, to solve

it, additional assumptions about the scene or the illumination have to be introduced. This can be done either by manually using some assumptions, or can be learned using some learning-based methods. In this thesis, the focus will be on using deep learning-based methods to create methods that are capable of detecting regions in images illuminated by one illuminant, as well as methods that are capable of producing a per-pixel map of illumination values for the whole scene.

1.2 Problem Statement

The main problem behind this thesis is illumination estimation and segmentation in realworld scenes. These scenes are often illuminated by more than one illuminant. In such cases, multi-illuminant estimation methods are needed. Furthermore, illumination localization is as important as estimation in those scenes. Thus, a method that can estimate per-pixel illumination vectors in scenes with multiple sources of illumination is needed.

The goal of this thesis is to create a multi-illumination estimation and segmentation method that can work with realworld scenes. Furthermore, it can be shown that many realworld scenes have at most two sources of illumination. In addition to that, the only large enough dataset with annotated realworld multi-illuminant scenes contains only scenes with two sources of illumination. Thus, the proposed methods focus mostly on this case, even though some of them can work on scenes with an unknown number of illuminants.

1.2.1 Illumination Estimation

Illumination estimation is the central problem of any computational color constancy algorithm. Its goal is to detect the value of the illumination for some part of the scene. These methods can be separated by the granularity of the estimation. Global methods produce only one illumination vector for the whole scene. On the other hand, local estimation methods produce estimation vectors for different areas in the scene. These areas can vary in size, from pixel perfect estimations, to estimations for larger patches or regions in the image. Furthermore, this means that if the illumination of the scene is constant, global illumination methods are enough. However, in numerous realworld scenes, this is not true, with a common example being outdoor scenes illuminated by sunlight and skylight. This means that local illumination estimation methods (often referred to as multi-illumination estimation methods) are necessary in most scenes to create accurately white balanced images. For this reason, this will be the central topic explored by this thesis. However, these local estimation methods can benefit greatly if the influence of the illuminants can be localized. For this, segmentation methods based on illumination can be used.

1.2.2 Segmentation Based on Illumination

The second problem explored in this thesis is the problem of scene segmentation based on illumination. The goal of these methods is to detect areas in the input scene that are illuminated by one source of illumination. This allows for the use of global illumination estimation methods on those regions. The use of global estimation is beneficial as those methods are usually simpler that produce only one illumination estimation for the input. This way, multi-illuminant estimation can be separated into two simpler steps, rather than having to directly solve the more complex problem of producing the per-pixel estimation map for the input scene.

1.3 Scientific Contributions

This thesis presents two main contributions. First, a novel method for illumination segmentation in scenes where one illuminant is known is presented. Then, that segmentation method is incorporated into a framework that is capable of accurately predicting the per-pixel illumination for realworld scenes. This is achieved by combining the segmentation model with well researched single-illumination estimation models. Furthermore, it is shown that this type of framework can be trained end-to-end, and that it then exceeds the performance of individually trained models.

The second contribution is a method capable of direct estimation of the illumination. This method is completely agnostic to the number of illuminants. The method is trained using a novel autoencoder procedure, using a novel triplet loss. This type of training is designed to make the model learn the best assumptions about the reflectance properties of common realworld scenes.

Finally, methods are compared to other state-of-the-art models, in both segmentation and estimation tasks. A statistical analysis is conducted and conclusions are drawn about the performances and tradeoff of the proposed models.

1.4 Thesis Structure

This thesis is structured as followed. First, an overview of theoretical background about image formation, color, color constancy and deep learning is given. In Chapter 3, an overview of current research into illumination estimation for single and multi illuminant scenes is given, as well as research about image segmentation based on illumination. Then, in Chapter 4, two new multi-illuminant datasets used in this thesis are described. The main scientific contributions are presented in chapter 5, while Chapter 6 presents the conclusion of the thesis. Chapter 7 presents the list of publications that describe in full the main scientific contributions. Finally, a summary of the author's contributions in the included publications is given in Chapter 8.

Chapter 2

Overview

2.1 Color Constancy

The sense of sight is one of the most important senses possessed by human beings. It allows us to perceive things in our surroundings, navigate around, avoid potential dangers, recognize objects and people etc. Furthermore, its mechanisms are not yet fully understood. One such mechanism is the ability of the Human Visual System (HVS) to adapt to the color of illumination. This allows humans to perceive the colors of objects independent of the colors of the illumination that is illumination in many natural scenarios. This property is called color constancy. Even though its mechanisms and scope are still not fully understood, it has long been a highly studied topics. The first modern experiments date back at least 200 hundred years, with [3, 4, 5, 6, 7, 8] being just a few examples of such experiments. More recently, numerous experiments testing various aspects of color constancy in humans have been conducted. They usually focus on six main questions. *How is color constancy physically possible? What do observers judge? What experimental methods are suitable? What physical scene properties are relevant? What neural mechanisms support color constancy? Are natural scenes and surfaces special?* This shows the importance of color constancy and the research done in trying to understand its underlying mechanisms. However, color constancy is not only related to the HVS. It is also closely related to digital cameras and the field of image processing. There, the goal is to mimic the human ability of color constancy to produce images that look natural.

However, to understand aspects of color constancy, it is important to first understand the spectral properties of light and creation of perceived images. Any scene can be understood as a collection of materials, each with their on spectral reflectivity function R, illuminated by some incoming light with the spectral function I. The function R defines the amount of incoming light that will be reflected by the material for each wavelength. Spectral function I defines the power for each wavelength. The product of those two functions defines the spectral characteristics of a scene. This energy in turn excites the receptors in the human eyes and produces a reaction that

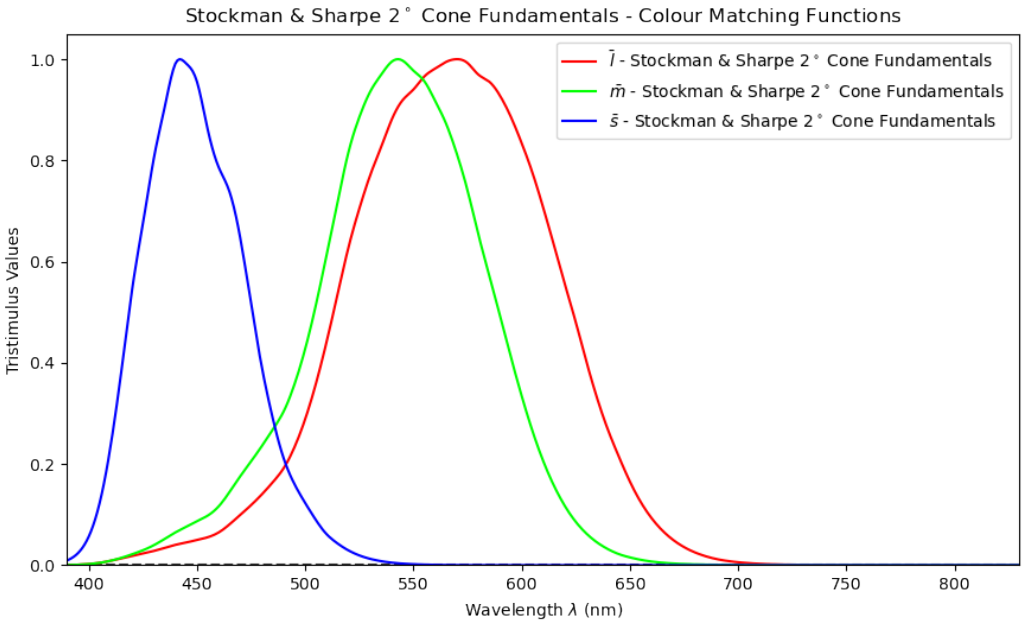


Figure 2.1: Spectral response of the human cone cells defining the LMS color space.

is translated in the visual cortex of the human brain, producing the image we see. This can be described using the following mathematical equation:

$$p_c(x,y) = m_b(x,y)\int_{\omega} I(x,y,\lambda)R(x,y,\lambda)S_c(\lambda)d\lambda + m_c(x,y)\int_{\omega} I(x,y,\lambda)S_c(\lambda)d\lambda\,, \tag{2.1}$$

where ω is the visible spectrum, p_c is the final response per channel c, and m_b and m_c are scale factors. These factors model the amount of body and specular reflectance of the light reflected from coordinates (x,y), thus accounting for different types of surfaces. Furthermore, S_c describes the spectral sensitivity of the sensor that takes in the light, for each of the channels c it contains. In the case of the HVS, the sensors are the cone cells located on the retina of the human eye. Research [9] has proved that they can further be separated into three types, S-, M-, and L-cones, responsible for the short, medium, and long wavelengths respectively. Thus, they represent the different three channels c in the trichromatic human visual system. These spectral characteristics are shown in Figure 2.1. This equation can be simplified by introducing the *Lambertian assumption*, i.e., that the specular reflectance can be ignored. With this assumption, the model for each part of the scene is as follows:

$$p_c(x,y) = m(x,y)\int_{\omega} I(x,y,\lambda)R(x,y,\lambda)S_c(\lambda)d\lambda\,, \tag{2.2}$$

where $m(x,y)$ is Lambertian shading [10].

However, as we can see from both Equations (2.1) and (2.2), the excitation of the sensor is related to the product of spectral functions I and R. This poses a problem, as it is impossible

to know exactly which combination of those functions produced the observed function at the sensor. Furthermore, since a whole continuous spectral functions is compressed down into three discrete responses, different functions will be mapped to the same three responses. This is known as a metameric match. It indicates that the task of color constancy is ill-posed. These issues show that exact color constancy is a very difficult problem. Furthermore, there exists some research showing that color constancy does not work well under highly chromatic light scenarios [11, 12, 13, 14]. Nevertheless, a large corpus of research still indicates that under normal conditions, the HVS is capable of discerning the colors of objects invariant to the color of illumination. Thus, it remains a highly studied topic in both perceptional and behavioral science, with some experiments in this area have been published in [15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29].

On the other hand, the similar problems arise when a scene is captured through a camera. In this case, the sensor that is capturing the light is not the human eye, but the camera sensor. While it does have different spectral characteristics than the human eye, the Equations (2.1) and (2.2) are still applicable. This means that all the previously mentioned problems still hold true. However, unlike the HVS, digital cameras have to rely on image processing to determine the color of illumination. This type of color constancy is known as computational color constancy. It can be best described in two steps. The first step is the estimation of illumination. Second is color correction (also commonly referred to as white balancing). The goal of computational color constancy is to create an image that looks like it was taken under white (canonical) light, thus eliminating the influence of illumination. This is the underlying problem that will be addressed in this dissertation. A broader overview of the problem definition, performance metrics, currently implemented methods, image datasets and proposed methods will be described in other chapters. However, the following sections will give a general definition of the underlying physical properties of lights and scenes, as well as assumptions that are made on those properties.

2.1.1 Image Creation and Color Spaces

In the last section, the mechanism for creating an image based on the three types of cone cells was described in Equation (2.1). However, the choice of S_c was not strictly defined. The usage of Long, Medium and Short spectral response functions defines the appearance of color in the LMS color space. This is a color space that represents the response of the three types of cones of the human eye. The term color space refers to the mapping of the physically produced colors from mixed lights, pigments, surfaces etc. to a numerical representation of the color sensation that is exciting the receptors in the human eyes. In addition to the LMS color space, many other color spaces have been used to quantify colors.

One well-known space is the CIE XYZ color space. It is often utilized as a standard refer-

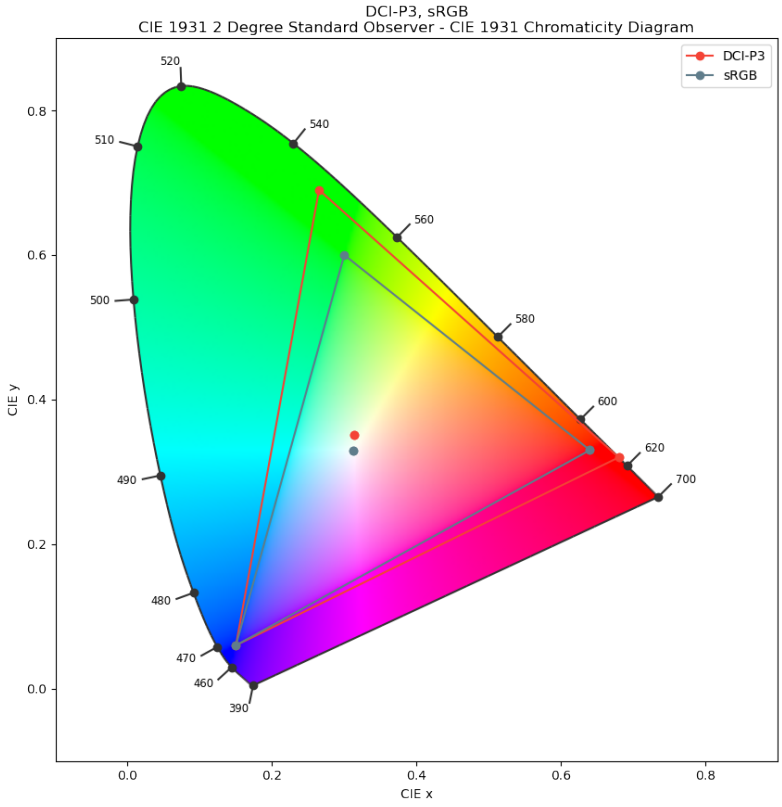


Figure 2.2: Example of two RGB color spaces over the CIE1931 xy plane. The RGB color spaces are additive, and all colors are created by some combination of the three primaries (vertices of the triangles). The xy plane encompasses both of the RGB color spaces.

ence color space. It is device-invariant and encompasses all colors that are visible to a human with average eyesight. Similarly to the LMS color space, it is defined by three color matching functions. Furthermore, it is possible to convert colors from the LMS color space to XYZ space, and back, using only linear matrix transformations. This is true of many other color spaces, such as the CIE RGB, sRGB, LUV, Lab and many others. A prominent example of such conversion is shown in Equation (2.3), which defines the Hunt-Pointer-Estevez matrix [30] for converting from LMS to XYZ color space.

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 1.91020 & -1.11212 & 0.20191 \\ 0.37095 & 0.62905 & 0 \\ 0 & 0 & 1.00000 \end{bmatrix} \begin{bmatrix} L \\ M \\ S \end{bmatrix} \tag{2.3}$$

In some cases, the resulting conversion will not produce viable colors, as different color spaces can represent only certain colors. This can best be seen in Figure 2.2.

More recently, with the advent of the internet and the prevailing usage of computers, there

was a need for a standardized color space for displaying images. This was done by creating the sRGB color space. It defines the standard color space for the internet. Moreover, it is possible to convert sRGB space to a XYZ space using linear transformation. This allows us to display all of our images in this space for easier visualization of the results.

2.1.2 Chromatic Adaptation

Chromatic adaptation is the ability of the HVS to adapt to the changing color of illumination. It allows us to retain the appearance of the colors of objects invariant to the changes in the illumination. This means that to us, a yellow banana will look yellow under both sunlight and a fluorescent office light. However, as stated earlier, this mechanism is not fully understood. For digital cameras, which have to produce images similar to those produced by human eyes, this is a hard problem. There, the process of removing the influence of illumination is sometimes referred to as white balancing. To achieve this, an often used method in digital cameras is the von Kries [21] coefficient rule. It assumes that color responses of each cone can be individually manipulated. This means that a diagonal matrix can be used to adapt the tristimulus values of each pixel. Von Kries transformation is described in the following equation:

$$\begin{bmatrix} c_L \\ c_M \\ c_S \end{bmatrix} = \begin{bmatrix} d_L & 0 & 0 \\ 0 & d_M & 0 \\ 0 & 0 & d_S \end{bmatrix} \begin{bmatrix} u_L \\ u_M \\ u_S \end{bmatrix}, \tag{2.4}$$

where d_L, d_M, d_L are the coefficients for the transformation from one adaptation state to another. They are calculated as a ratio between the LMS values of the illumination in the second adaptation state and the first adaptation state. This means that the pixel values recorded by the camera sensor have to first be transferred to the LMS space, usually through the XYZ space. However, this usually does not present a problem as all of those operations are linear and reversible. Thus, this can be expressed as just a series of matrix multiplications. The most common type of transformation used in digital cameras is the adaptation to the canonical illuminant. This means that the goal is to convert the image to look as though it were taken under white light. It is done by setting the second adaptation state to white light and the first to the original illuminant. However, this presents a problem. The value of the original illuminant is often unknown. That is why this is only one step in performing the task of computational color constancy.

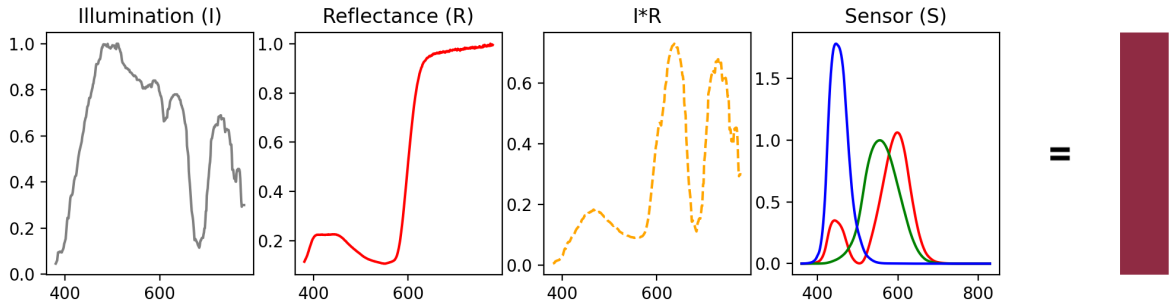


Figure 2.3: Creation of the color of a pixel as seen by a camera sensor.

2.1.3 Illumination Estimation

When a scene is captured by a camera sensor, the value of each pixel is equal to the amount of electric current produced by a sensor element. Then this current is converted to a value for either red, green, or blue component of a pixel. However, this RGB value is a product of all the factors described in Equation (2.1). Figure 2.3 shows the whole process of how a pixel value is created. This means that if the illumination I and reflectivity R are both unknown, then there is not enough information from the RGB values to reconstruct either of those functions. That is why the problem of illumination estimation is ill-posed.

One way to solve this problem is to incorporate some surface, whose reflectivity spectral distribution R is known, into the scene. While this is done in professional settings, where the exact illumination value is demanded, it is not a generally acceptable solution for most scenarios. However, if some assumptions are made, it is possible to estimate the illumination value approximately. This is usually done by assuming some property of the whole scene or of some objects in the scene. Initially, those assumptions were handcrafted, but recently, more and more methods use some sort of learning-based methods to learn the best set of assumptions. One common way to do this is by using deep neural networks, that are described in detail in Section 2.2 Furthermore, there is no one assumption that will work in all scenarios, which gave rise to many methods for the task of illuminant estimation. Another common assumption made was to assume the presence of only one illumination in the scene. These types of methods are called global illumination estimation methods. However, this assumption can be broken in many realworld scenes, as can be seen in Figure 2.4. Common examples of such scenes are an outdoor scene with sunlight and skylight illuminating different parts of the scene, or an indoor scene with artificial illumination and sunlight coming in through a window. A more detailed overview of current methods for illumination estimation is presented in Chapter 3

Once the illumination vector has been estimated, the process of white balancing can take place as described in Section 2.1.2. These two steps are necessary for any computational color constancy method. However, an error in the illumination estimation part will have a negative effect on the white balanced image. This creates two main problems. First, the image will look

Figure 2.4: An outdoor daytime, nighttime, and indoor realworld scene where the single illuminant assumption is broken.

unnatural, which is a problem for camera manufacturers and users. Second, it has been shown that incorrect white balancing of an image can produce errors in downstream image processing and analysis tasks [2]. That is why it is important to estimate the illumination as accurately as possible for each part of the scene.

2.2 Deep Learning

This section summarizes the main ideas and techniques used in deep learning. It also shows some of the models that are used for image processing, that will be used further on in the dissertation.

Deep learning can be described as a collection of mathematical and computer science tools and techniques, which are used for solving complex computer science problems. The main idea is that some unknown function that maps some input to an output, can be approximated using a learning-based model, given enough input and output pairs. Such an approach is called supervised learning. The idea behind learning-based algorithms has been proposed a long time ago, with the main drive being the simulation of connected neurons in the human brain. This gave rise to the perceptron [31], a machine learning algorithm for binary classification. With further combination of multiple perceptrons, fully connected networks were created. And as the computational power of computers increased, more and more layers were added to these networks, which allowed them to represent functions of greater and greater complexity.

However, these types of fully connected networks did little to include any sort of induction bias about the data they were representing. They proved to be less effective at specific applications, such as time series or image processing. For these tasks, specialized networks such as recurrent networks, transformers, and convolutional networks were created.

In the following sections, the underlying mechanisms of deep learning and training of such algorithms will be described. Moreover, a broader description of fully connected, convolutional and transformer networks will be provided. Finally, common tasks set before deep learning models, that these networks perform, will be presented.

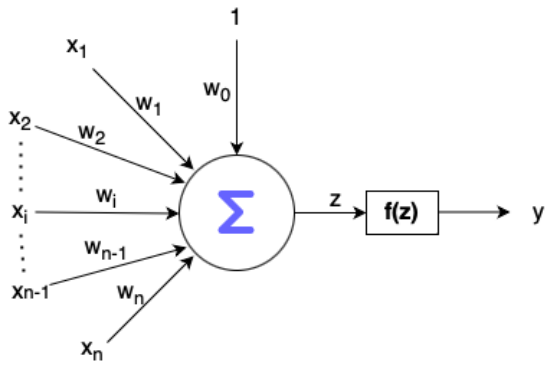


Figure 2.5: A schema of an artificial neuron. f represents the activation function, x are the inputs, w are the weights and y is the nonlinearity applied to the weighted sum z.

2.2.1 Artificial Neuron

An artificial neuron is a main building block of all modern artificial neural networks. It is modeled to mimic a real neuron that is found in the brain. A simplification of a biological neuron is composed of a body, which has small tendrils known as dendrites, and one long tail known as an axon. Dendrites are used to connect to axons of other neurons, and they conduct electric current from the previous neurons. When a sufficient charge is reached, the neuron is activated and discharges thorough its axon, which can in turn be connected to other neurons.

The idea behind an artificial neuron is similar. It has input connections, and one output. This output is passed through an activation function f. Furthermore, each of the inputs contains a weight that multiplies the value coming in at that input. The output is computed as a sum of all the weighted inputs. A schema of the artificial neuron can be seen in Figure 2.5. Thus, the operation of one neuron is described in Equation 2.5.

$$y = f(\mathbf{x}^{\mathrm{T}}\mathbf{w} + w_0)\,, \tag{2.5}$$

where y is the output, x is the input, w are the weights, f is the activation function, and w_0 is the bias of the neuron. Equation 2.5 shows that a neuron is fairly simple, and it does nothing more than a dot product of the input and the weights, and the application of the activation function.

2.2.2 Fully Connected Networks

Fully connected networks, which can also be called multi layer perceptrons (MLP), are some of the first invented deep neural networks. They consist of an input layer, hidden layers, and an output layer. Each hidden layer can be described as performing some function f_h, and the goal of the network is to approximate some function f. Thus, the whole network is performing a composition of all the functions of hidden layers. Finally, then the approximation of the function f can be described as $f^* = f_o(f_h(f_{h-1}\dots f_1(f_i(x))))$. Each of these hidden layers is

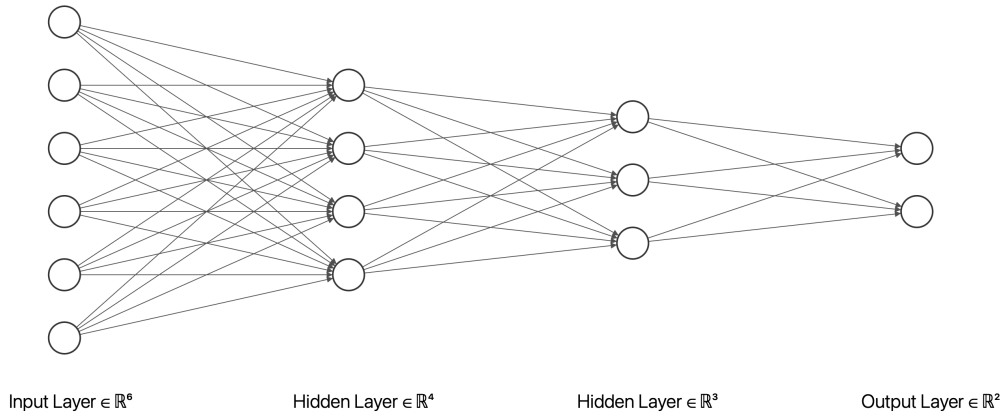


Figure 2.6: Schema of a fully connected feedforward neural network with two hidden layers and an output layer with two components.

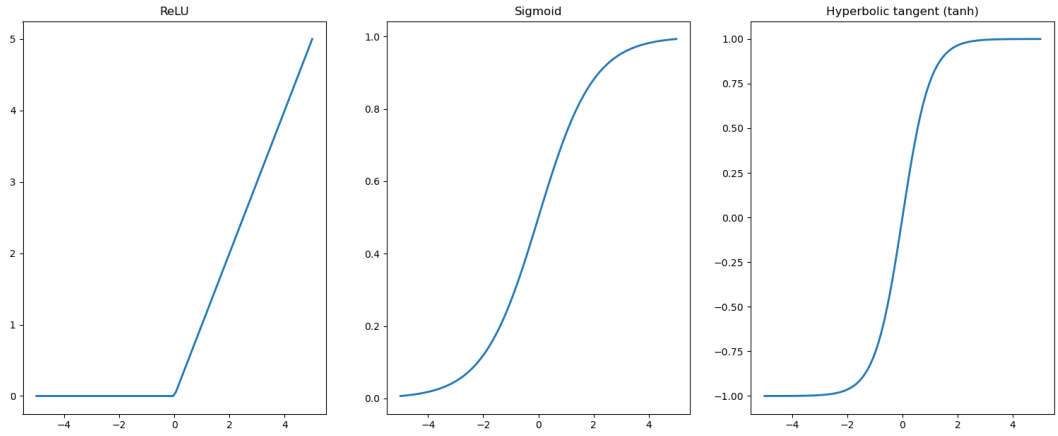


Figure 2.7: Three different nonlinearity functions commonly used in deep learning.

composed of neurons, which are connected to all the neurons in the previous layer, and their output is passed to all neurons in the next layer. This is where the name fully connected comes from. The output of each neuron is passed through some nonlinear function. A schema of a fully connected network is shown in Figure 2.6.

An important factor in any deep neural network is the activation function of the neurons. If the activation function is linear, the whole deep network can be viewed as just having one layer. This comes from the fact that each neuron then does just a linear combination of the inputs, and a linear combination of linear combinations is again a linear combination. To fix this, a nonlinear function must be used as the activation of the neurons in the hidden layers. There exists a wide variety of nonlinear functions that can be chosen. However, to assure adequate performance, these functions need to have a few key features. One of the most important feature is that they should be easy to compute. Furthermore, the same should hold true for their gradients. Some common nonlinear functions used are the rectified linear unit (ReLU), sigmoid, tanh, *etc.* Graphs of these functions are shown in Figure 2.7.

2.2.3 Convolutional Networks

The previous section described one of the fundamental type of network, the fully connected network (FC). They treat every input as separate. However, that is why they are not ideally suited for image processing.

Consider an example of an image containing a dog. For a fully connected network, each pixel is considered a separate input with its weights, and different contributions to the output. This means that there is a difference to the network if the dog is located in the upper right or the lower left part of the image. However, if the goal is just to tell whether this is an image of a dog, its position should not play any part in that decision. Thus, fully connected neural networks are not invariant to translation. To solve this issue, a strong prior about translation should be introduced. This is how convolutional neural networks (CNN) [32] were created.

Convolutional neural networks are a specialized type of network that work with a structured grid of data. Such grids can either be 1-D (time series), 2-D (images) or even 3-D (CT scans). The main idea behind CNNs is the use of a small kernel sliding over input to create a translation equivariant feature map. The kernel contains a set of learnable weights and performs the convolution* operation with the elements of the structured grid. The convolution operation as it is used in CNNs for image processing can be described as:

$$S(i,j) = (K * I)(i,j) = \sum_m \sum_n I(i+m, j+n) K(m,n)\,, \tag{2.6}$$

where K is the kernel with learnable parameters, I is the 2-D grid of pixels, m,n are the positions in the kernel, and i,j are positions in the image. There are three main benefits of using convolution: sparse interactions, parameter sharing and equivariant representations.

Sparse interactions feature means that not every input unit has an interaction with the output unit, unlike in FC networks. This is important as the image can have several millions of pixels, but important features can be discovered using kernels of much smaller size. This can drastically improve computational time. However, this does not mean necessarily mean that the kernel interacts only with this small number of parameters. By adding more layers to the CNN, the effective receptive field of a kernel increases, as each deeper layer processes the information from higher layers, which contain information about greater are in the original image.

The second feature of CNNs is parameter sharing. It refers to the usage of the same parameters for more one computation in the model. In the case of FC networks, each parameter of the weight matrix is used only once. However, in CNNs each kernel is moving across the image, and for each position i,j it performs the convolution operation (Equation (2.6)). This, combined with sparse interactions, decreases the number of parameters of CNN drastically compared to

*In fact, it performs the cross-correlation operation, but since the parameters are learned, their place is not relevant, as long as it is consistent.

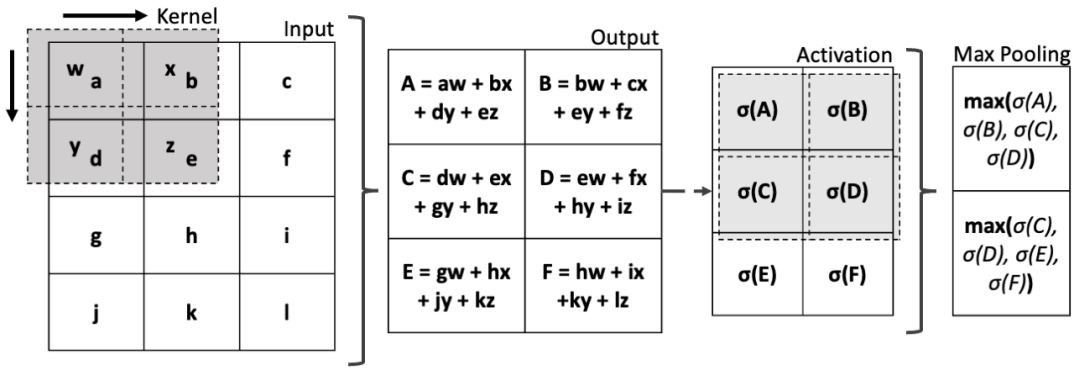


Figure 2.8: Example of a classic 2-D convolutional layer with convolutions, nonlinear activation and a max pooling layer. Both kernels shown are of size 2×2 and with stride 1×1.

FC networks.

Finally, the parameter sharing and sparse interactions give CNNs the additional property of equivariance to translation. Equivariance of a function means that if the input to the function changes in some way, the output changes in the same way. In case of translational equivariance, it means that if the input is translated by n pixels, then the output will also be shifted by n pixels. In the example with the picture of a dog from the beginning of this section, if the dog is moved from top right to bottom left, the output of the CNN that detects the dog will move from top right to bottom left as well.

A typical CNN is usually composed of a set of layers stacked on top of one another. A CNN layer is usually composed of three main operations. First, several kernels perform convolution in parallel to create a set of linear activations. Then, those activations are passed through a nonlinear activation function like the ones shown in Figure 2.7. Finally, a pooling function is applied to the nonlinear activations. A pooling function transforms the output at the location i, j with the summary statistics of its neighborhood n, m. One popular type of pooling is max pooling [33], where the output is replaced by the maximum value of its neighbors. Other types include average pooling, L^2 norm of the neighborhood, weighted average pooling etc. The pooling operation is also shown in Figure 2.8. One useful feature of pooling layers is that they reduce the overall dimension of the feature map produced in the first two stages of the layer.

Finally, a typical CNN layers are used as an encoder for the input image. This produces an encoding that can then later be used for different tasks. One common task is the image classification, where this encoding is used to determine what object (from a predetermined set) is present in the image. This can be done using fully connected layers that are added to the end of the convolutional layers. This is a viable strategy as the usage of pooling layers in the CNN reduced the overall dimension of the input, making it feasible to employ the more memory intensive FC layers. Another approach is to utilize fully convolutional networks, which rely solely on convolutions to generate the final output of the model. This approach can be invariant to the size of the image and still has a reduced number of parameters compared to the network

using fully connected layers.

2.2.4 Transformer Networks

Transformer networks [34] are a much newer type of network than CNNs and fully connected networks. They were created for natural language processing tasks. The main idea behind transformers is that a series of inputs can be encoded, and then a self-attention (relevance) score can be assigned between each pair of embedded inputs. Thus, each part of the input data is weight differently. Another benefit of the self-attention approach, is that the model can access the whole state at the same time, which greatly improves the parallelization capabilities of such models when compared to traditional recurrent models. This is essential, for example, in language translation, where there is a need to access the whole context of the sentence.

A transformer is composed of several layers which, all of which contain multi-head self-attention modules (MHSA) followed by layer normalization. Layer normalization is then connected to two fully connected (FC) layers with ReLU activation. MHSA and FC layers are connected through skip connections. The $j^{th}, k \in 1...K$ layer performs operations:

The input P_{j-1} is passed to the MHSA layer which computes the query q_i, key k_i and value v_i matrices. This is done using learnable weights $W_{q,i}, W_{k,i}, W_{v,i}$ for each attention head $i \in [0,m]$.

$$q_i = W_{q,i}P_{j-1}, k_i = W_{k,i}P_{j-1}, v_i = W_{v,i}P_{j-1} \tag{2.7}$$

$$A_i = softmax(q_i k_i^{\mathrm{T}}) \tag{2.8}$$

$$SA_i = A_i V_i \tag{2.9}$$

The self-attention output are of each head i is concatenated. This creates the final self-attention matrix $SA = [SA_0, SA_1, \ldots SA_i, \ldots, SA_m]$. The self-attention is added to the input P_{j-1}

$$X_{j,sa} = P_{j-1} + SA_j \tag{2.10}$$

The output $X_{j,sa}$ of the MHSA layer is then normalized by layer normalization. Then, it is passed through two fully connected layers. The skip connection adds $X_{j,sa}$ to the output of the fully connected layer $X_{j,fc}$ and then it is again normalized through layer normalization.

$$X_{j,fc} = \mathrm{ReLU}(X_{j,sa}W_{j,1} + b_{j,1})W_{j,2} + b_{j,2} \tag{2.11}$$

$$X_k = X_{j,fc} + X_{j,sa} \tag{2.12}$$

This creates an encoding for each of the inputs, and the relevance of each one pertaining to all the others. This encoding can then be used for various purposes. Another benefit of this

approach is that this encoding can be pretrained on a large corpus of data.

As was mentioned before, transformer models were first introduced in the field of NLP. Nevertheless, they have achieved great success in the field of computer vision as well. One well-known example of such a model is the Vision Transformer [35]. It used image patches to create a sequence out of a single image. However, unlike the CNN models described in Section 2.2.3, transformer models do not introduce any hard inductive bias about locality. This can be both helpful in some task, but detrimental in others. Moreover, it has been shown that transformers can learn to focus on local context in images. However, they do require a considerable corpus of data to match the performance of CNNs. This, combined with their size, can prove problematic in some problem where performance is critical and data is not abundant. Nevertheless, they do remain an active area of research for NLP and CV related problems.

2.2.5 Deep Learning Training

All learning-based models contain parameters that have to be learned based on the data that is given to the model. This process is known as training. To train a deep learning model, a few things need to be defined. These include the data, loss function and the optimizer. Finally, an algorithm for training deep networks that is fast and can propagate the error through the whole network is needed.

For supervised deep learning, data comes in the form of pairs of input and expected output, also known as the label. For unsupervised learning, no label is provided. Training of a deep learning model is done in epochs, where one epoch means one pass through the training data. However, since it is impractical in terms of memory to pass all the data through a model at the same time, the data is split into batches. The size of the batches influences the speed of training, with the smaller batches taking less memory, but more time, while it is the opposite for larger batches.

Furthermore, a separate set of labeled data is set aside and is used as validation. A model is considered to generalize the best, when it achieves the minimum error on the validation data. It is important to note that validation data is never used for training. Finally, when the model has achieved the desired performance on the validation data, it is used on a new data that has not been seen anywhere during the training process. This data is called test data. This data is used to evaluate the final performance of the model.

Loss Function

A loss function L defines the error in the output of the model and the real output. If deep learning is looked at as function approximation, then the loss function would measure the difference between the output of the real function and the learned approximation. However, in most cases

the real function is not known, and only a set of labeled data is available. In that case, loss is computed between the output and the label for each set in the training data. A choice of a loss function is important for the task that the neural network is trying to solve.

Another key property that the loss function has to have is a derivative that is different from zero. This gradient will then point in the direction in which the loss function increases. So, to minimize the loss, the parameters of the model should be moved in the opposite direction. The amount of movement applied is calculated using an optimizer.

Optimization

Optimization of the parameters of a deep neural network is done using an optimizer. An optimizer can be considered as a way of calculating the amount that the parameters have to be moved depending on the gradient of the current loss function, and the gradients from the previous steps. Another value important for training is the learning rate. It multiplies the gradient to prevent too fast changes in the parameters of the model.

The parameters are then updated using the computed gradient, learning rate and the backpropagation algorithm. Backpropagation will be explained in the next section. One possible optimizer would be to calculate the gradient of all the outputs and update the parameters using these gradients. This type of optimization is called gradient descent. However, the amount of data can be massive, and in that case, the process would be very slow. That is why a commonly used alternative is the stochastic gradient descent (SGD) [36]. It samples the output data and calculates the average on some sampled set. Then this gradient is used to update the parameters. This way, only a portion of the output data is used randomly.

Stochastic gradient descent does not contain any knowledge about the previous computed gradients. This means that it is relying solely on the learning rate being set properly to prevent the parameters of the network from diverging. However, some other optimizers incorporate the previous gradients into the calculation of the move in each training step. They rely on the rolling averages of gradient or gradient moments. A few notable examples are AdaGrad [37], RMSProp [38], and Adam [39] optimizers.

Backpropagation Algorithm

The backpropagation algorithm is the most used method for training feedforward neural networks like the ones described in this chapter. The main idea of this algorithm is to compute the gradients of the loss function in relation to the weights of networks. This can be done either on a single example, or averaged over multiple examples of the training data. However, unlike direct computation of the gradients for each weight, which would be computationally expensive, backpropagation does this efficiently.

The main part of this algorithm is the efficient passing of the gradients from the output towards the input. Here, the chain rule of calculus plays an important part:

$$\nabla_x z = \left(\frac{\partial y}{\partial x}\right)^{\mathrm{T}} \nabla_y z \tag{2.13}$$

$$\nabla_{\mathbf{X}} z = \sum_j (\nabla_{\mathbf{X}} \mathbf{Y}_j) \frac{\partial z}{\partial \mathbf{Y}_j}, \tag{2.14}$$

where Equation (2.13) is the chain rule for vectors and Equation (2.13) is the chain rule for tensors, and $\mathbf{Y} = g(\mathbf{X})$ and $z = f(\mathbf{Y})$. The chain rule can recursively be applied to components of the network to produce the backpropagation algorithm. Thus, the gradient for any weight w of the network f, with the activation function, σ can be computed in relation to the inputs x as:

$$\delta_j^H = \frac{\partial L}{\partial y_j^H} \sigma'(z_j^H) \tag{2.15}$$

$$\delta_j^h = \sum_k w_{kj}^{h+1} \delta_k^{h+1} \sigma'(z_j^h), \tag{2.16}$$

where h is a hidden layer, H is the output layer, k goes over all the neurons in the layer h, and δ_j^h is the gradient of neuron j in the hidden layer h, in relation to the input. z_j^h is the output of the neuron j in the hidden layer h before the activation function has been applied. y_j^H is the output of the j neuron in the output layer. Finally, these two equations allow us to update the parameters of the network based on the gradient of the loss function.

2.2.6 Deep Learning Application to Computer Vision

The previous sections described various forms of neural network architectures and their training procedures. However, they were not applied to any concrete problems. This section presents some of the most common problems in computer vision that can be solved using deep learning methods. The list is not exhaustive, but presents only the problems related to those that will be explored in this dissertation.

Image Classification

Image classification is one of the most common usage of neural networks in computer vision. The goal of such methods is to take an image as input and classify it into one of the predetermined classes. One of the largest datasets in computer vision is the ImageNet dataset, which is created for the image classification problem. It features more than 14 million images and corresponding labels are separated into more than 20000 classes. Furthermore, this is one of the easier computer vision tasks, as the output for one image is just the class score for each class.

Image Estimation

The goal of image estimation is to estimate some value from the given image. It differs from image classification as there is no predetermined class given to each image. The output of the network is a continuous scale of numbers, and the goal is to estimate the number as accurately as possible Computational color constancy is an example of such a problem. There, an input to the deep learning model is a raw image taken directly from a camera sensor, and the output is the value of the illumination vector. Another example of estimation is estimating of the age of a person based on their dental X-rays.

Object Detection

Object detection is a computer vision problem where some set of object have to be located in the scene. This is usually done using bounding boxes to describe the objects. The output of such networks is usually a set of zero or more bounding boxes and corresponding class labels for each of the detected images in the scene. A common example of object detection would be traffic sign detection done by remotely controlled vehicles.

Image Segmentation

Image segmentation is a task of finding the class label for each pixel in the input image. This means that each pixel in the scene will correspond to one class. This type of task is, similar to object detection, often used in autonomous vehicles. There, the goal is to segment out classes such as roads, traffic signs, pedestrians, other vehicles etc. Furthermore, image segmentation will be used extensively in the scope of this dissertation. Many realworld scenes contain more than one illuminant. Thus, it would be useful to be able to segment a given scene based on the illumination that is illuminating each part.

Chapter 3

Related Work

This chapter describes the state-of-the-art methods used for color constancy. If focuses on two main types of methods, the single-illuminant and multi-illuminant estimation methods. The main difference between those types of methods is that single-illuminant methods assume the existence of only one global illuminant. On the other hand, multi-illuminant methods do not have this assumption. This means that they have to estimate the illumination vector for each point in the input scene. Furthermore, a boundary between the illuminants can be detected. This gives rise to the need of multi-illumination segmentation methods. The following sections describe first the single illuminant methods, followed by multi illuminant methods. Finally, current datasets and performance measures for the task of color constancy are described.

3.1 Single Illuminant Methods

3.1.1 Statistics-Based Methods

Statistics-based methods represent some of the first methods used for color constancy. They use low-level image statistics to impose assumptions on spectral reflectance R. This provides enough definition to make finding the illumination vector solvable. Most of these methods can be described using a single framework [40]. Equation (3.1) describes the framework:

$$\left(\int \left\| \frac{\partial^n f_{c,\sigma}(x)}{\partial x^n} \right\|_p dx \right)^{\frac{1}{p}} = k e_c^{n,p,\sigma}, \tag{3.1}$$

where $\| \cdot \|_p$ is the Minkowski norm of order p, n is the order of the derivative, and x is an image pixel. Derivatives of the image are defined by applying convolution to the original image with Gaussian derivative filters with deviation σ.

One basic method which is encapsulated in this framework is the Max-RGB (White-Patch) method [41]. It assumes that in each image, there is a specular highlight, which is the brightest

part of the image. The highlight is a direct reflection of the illuminant. This assumption can be represented as:

$$\|f_c(x)\|_\infty = e_c\,, \tag{3.2}$$

where the Minkowski norm becomes the max norm when $p \to \infty$. Thus, it is equivalent to Equation (3.1) when $n = 0, \sigma = 0, p \to \infty$.

Another often used statistics-based method is the Gray-World [42] method. It assumes that the average reflectance of any scene is gray. Thus, any deviation from this is caused by the illumination. This assumption can be expressed by the equation:

$$\int f_c(x)dx = k'e_c\,, \tag{3.3}$$

This method is a special case of Equation (3.1) with $n = 0, p = 1, \sigma = 0$. A small variation from this method is the Shades-of-Gray [43] algorithm. It is derived by setting $n = 0, \sigma = 0$ and leaving p as a free parameter in Equation (3.1):

$$\left(\int \|f_c(x)\|_p dx\right)^{\frac{1}{p}} = k'e_c^p. \tag{3.4}$$

Finally, the Gray-Edge algorithm [44] is a generalization of the Gray-World assumption. Where Gray-World assumes that the average reflectance of the whole scene is gray, Gray-Edge assumes only that some derivative of the image is gray. The name Gray-Edge comes from the fact that the first derivative of the image is used for edge detection. This algorithm can be expressed using Equation (3.1) by using $p = 0$, $\sigma = 0$, and by leaving n as a free parameter which defines the order of the derivative:

$$\left(\int \frac{\partial^n f_c(x)}{\partial^n x}dx\right) = ke_c\,, \tag{3.5}$$

where n is usually set to either 1 or 2.

Furthermore, newer statistics-based methods are still being developed. The benefit of such methods is that they are usually faster to compute and invariant to the sensor of the camera that took the image. These two factors make them ideal for implementation in low-power devices, such as digital cameras. However, because of their simplicity, they do not perform adequately in more difficult conditions. In [45] authors show that specific types of scenes present a problem for statistics-based methods. Thus, more complicated methods need to be taken into account.

3.1.2 Physics-Based Methods

One such approach is to try to model the physical properties of illumination and its interaction with objects. These methods tend to be more complex than statistics based methods. They

usually rely on the dichromatic reflectance model, which separates the reflectivity into specular and body reflectances. Here, specular reflections are usually much brighter than their body counterparts and only reflect the pure color of the illuminant (e.g., mirrors). Max-RGB [41] method can be considered as the simplest physics-based method. Some other physics based methods include [46, 47, 48, 49, 50]. However, these methods are usually quite difficult to compute while not providing much improvement in performance compared to their statistics-based counterparts.

3.1.3 Learning-Based Methods

Learning-based color constancy methods can roughly be separated into classical machine learning models and deep learning models. One classical machine learning approach was proposed in [51]. The authors use exemplar learning to learn surfaces in the train set. Surface representation is used to determine N nearest neighbors to each surface. These are used to determine N possible illuminant estimations. The final estimation is obtained using either their mean or median once outliers have been removed.

Other machine learning methods like [52], [53], [54] use linear models like kernel or support vector regression. Methods like [55], [56], [57], [58], [59] model the illumination and reflectance functions as random variables.

Deep learning models for color constancy cover a wide variety of different methods. They have proven to perform much better than statistics-based methods and simpler machine learning methods in many categories. However, they are computationally more complex than statistics-based methods.

One of the first methods that used convolutional deep neural networks for illuminant estimation was proposed in [60]. The authors proposed a small CNN that was composed of only one convolutional layer with 1×1×240 kernels. This was followed by a max pooling layer with size 8×8 and stride 8. Two fully connected layers with ReLU activation followed the max pooling layer, and they produced the three channel illumination vector. The network was trained on image patches to increase the number of training examples. Median pooling was used to create the final estimation from image patches.

The authors proposed a similar approach based on image patches in [61] However, all the patches were given to the network to at the same time. This network was fully convolutional. However, the main contribution was the last layer of the network, which introduced an attention mechanism to combine image patches. Three channels of the output were the illumination vector, while the fourth was the attention map. The attention was used to scale the illumination

estimations for each patch. The scaling was computed using the following formula:

$$p_g = \text{normalize}(\sum_i c_i \hat{p}_i)\,, \tag{3.6}$$

where c_i is the scaling factor. The scale determines how much the estimation of the patch i contributes to global estimation. p_g represents final global estimation, while $\hat{p}_i$ is the illumination estimation of patch i.

The previously shown models were all relatively small compared to many deep learning models used for other computer vision tasks. There was some research that suggested that the increase in depth of the network does not benefit estimation accuracy. However, there exist some color constancy methods that utilize large CNNs. One such approach was presented in [62]. There, the authors use a large VGG16 [63] network iteratively to achieve state-of-the-art performance. Furthermore, in [64] authors propose a very deep architecture which uses residual connections to stabilize training. On the other hand, in [65] propose an ensemble of two networks. One of these networks provides two hypotheses about the illumination of a given patch, and the second one chooses the correct hypothesis. In [66] a novel feature extraction layer is used. This feature extractor disregards the spatial information in the image and focuses solely on chromaticity. Finally, in [67] the authors propose a two stage method. The first stage is to classify the type of illumination present in the scene. Then, the second stage chooses an appropriate model for such classification, and that model produces the illumination estimation.

Another approach to convolutional color constancy was proposed in [68]. Unlike all the previously described deep learning methods, this method discards spatial information. It focuses only on the chromaticity log UV histograms of images. The 2-D log UV histogram H was passed through a convolutional filter $F(u,v)$. This linear activation is passed through the softmax activation function. This is done to filter out noise and learn the distribution of possible illuminations. The output is the $P(u,v)$ filtered histogram. The final estimation was done by simply taking $\text{argmax}_{u,v}P(u,v)$. The log UV color space is then converted back to RGB color space. To train their model, the authors use a novel loss function:

$$L = \sum_{u,v}(P(u,v)C(u,v,L_u,L_v))\,, \tag{3.7}$$

where P is the filtered histogram and C is the angular distance (see Equation (3.8)). In [69], the same idea was extended with mapping more than one illuminant value to the same point in the UV histogram. This was done by wrapping the original histogram around a torus. This reduced the size of the histogram and improved computation time.

3.2 Multi-Illuminant Methods

The multi-illuminant estimation problem has been much less studied than single-illuminant estimation. One reason for this is the lack of a large multi-illuminant dataset. It is difficult to accurately annotate multi-illuminant images. Numerous methods proposed for this problem are learning-based. Nevertheless, it is important to note several statistics-based methods have been proposed in [70], [71], [72], [73]. They usually view segmentation and estimation as separate tasks that are combined later. The first task of such methods is localization. For this, image textures are used in [72]. On the other hand, Kmeans is used in [71]. Then, Max-RGB method is used for estimation. The localization is used to compute the final per-pixel illumination of the scene.

On the other hand, in [74] the authors propose a white-balancing method for scenes in which the total number of illuminants is not known. They achieve this by selecting N white-balance points that are then mapped to ground truth ones. Finally, [75] proposed a method that imitates the properties of the human eye. They use Adaptive Surround Modulation (ASM) capability of the human eye to regulate the receptive field of neurons based on contrast. This method achieves excellent results, especially in single-illuminant estimation conditions.

Some single-illuminant methods described in the previous chapter can be adapted to work with multi-illuminant scenes. These are [60], [65], [51]. Still, they were primarily designed for single illuminant estimation tasks. One machine learning method for multi-illuminant estimation was proposed in [76]. The method uses Conditional Random Fields to model the dependency between illumination estimation for patches of the image.

In [77] the authors proposed a convolutional neural network that was a multi-illuminant upgrade to the method proposed in [60]. The main idea was still to perform illuminant estimation on image patches. However, median pooling is replaced with a more complex Kernel Density Estimation (KDE) method. The benefit of this approach is that can detect the presence of an unknown number of illuminants. Then, the illuminants are grouped, and the final illumination estimation is computed for the whole scene. Another deep learning approach, described in [78], used Generative Adversarial Networks (GANs) to estimate illumination. Here, methods such as Pix2Pix [79] were completely agnostic to the number of illuminants.

In [80], a simple brightness threshold-based method for outdoor images was proposed. Their method had two stages. The first step was to create a segmentation of the image into sunlit and shaded regions. This was done using a brightness threshold. Thereafter, one statistics-based method for single-illuminant estimation (see Section 3.1) was used on the sunlit regions. Final estimation for the shaded regions was done by adapting the ratio of the red and blue channels.

3.3 Datasets

Color constancy dataset can, like the methods, be split into single- and multi-illuminant datasets. There is much more variety of single-illuminant datasets. Moreover, single-illuminant datasets contain vastly more images than the multi-illuminant ones. The biggest reason for such a discrepancy is the fact that multi-illuminant datasets are much harder to create. In addition to capturing the image and the groundtruth using some calibration object, multi-illuminant datasets require spatial distribution masks of the illuminants. These masks can be generated automatically in laboratory conditions. However, for realworld images, these masks have to be manually annotated. This poses a problem for the development of multi-illuminant methods. Most multi-illuminant methods were either trained on image patches, or were trained on artificially generated images.

Single illuminant datasets are much more diverse. Some of the most commonly used datasets are the Gray Ball [81], the ColorChecker [82], NUS8 [83], Cube+ [1], and Intel TAU [84] datasets. All single illuminant color constancy datasets (except the Gray Ball dataset) provide raw, unprocessed linear images. These types of images are used most often in color constancy research.

The ColorChecker uses a Macbeth Color Checker placed in the scene to extract ground truth illumination. It contains 568 images taken with two Canon DSLR cameras. However, it proved problematic as many scenes contained multiple sources of illumination. Later, in [85], it was reprocessed to mitigate some issues. Despite that, it remains an often misused color constancy dataset [86].

A more modern color constancy dataset was introduced in [1]. It contains 1707 indoor and outdoor images. It uses a SpyderCube calibration object to acquire ground truth illumination in the scene. Examples of indoor, outdoor and nighttime images are shown in Figure 3.1. It contains images taken by only one Canon DSLR camera. This poses a problem as all the images are captured using only one sensor, with its sensor function S.

The issue with the small number of sensors was addressed in NUS and Intel TAU datasets. They both contain images taken with multiple camera models and manufacturers. The NUS dataset incorporates images taken by 8 different cameras. In the Intel TAU dataset, three different cameras were used to take the images. These include two DSLR and one Sony mobile phone sensor. An example of the images from this dataset can be seen in Figure 3.2. It is the largest single-illuminant dataset with raw images, containing 7022 images.

As stated earlier, the number of multi-illuminant datasets is much smaller. Moreover, they contain much fewer images. The first multi-illuminant dataset was proposed in [87]. It contains 67 small images (256×384 pixels), divided into two sets. The first set contains laboratory images. The second set contains outdoor images. They are stored in sRGB color space that

Figure 3.1: Three different images from the Cube+ dataset. Outdoor, indoor and nighttime scenes are presented, respectively.

Figure 3.2: Three different images from three different camera sensors from the Intel-Tau dataset.

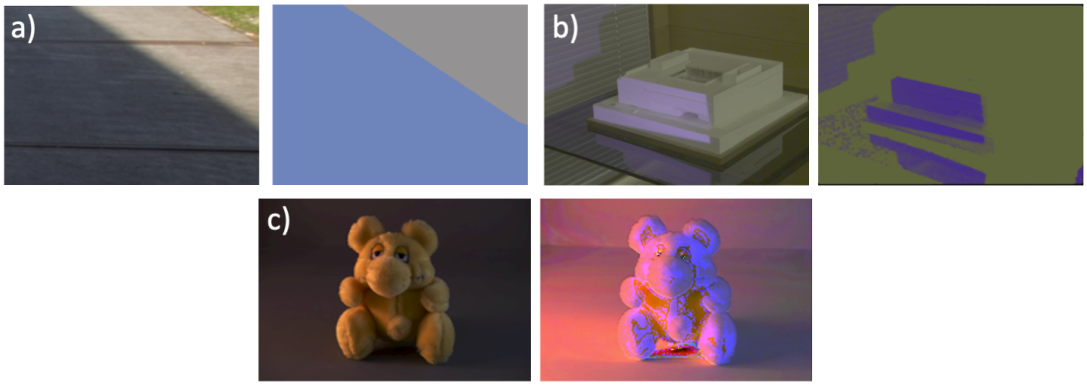


Figure 3.3: Laboratory and realworld images with two sources of illumination and the groundtruth information. a) Dataset proposed in [87] b), c) Dataset proposed in [76]

were only later converted to linear images. This means that these are not real raw images, and are thus not useful for most color constancy research.

Another similar dataset with laboratory and realworld images is the dataset presented in [76]. It contains 60 laboratory images and 20 realworld images. The groundtruth masks were generated automatically. However, this was possible only since both sources of illumination in the scene were artificial. The groundtruth was automatically by alternately switching off light sources. However, this is not possible in realworld images. Examples of images from these two datasets can be found in Figure 3.3.

In [80], the authors mention they use approx. 30 realworld linear images taken by a mobile phone camera. However, those images are not publicly available. The lack of a large unified multi-illuminant dataset poses a problem for the creation of larger learning-based multi illuminant methods.

3.4 Performance Metrics

Evaluation of color constancy methods is closely tied to the accuracy of illumination estimation. For single-illuminant estimation methods, the estimated and groundtruth illumination values are compared. This means that the comparison is done on a single vector. The most widely used metric is angular distance. It is computed as:

$$err_{ang} = \cos^{-1}\left(\frac{e_r \cdot e_p}{e_r e_p}\right), \tag{3.8}$$

where $\cdot$ is the dot product, e_r is the true illuminant and e_p the estimated illuminant.

However, in [88], it is shown that this metric can become unstable. That is why, in [88], the authors proposed a metric called reproduction angular error. It calculates the angular distance of the correction factor from the canonical light $e_c = [1,1,1]^T$. Equation (3.9) shows the way

this metric is calculated:

$$err_{rep} = \cos^{-1}\left(\frac{(e_r/e_p)\cdot e_c}{(e_r/e_p)\sqrt{3}}\right). \tag{3.9}$$

During training of learning-based models, angular error can be unstable because of the use of the inverse cosine function, which tends to infinity when the angular distance approaches π. In [89], the authors propose the loss function described in Equation (3.10):

$$err = 1 - \frac{e_r \cdot e_p}{e_r e_p}, \tag{3.10}$$

which simplifies computation and also makes the derivative stable.

Another loss that was recently proposed in [90]. It is more robust regarding different distributions of illumination. It can be computed as:

$$err_{rob} = \left\| \frac{e_p - e_{gt}}{e_{gt}} \right\|_2, \tag{3.11}$$

where e_p is the estimated illuminant and e_{gt} is the true illuminant.

The above described loss functions and performance metrics were all proposed for single-illuminant estimation. However, they can be used in multi-illuminant with multi-illuminant estimation methods. The most common use of these metrics is to compute the mean value of the metric for all the estimated illuminants. However, it is important to note that then, the illumination maps have to be balanced. Otherwise, one of the illuminants will more greatly influence the final performance metric.

Multi-illuminant segmentation tasks require the definition of a metric related only to segmentation. It does not incorporate any information about the color of the illuminants. For this purpose, Dice coefficient [91] or Intersection over Union (IoU) metrics can be used. Definitions of the Dice coefficient and IoU are shown in Equation (3.12):

$$\begin{aligned} \text{Dice} &= \frac{2|\text{TP}|}{2|\text{TP}| + |\text{FP}| + |\text{FN}|} \\ \text{IoU} &= \frac{A \cap B}{A \cup B}, \end{aligned} \tag{3.12}$$

where TP, FP, FN are true positive, true negative and false negative values when comparing the prediction to the groundtruth. $|\cdot|$ represents the cardinality (number of elements) of the set. A and B are areas in the real and predicted masks that are classified in the same class.

These metrics are reported on the test set, which was not seen during training of learning-based methods. Furthermore, it is important that the test set is the same for all the methods that are compared. Once all the results are reported, a statistical analysis is conducted. For color constancy related task, the most often reported statistics in literature are the mean, median,

trimean, average of the best 25% and the average of the worst 25% samples. In [92], the authors show that the usage of the median statistic for illuminant estimation is encouraged, as the distribution of angular errors is not symmetrical. The worst 25% metric is also important as it quantifies the robustness of the model, since most of the models produce the mean and median values that are lower than the human eye can distinguish [93].

Chapter 4

Multi-illuminant Dataset

This chapter describes the multi-illuminant dataset that was used to train and test models described in this thesis. It was created as a joint effort on development of multi-illuminant estimation and segmentation project at the Image Processing Group of the Faculty of Electrical Engineering and Computing. It contains 2500 images of real-world scenes with two sources of illumination. The next section describes the dataset in detail.

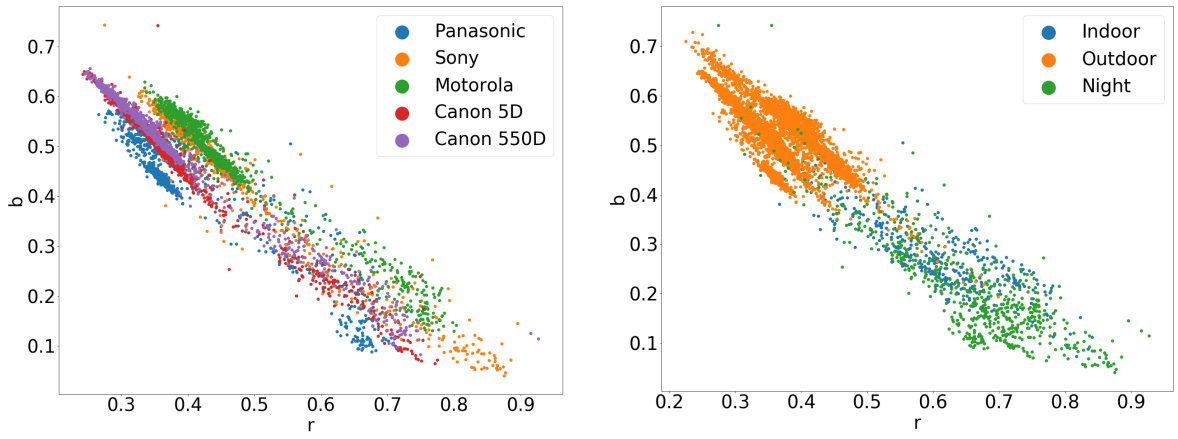
Furthermore, an artificially created set of images with more than two sources of illumination was created. It was used to test the performance of proposed methods on images in scenes that were difficult to capture in real-world conditions. Images from the Cube+ dataset [1] were used in the creation of this set of images.

4.1 Two Illuminants Dataset

In this section, a large dataset with real-world images with two sources of illumination is described. The dataset contains over 2500 different images. The images were taken using 5 different cameras. These were: Canon 5D, Canon 550D, Motorola one fusion+, Sony α300, and Panasonic FZ1000. The dataset contains approximately equal number of images taken by each camera.

Additionally, images can be separated based on the type of illumination. There are three different types of images: indoor images, outdoor daytime, and outdoor nighttime. They are each characterized by different illuminations sources. Outdoor daytime images are illuminated by sunlight and skylight, taken at different times during the day. Outdoor nighttime images contain mostly artificial LED and sodium vapor lights. Finally, indoor images contain the widest variety of illuminants. They include outdoor sunlight, LEDs, fluorescent lights, *etc.* The illumination distributions by camera type and scene type are shown in Figures 4.1a and 4.1b.

All the images are GDPR-compliant. This is done by putting a black box over privacy-violating areas. The illumination was extracted from each image using SpyderCube calibration



(a) (b)

Figure 4.1: Illumination distributions in the two-illuminant dataset. (a) The distribution by camera. (b) The distribution by type of scene.

objects that were placed in the scene. Additionally, a binary segmentation mask was manually annotated. The mask segment the image into 2 regions based on which illuminant of the two is dominant. Thus, each of the regions is, in fact, a scene illuminated by only one illuminant. The dataset has a diverse set of scenes that were taken from three different countries: Croatia, Slovenia, and Italy. Images were taken during all four seasons of the year.

Each scene contains two illuminants: the dominant illumination and the ambient illumination. For example, in a daytime scene, the sun is the dominant illuminant the ambient illuminant is the sky. To extract the illumination vectors for the scene, SpyderCube calibration objects were placed at various point in the scene. Two examples of scenes, per-pixel ground truth illumination vectors and marked SpyderCube calibration objects are shown in Figure 4.2.

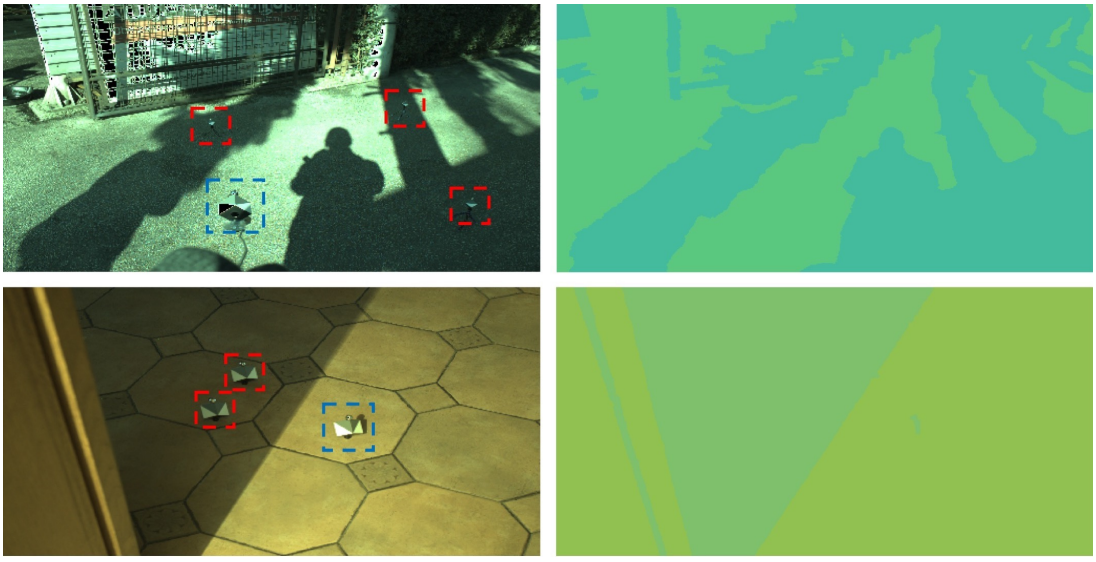


Figure 4.2: Example of two scenes and corresponding per-pixel illumination vectors. SpyderCube calibration objects are highlighted by red and blue squares, for ambient and dominant illuminants, respectively.

Each image in the dataset contains three or four SpyderCube calibration objects. One SpyderCube object was placed in the region illuminated by the dominant illuminant. The rest were placed in the regions illuminated by the ambient illuminant. More calibration objects are placed in the regions with the ambient illuminant because the ambient illumination can vary greatly through the scene. This is not the case with the dominant illuminant. To determine which images were illuminated by only two sources, the values extracted from the calibration objects for the ambient regions were compared. If the angular distance between the extracted vectors was higher than 3°, the illumination was considered nonuniform. This was done because, in [94, 95], the authors show that the human eye is sensitive to angular changes in illumination that exceed 2°-3°. This was the case for 20% of images. Therefore, these images were not included in the dataset.

The segmentation masks were obtained manually, similar to the procedure described in [87]. Masks were difficult to produce, as it was necessary to accurately annotate regions where illuminants were different. To minimize errors, the annotations were done by one person and another then checked all the annotated masks. This procedure was repeated until all the masks were satisfyingly annotated. However, since all the images contained only ambient and dominant illuminants, the amount of illuminant mixing in the scenes was minimal. Thus, the illumination masks could be approximated by hard binary masks. Without this property, manual annotation would not have been possible. However, this means that there still exists some error in the transitory areas between illumination regions.

4.2 Three Illuminants Artificial Dataset

In the previous section, real-world images with two sources of illumination were described. However, to test the performance of models that work with more than two illuminations, a dataset with at least three sources of illumination is needed. Thus, a dataset of artificially illuminated scenes is presented in this section.

The images with three sources of illumination were created artificially, as the process of gathering and annotating such images would be difficult. Furthermore, even if a number of such images could be collected, they would have to be taken in very specific conditions. This would limit the number of images well below the needed number for training of deep learning models.

The artificially illuminated images were created by taking canonically illuminated images from the Cube+ [1]. They were then illuminated with new illuminants using the von Kries model. The segmentation masks for the creation of the artificial illumination were generated by combining random linear segments. This creates a complex enough mask to be representative of real-world scenarios. Further experimentation using simpler masks was conducted There,

the regions were separated by straight lines, or by coloring random patches. However, those but those proved to be less representative of real-world scenes. Figure 4.3 shows examples of artificially relighted images.

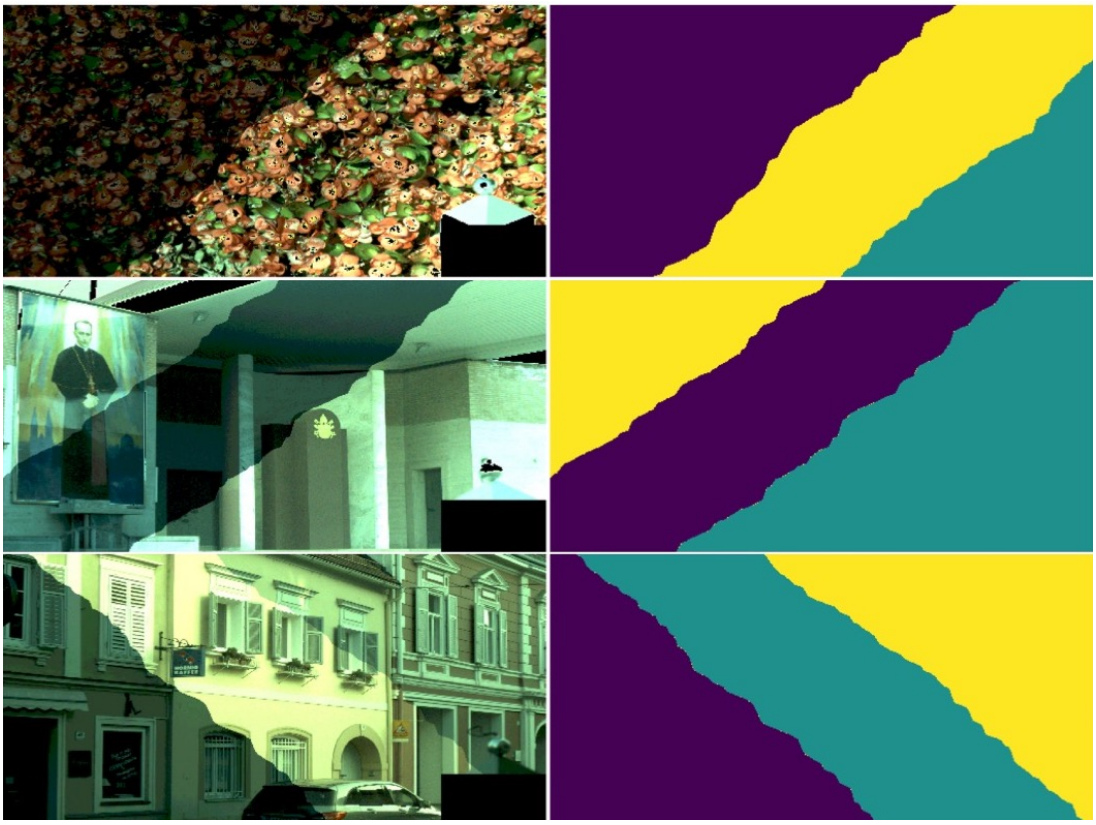


Figure 4.3: Example of scenes that were artificially illuminated by three illuminants.

These images were used to train models capable of estimating more than two illuminants per scene. They were also used for testing. However, models that were trained on such images were also tested on real-world images from the two illuminant dataset described in Section 4.1. This was done so that generalization performance of the trained models could be tested.

Chapter 5

The Main Scientific Contributions of the Thesis

The main scientific contributions of this thesis are as follows:

1. A method for illumination estimation error detection based on a transformer architecture Pub1
2. Method based on deep learning illumination encoding and detection for illuminant-based segmentation in multi-illuminant scenes with known illuminants Pub2
3. Deep learning-based method and corresponding training procedure based on autoencoder network for per-pixel multi-illuminant estimation in scenes with unknown number of illuminants Pub3
4. Deep learning framework for simultaneous illumination-based segmentation and multi-illuminant estimation for scenes with two sources of illumination Pub4

5.1 Transformer-Based Illumination Estimation Error Detection

Vision Transformers [35] have become a popular tool alongside convolutional neural networks for computer vision tasks. One such use case is the image segmentation tasks. First developed for segmentation of medical images [96], they can be repurposed, with some modifications, for other tasks. Thus, in Pub1 a method based on that transformer architecture was proposed to detect regions where the image was incorrectly white balanced. These images can easily be created by applying a global correction to scenes with more than one source of illumination. The method is based on overlapping image patches, which are then separated into smaller patches. These smaller patches are encoded using an encoding matrix and feed to the transformer, where the multiple layers of multi head self attention mechanism is used to transform those encodings

into the final prediction for each patch. The final segmentation is created by taking the smaller middle patches, and arranging them back into the image.

The method achieves state-of-the-art results, outperforming simpler models, and achieving comparable results to those produced by much more complex models. Furthermore, visual comparison of results shows that unlike simpler models, which rely heavily on brightness, the proposed model incorporates both brightness and color to create a more accurate segmentation.

5.2 Deep Learning Illumination-Based Segmentation

Illumination segmentation tasks are important because they allow for localization of the influence of different illuminants. In this section, the proposed deep learning-based method for illumination-based segmentation is presented. The method relies on the knowledge of at least $n-1$ illuminant values in scenes with n sources of illumination. However, it is shown that this isn't too restrictive of an assumption because it can be shown that global illumination estimation methods are good at accurately estimating one illuminant in the scene. The proposed method consists of two main parts. The first part performs the encoding of the illumination based on the original input image. It was shown that the best encoding is to produce an estimation of the illumination for each pixel. However, this encoding is error-prone and is not suitable for use as is. Thus, this encoding, alongside the original image and the dominant illuminant, are fed to another network which produces the final segmentation mask of the influence of that illuminant. Furthermore, an iterative approach is proposed to segment images with more than two illuminants.

The experimental results demonstrate that this method achieves state-of-the-art results on multi-illumination segmentation tasks. Furthermore, it is shown that if the dominant illuminant is first estimated instead of provided as an input, the performance does not decrease. This proves that the assumption about accurately estimating the dominant illuminant holds.

5.3 Autoencoder Based Training for Illumination Estimation

The main problem of color constancy is the fact that illumination estimation is an under-defined problem. This means that some sort of assumption has to be made about the surfaces and scenes to create an estimation. However, deep learning has proven to be a very efficient at producing implicit assumptions about the problem that is being solved. For this reason, an autoencoder type method was proposed, whose main goal was to recreate the input, but by first separating it into a per-pixel map of illumination and the canonically illuminated image. By doing this, the encoder network is forced to learn the best encoding to produce the canonically illuminated image. Furthermore, a new tri-component loss function is proposed, where mean squared error

is calculated for all three outputs (illumination estimation, canonical image, and the reconstruction). Another regularization term is used to keep the output of the illumination estimation map smooth. This regularization is based on the Laplacian filter of size n, which is used to find the discrete second derivative of the image. Finally, a K-means clustering algorithms is applied to the per-pixel estimation mask to produce the segmentation based on the illumination.

When compared to the other state-of-the-art models, the proposed model achieves better results, while also being completely independent of the number of illuminants. Furthermore, it is shown that the autoencoder model has the best generalization properties out of all the tested models. This was tested by training the model on artificial images with three illuminants and tested on real-world images with two or one illuminants.

5.4 Framework for Multi-Illumination Estimation and Segmentation

Previous sections describe methods for segmentation and estimation in scenes with multiple sources of illumination. In this section, a framework encompassing both of these tasks is presented, for the scenes with a known number of illuminants The number of illuminants must be known before training. Here, the number of illuminants is set to two, as most realworld scenes are illuminated by two sources (i.e., indoor scenes with daylight and artificial illumination, outdoor scenes with sunlight and skylight, early nighttime scenes with artificial illuminants and skylight). The framework is based on the idea that global illuminant estimation methods can produce a good estimate of the dominant illuminant in the scene. This is followed by a segmentation model that can localize the influence of the dominant illuminant to some regions of the scene. Finally, two estimation models are used to produce estimations for the areas with the dominant illuminant and the secondary illuminant. These components are first trained separately and then trained jointly.

The framework achieves state-of-the-art results for both image segmentation and estimation tasks. Furthermore, it achieves results comparable to other state-of-the-art single illumination models on single illuminant datasets, even though it was never trained on those. Finally, the results indicate that the joint training of the framework greatly increases the performance of the framework.

Chapter 6

Conclusion and Future Work

6.1 The Main Conclusions of the Thesis

The conclusions of the thesis can be separated into two parts, for the illumination estimation and segmentation. Firstly, the discussion about the image segmentation models based on the illumination was conducted. Then, models and techniques for multi-illuminant estimation were presented. These are both challenging problems that cannot be solved explicitly because they are severely under-defined. For that reason, models and methods for approximately solving these problems in various conditions are presented as the main scientific contributions of this thesis. Each of these contributions has its own conclusion. However, all of these form a whole in the problem space of multi-illuminant color constancy methods.

First, the conclusions about the scene segmentation based on illumination are presented. As a part of the thesis, two new segmentation models were presented. First is capable of detecting regions where the white balancing was not done correctly. This model is based on the vision transformer architecture. It achieves state-of-the-art performance on the two-illuminant dataset, while having at least 10 times fewer parameters than other tested models that achieve comparable results. Then, a method capable of segmenting scenes into regions illuminated only by a single source of illumination. It requires that all but one illuminant are known in the image. The method is composed of two deep convolutional neural networks which are employed to solve two tasks. The first task is to produce an encoding of the illumination from the scene, while the second model is responsible for creating the final segmentation mask based on the encoding, the known illuminant and the scene. This model outperforms all other state-of-the-art segmentation models, while having a reasonable number of parameters. Furthermore, it is shown that global illuminant estimation methods are good at estimating one illuminant in the scene accurately enough so that the real-world performance of the model is not limited by the need to know the illuminants.

By presenting these two illumination-based segmentation models, the problem of multi-

illumination estimation is simplified. These models create regions where only one illuminant is present. Now, more researched and accurate single illumination estimation models can be applied to these segmented regions. This approach was also presented as part of the contributions of this thesis in the form of a framework for simultaneous estimation and segmentation. The framework is based on the idea that, first, global illumination can be estimated using single illuminant models. Then, using the above presented segmentation models, regions with only one source of illumination are acquired. Finally, single illumination estimation models are applied to those regions to produce illuminant estimations, which are then combined using the segmentation mask into the final per-pixel estimation of the scene. This type of approach produces the best results on the two-illuminant dataset out of all the tested models, by a large margin. Furthermore, it achieves comparable results to other state-of-the-art models on single illuminant images, even though it was never trained on those datasets. However, one drawback of this model is the need to know the number of illuminants at the start of training. On the other hand, most models are still limited by the datasets in the number of illuminants they can learn to estimate.

Finally, a direct deep learning-based per-pixel estimation model was presented. It is based on an autoencoder network which is trained to learn to reproduce the canonically illuminated image and the per-pixel map of illumination of the scene. A novel tri-component loss function with smoothness constraint is used to train such a model to produce accurate reconstructions and estimations. This model again shows excellent generalization properties because it can adapt to estimating any number of illuminants when trained on artificial images with more than two sources of illumination.

Thus, it can be concluded that the problem of multi-illumination estimation is possible and has been implemented a part of this thesis. This can be done either directly or by using segmentation and single illuminant estimation as separate steps. Furthermore, visual comparison of the results also shows the clear benefits of using multi-illumination estimation models in multi-illuminant scenes. However, benefits can also be seen in images from single illuminant datasets, where some images can be illuminated by two sources of illumination accidentally. Finally, the use of multi-illumination methods in single illuminant images produces visually pleasing images in most scenarios, which shows the good generalization properties of the proposed models.

6.2 Future Work

Even though reliable methods based on deep learning have been presented in this thesis, there is still much research that can be done in the field of multi-illuminant estimation and segmentation. These can best be summarized in three research directions. First is the need for a large dataset with more than two sources of illumination. However, the creation of such a dataset is limited

by the number of such realworld scenes, thus some sort of artificial dataset would need to be constructed to augment training of deep learning models.

The second area of research is the problem of cross camera deep learning-based methods. As it stands currently, most deep learning-based methods for color constancy are implicitly tied to the sensors of the cameras present in the training dataset, and their performance usually degrades when a new camera sensor is introduced. This poses a big problem for reusability of such methods. Thus, it is necessary to create either a model or some training strategy for models that could generalize well over different and unseen camera sensors.

Finally, light sources and reflectances are inherently defined by their spectral characteristics. Thus, methods that could, in some ways, incorporate or estimate those functions, would be beneficial to solving the problem of color constancy. For that, models that can estimate the illumination spectral function for each pixel of the image could be researched. These methods would then inherently be invariant to the camera sensor, as the output would not be tied to any camera, but instead to the realworld physical property.

Chapter 7

List of publications

Pub 1 **Vršnak, D.**, Domislović, I., Subašić M., Lončarić S., "Illuminant estimation error detection for outdoor scenes using transformers", *2021 12th International Symposium on Image and Signal Processing and Analysis (ISPA)*, pp. 276-281

Pub 2 **Vršnak, D.**, Domislović, I., Subašić M., Lončarić S., "Illuminant segmentation for multi-illuminant scenes using latent illumination encoding", *Signal Processing: Image Communication*, Vol. 108, 2022, pp. 116822

Pub 3 **Vršnak, D.**, Domislović, I., Subašić M., Lončarić S., "Autoencoder-based training for multi-illuminant color constancy", *Journal of the Optical Society of America A*, Vol. 39, 2022, pp. 1076-1084

Pub 4 **Vršnak, D.**, Domislović, I., Subašić M., Lončarić S., "Framework for Illumination Estimation and Segmentation in Multi-Illuminant Scenes", *IEEE Access*, Vol. 11, 2023, pp. 2128-2137

Chapter 8

Author's contribution to the publications

The results presented in this thesis are based on the research carried out during the period of 2020-2022 at the University of Zagreb, Faculty of Electrical Engineering and Computing, Unska 3, HR-10000 Zagreb, Croatia, as a part of an industry financed project.

The thesis includes four publications written in collaboration with coauthors of the published papers. The author's contribution to each paper consists of the experiment idea, software implementation, performing the required experiments, results analysis and text writing.

[Pub1] In the paper **"Illuminant estimation error detection for outdoor scenes using transformers"** the author has proposed a novel transformer network capable of detecting regions in images where white balancing is incorrect. The method is used on multi-illuminant images to determine regions where further correction will need to be applied. The benefit of this approach is the ability to utilize well-researched single illuminant estimation methods on multi-illuminant images.

[Pub2] In the paper **"Illuminant segmentation for multi-illuminant scenes using latent illumination encoding"** a deep learning-based method for segmenting multi-illuminant scenes based on known illumination vectors is proposed. The method is tested on realworld images with two sources of illumination, as well as on artificially generated images with more than two sources. On all images, this method produces the best results, outperforming even more complex methods.

[Pub3] In the paper **"Autoencoder-based training for multi-illuminant color constancy"** a model for per-pixel estimation of illumination is presented. The model is based on the autoencoder network, where the goal is to recreate the raw input image. The evaluation of the autoencoder training method shows it outperforms other methods and achieves better generalization properties independent of the number of illuminants.

[Pub4] In the paper **"Framework for Illumination Estimation and Segmentation in Multi-Illuminant Scenes"** the author proposed a framework capable of segmenting and estimating per-pixel illumination in scenes with two sources of illumination. The method pro-

duces outperforms other state-of-the-art results on the two-illuminant dataset. Additionally, it achieves comparable results to other state-of-the-art methods on single illuminant dataset, even though it was not trained on them, demonstrating good generalization properties.

Bibliography

[1] Banic, N., Loncaric, S., “Unsupervised learning for color constancy”, str. 181-188, dostupno na: http://arxiv.org/abs/1712.00436 2017.

[2] Afifi, M., Brown, M., “What else can fool deep learning? addressing color constancy errors on deep neural network performance”, in 2019 IEEE/CVF International Conference on Computer Vision (ICCV), 2019, str. 243-252.

[3] Monge, G., “Mémoire sur quelques phénomènes de la vision”, in Annales de Chimie, Vol. 3, 1789, str. 131–147.

[4] Hering, E., Grundzüge der Lehre vom Lichtsinn. Springer, 1920.

[5] von Kries, J., “Theoretische studien über die umstimmung des sehorgans”, Festschrift der Albrecht-Ludwigs-Universität, 1902, str. 145–158.

[6] Judd, D. B., “Hue saturation and lightness of surface colors with chromatic illumination”, JOSA, Vol. 30, No. 1, 1940, str. 2–32.

[7] Land, E. H., McCann, J. J., “Lightness and retinex theory”, Josa, Vol. 61, No. 1, 1971, str. 1–11.

[8] Helson, H., Jeffers, V. B., “Fundamental problems in color vision. ii. hue, lightness, and saturation of selective samples in chromatic illumination.”, Journal of Experimental Psychology, Vol. 26, No. 1, 1940, str. 1.

[9] Schacter, D., Gilbert, D. T., Wegner, D. M., Psychology (2nd Edition). New York: Worth, 2011, dostupno na: http://www.amazon.com/Psychology-Daniel-L-Schacter/dp/1429237198/ref=sr_1_1?s=books&ie=UTF8&qid=1313937150&sr=1-1

[10] Koppal, S. J., Lambertian Reflectance. Boston, MA: Springer US, 2014, str. 441-443, dostupno na: https://doi.org/10.1007/978-0-387-31439-6_534

[11] Logvinenko, A. D., Kane, J., Ross, D. A., “Is lightness induction a pictorial illusion?”, Perception, Vol. 31, No. 1, 2002, str. 73–82.

[12] Logvinenko, A. D., Petrini, K., Maloney, L. T., "A scaling analysis of the snake lightness illusion", Perception & Psychophysics, Vol. 70, No. 5, 2008, str. 828–840.

[13] Logvinenko, A. D., Tokunaga, R., "Colour constancy as measured by least dissimilar matching", Seeing and perceiving, Vol. 24, No. 5, 2011, str. 407–452.

[14] Logvinenko, A. D., Funt, B., Mirzaei, H., Tokunaga, R., "Rethinking colour constancy", PLoS One, Vol. 10, No. 9, 2015, str. e0135029.

[15] Maloney, L. T., "Evaluation of linear models of surface spectral reflectance with small numbers of parameters", JOSA A, Vol. 3, No. 10, 1986, str. 1673–1683.

[16] D'Zmura, M., "Color constancy: surface color from changing illumination", JOSA A, Vol. 9, No. 3, 1992, str. 490–493.

[17] Hurlbert, A., "Formal connections between lightness algorithms", JOSA A, Vol. 3, No. 10, 1986, str. 1684–1693.

[18] Arend, L., Reeves, A., "Simultaneous color constancy", JOSA A, Vol. 3, No. 10, 1986, str. 1743–1751.

[19] Arend, L. E., Reeves, A., Schirillo, J., Goldstein, R., "Simultaneous color constancy: papers with diverse munsell values", JOSA A, Vol. 8, No. 4, 1991, str. 661–672.

[20] Troost, J. M., De Weert, C. M., "Naming versus matching in color constancy", Perception & psychophysics, Vol. 50, No. 6, 1991, str. 591–602.

[21] VON KRIES, J., "Influence of adaptation on the effects produced by luminous stimuli", handbuch der Physiologie des Menschen, Vol. 3, 1905, str. 109-282, dostupno na: https://ci.nii.ac.jp/naid/10030415665/en/

[22] Zeki, S., "The representation of colours in the cerebral cortex", Nature, Vol. 284, No. 5755, 1980, str. 412–418.

[23] Zeki, S., "Colour coding in the cerebral cortex: the reaction of cells in monkey visual cortex to wavelengths and colours", Neuroscience, Vol. 9, No. 4, 1983, str. 741–765.

[24] Wild, H., Butler, S., Carden, D., Kulikowski, J., "Primate cortical area v4 important for colour constancy but not wavelength discrimination", Nature, Vol. 313, No. 5998, 1985, str. 133–135.

[25] Burton, G. J., Moorhead, I. R., "Color and spatial structure in natural scenes", Applied optics, Vol. 26, No. 1, 1987, str. 157–170.

[26] Reeves, A., Amano, K., "Color and brightness constancies depend reciprocally on saturation", JOSA A, Vol. 37, No. 4, 2020, str. A237–A243.

[27] Reeves, A., Amano, K., Foster, D., "Color constancy: stimulus or task", Journal of Vision, Vol. 4, No. 11, 2004, str. 12–12.

[28] Reeves, A. J., Amano, K., Foster, D. H., "Color constancy: phenomenal or projective?", Perception & psychophysics, Vol. 70, No. 2, 2008, str. 219–228.

[29] Kelly, J. M. F., Daugirdiene, A., Kulikowski, J. J., Murray, I. J., "Chips in the sunshine: color constancy with real versus simulated munsell chips under illuminants adjacent to the daylight locus", J. Opt. Soc. Am. A, Vol. 35, No. 4, Apr 2018, str. B100–B105, dostupno na: https://opg.optica.org/josaa/abstract.cfm?URI=josaa-35-4-B100

[30] Schanda, J., Colorimetry: Understanding the CIE System. John Wiley & Sons, Ltd, 2007, dostupno na: https://onlinelibrary.wiley.com/doi/abs/10.1002/9780470175637

[31] Rosenblatt, F., "The perceptron: a probabilistic model for information storage and organization in the brain.", Psychological review, Vol. 65, No. 6, 1958, str. 386.

[32] LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., Jackel, L. D., "Backpropagation applied to handwritten zip code recognition", Neural computation, Vol. 1, No. 4, 1989, str. 541–551.

[33] Zhou, Y.-T., Chellappa, R., Vaid, A., Jenkins, B. K., "Image restoration using a neural network", IEEE transactions on acoustics, speech, and signal processing, Vol. 36, No. 7, 1988, str. 1141–1151.

[34] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, u., Polosukhin, I., "Attention is all you need", in Proceedings of the 31st International Conference on Neural Information Processing Systems, ser. NIPS'17. Red Hook, NY, USA: Curran Associates Inc., 2017, str. 6000–6010.

[35] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N., "An image is worth 16x16 words: Transformers for image recognition at scale", CoRR, Vol. abs/2010.11929, 2020, dostupno na: https://arxiv.org/abs/2010.11929

[36] Amari, S.-i., "Backpropagation and stochastic gradient descent method", Neurocomputing, Vol. 5, No. 4-5, 1993, str. 185–196.

[37] Duchi, J., Hazan, E., Singer, Y., "Adaptive subgradient methods for online learning and stochastic optimization", Journal of Machine Learning Research, Vol. 12, No. 61, 2011, str. 2121–2159, dostupno na: http://jmlr.org/papers/v12/duchi11a.html

[38] Igel, C., Hüsken, M., "Improving the rprop learning algorithm", in Proceedings of the second international ICSC symposium on neural computation (NC 2000), Vol. 2000, 2000, str. 115–121.

[39] Kingma, D. P., Ba, J., "Adam: A method for stochastic optimization", 2017.

[40] van de Weijer, J., Gevers, T., Gijsenij, A., "Edge-based color constancy", IEEE Transactions on Image Processing, Vol. 16, No. 9, 2007, str. 2207-2214.

[41] Land, E. H., "The retinex theory of color vision", Scientific american, Vol. 237, No. 6, 1977, str. 108–129.

[42] Buchsbaum, G., "A spatial processor model for object colour perception", Journal of the Franklin Institute, Vol. 310, No. 1, 1980, str. 1-26, dostupno na: https://www.sciencedirect.com/science/article/pii/0016003280900587

[43] Finlayson, G. D., Trezzi, E., "Shades of gray and colour constancy", in Color and Imaging Conference, Vol. 1. Society for Imaging Science and Technology, 2004, str. 37–41.

[44] Van De Weijer, J., Gevers, T., Gijsenij, A., "Edge-based color constancy", IEEE Transactions on image processing, Vol. 16, No. 9, 2007, str. 2207–2214.

[45] Zakizadeh, R., Brown, M. S., Finlayson, G. D., "A hybrid strategy for illuminant estimation targeting hard images", in Proceedings of the IEEE International Conference on Computer Vision Workshops, 2015, str. 16–23.

[46] Lee, H.-C., "Method for computing the scene-illuminant chromaticity from specular highlights", JOSA A, Vol. 3, No. 10, 1986, str. 1694–1699.

[47] Tominaga, S., Wandell, B. A., "Standard surface-reflectance model and illuminant estimation", JOSA A, Vol. 6, No. 4, 1989, str. 576–584.

[48] Tan, R. T., Ikeuchi, K., Nishino, K., Color Constancy through Inverse-Intensity Chromaticity Space. Boston, MA: Springer US, 2008, str. 323-351, dostupno na: https://doi.org/10.1007/978-0-387-75807_16

[49] Healey, G., "Estimating spectral reflectance using highlights", Image and vision computing, Vol. 9, No. 5, 1991, str. 333–337.

[50] Woo, S.-M., Lee, S.-H., Yoo, J.-S., Kim, J.-O., "Improving color constancy in an ambient light environment using the phong reflection model", IEEE Transactions on Image Processing, Vol. 27, No. 4, 2017, str. 1862–1877.

[51] Joze, H. R. V., Drew, M. S., "Exemplar-based color constancy and multiple illumination", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 36, No. 5, 2014, str. 860-873.

[52] Funt, B., Xiong, W., "Estimating illumination chromaticity via support vector regression", in Color Imaging Conference, Vol. 50, 01 2004, str. 47-52.

[53] Agarwal, V., Gribok, A. V., Abidi, M. A., "Machine learning approach to color constancy", Neural Networks, Vol. 20, No. 5, 2007, str. 559-563, dostupno na: https://www.sciencedirect.com/science/article/pii/S0893608007000846

[54] Agarwal, V., Gribok, A. V., Koschan, A., Abidi, M. A., "Estimating illumination chromaticity via kernel regression", in 2006 International Conference on Image Processing, 2006, str. 981-984.

[55] Laakom, F., Raitoharju, J., Iosifidis, A., Tuna, U., Nikkanen, J., Gabbouj, M., "Probabilistic color constancy", CoRR, Vol. abs/2005.02730, 2020, dostupno na: https://arxiv.org/abs/2005.02730

[56] Sapiro, G., "Color and illuminant voting", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 21, No. 11, 1999, str. 1210-1215.

[57] Gehler, P. V., Rother, C., Blake, A., Minka, T., Sharp, T., "Bayesian color constancy revisited", in 2008 IEEE Conference on Computer Vision and Pattern Recognition. IEEE, 2008, str. 1–8.

[58] Rosenberg, C., Ladsariya, A., Minka, T., "Bayesian color constancy with non-gaussian models", Advances in neural information processing systems, Vol. 16, 2003, str. 1595–1602.

[59] Brainard, D. H., Freeman, W. T., "Bayesian color constancy", JOSA A, Vol. 14, No. 7, 1997, str. 1393–1411.

[60] Bianco, S., Cusano, C., Schettini, R., "Color constancy using cnns", in Proceedings of the IEEE conference on computer vision and pattern recognition workshops, 2015, str. 81–89.

[61] Hu, Y., Wang, B., Lin, S., "Fc4: Fully convolutional color constancy with confidence-weighted pooling", in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, str. 4085–4094.

[62] Koščević, K., Subašić, M., Lončarić, S., "Iterative convolutional neural network-based illumination estimation", IEEE Access, Vol. 9, 2021, str. 26 755-26 765.

[63] Simonyan, K., Zisserman, A., "Very deep convolutional networks for large-scale image recognition", 2015.

[64] Choi, H.-H., Yun, B.-J., "Very deep learning-based illumination estimation approach with cascading residual network architecture (crna)", IEEE Access, Vol. 9, 2021, str. 133 552-133 560.

[65] Shi, W., Loy, C. C., Tang, X., "Deep specialized network for illuminant estimation", in European conference on computer vision. Springer, 2016, str. 371–387.

[66] Laakom, F., Passalis, N., Raitoharju, J., Nikkanen, J., Tefas, A., Iosifidis, A., Gabbouj, M., "Bag of color features for color constancy", IEEE Transactions on Image Processing, Vol. 29, 2020, str. 7722–7734, dostupno na: http://dx.doi.org/10.1109/TIP.2020.3004921

[67] Koščević, K., Subašić, M., Lončarić, S., "Deep learning-based illumination estimation using light source classification", IEEE Access, Vol. 8, 2020, str. 84 239-84 247.

[68] Barron, J. T., "Convolutional color constancy", in Proceedings of the IEEE International Conference on Computer Vision, 2015, str. 379–387.

[69] Barron, J. T., Tsai, Y.-T., "Fast fourier color constancy", in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, str. 886–894.

[70] Hussain, M. A., Akbari, A. S., "Max-rgb based colour constancy using the sub-blocks of the image", in 2016 9th International Conference on Developments in eSystems Engineering (DeSE), 2016, str. 289-294.

[71] Hussain, M. A., Akbari, A. S., "Color constancy algorithm for mixed-illuminant scene images", IEEE Access, Vol. 6, 2018, str. 8964-8976.

[72] Hussain, M. A., Sheikh-Akbari, A., Abbott Halpin, E., "Color constancy for uniform and non-uniform illuminant using image texture", IEEE Access, Vol. 7, 2019, str. 72 964-72 978.

[73] Gijsenij, A., Lu, R., Gevers, T., "Color constancy for multiple light sources", IEEE Transactions on Image Processing, Vol. 21, No. 2, 2012, str. 697-707.

[74] Akazawa, T., Kinoshita, Y., Shiota, S., Kiya, H., "N-white balancing: White balancing for multiple illuminants including non-uniform illumination", IEEE Access, Vol. 10, 2022, str. 89 051-89 062.

[75] Akbarinia, A., Parraga, C. A., "Colour constancy beyond the classical receptive field", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 40, No. 9, 2018, str. 2081-2094.

[76] Beigpour, S., Riess, C., Weijer, J., Angelopoulou, E., "Multi-illuminant estimation with conditional random fields", IEEE transactions on image processing : a publication of the IEEE Signal Processing Society, 01 2014, str. 83-96.

[77] Bianco, S., Cusano, C., Schettini, R., "Single and multiple illuminant estimation using convolutional neural networks", IEEE Transactions on Image Processing, Vol. 26, No. 9, 2017, str. 4347-4362.

[78] Das, P., Baslamisli, A. S., Liu, Y., Karaoglu, S., Gevers, T., "Color constancy by gans: An experimental survey", CoRR, Vol. abs/1812.03085, 2018, dostupno na: http://arxiv.org/abs/1812.03085

[79] Isola, P., Zhu, J.-Y., Zhou, T., Efros, A. A., "Image-to-image translation with conditional adversarial networks", 2018.

[80] Lee, S.-H., Woo, S.-M., Choi, J.-H., Kim, J.-O., "Two-step multi-illuminant color constancy for outdoor scenes", in 2017 IEEE International Conference on Image Processing (ICIP), 09 2017, str. 710-714.

[81] Ciurea, F., Funt, B., "A large image database for color constancy research", in Color Imaging Conference, 2003.

[82] Gehler, P. V., Rother, C., Blake, A., Minka, T., Sharp, T., "Bayesian color constancy revisited", in 2008 IEEE Conference on Computer Vision and Pattern Recognition, 2008, str. 1-8.

[83] Cheng, D., Prasad, D. K., Brown, M. S., "Illuminant estimation for color constancy: why spatial-domain methods work and the role of the color distribution", J. Opt. Soc. Am. A, Vol. 31, No. 5, May 2014, str. 1049-1058, dostupno na: http://josaa.osa.org/abstract.cfm?URI=josaa-31-5-1049

[84] Laakom, F., Raitoharju, J., Iosifidis, A., Nikkanen, J., Gabbouj, M., "Intel-tau: A color constancy dataset", arXiv preprint arXiv:1910.10404, 2019.

[85] Shi, L., "Re-processed version of the gehler color constancy dataset of 568 images", http://www. cs. sfu. ca/~ color/data/, 2000.

[86] Banić, N., Koščević, K., Subašić, M., Lončarić, S., "The past and the present of the color checker dataset misuse", in 2019 11th International Symposium on Image and Signal Processing and Analysis (ISPA). IEEE, 2019, str. 366-371.

[87] Gijsenij, A., Lu, R., Gevers, T., "Color constancy for multiple light sources", IEEE Transactions on Image Processing, Vol. 21, No. 2, 2012, str. 697-707.

[88] Finlayson, G., Zakizadeh, R., "Reproduction angular error: An improved performance metric for illuminant estimation", in BMVC, 2014.

[89] Sidorov, O., "Artificial color constancy via googlenet with angular loss function", CoRR, Vol. abs/1811.08456, 2018, dostupno na: http://arxiv.org/abs/1811.08456

[90] Li, Z., Ma, Z., "Robust white balance estimation using joint attention and angular loss optimization", in Thirteenth International Conference on Machine Vision, Vol. 11605. International Society for Optics and Photonics, 2021, str. 116051E.

[91] Zou, K., Warfield, S., Bharatha, A., Tempany, C., Kaus, M., Haker, S., Wells, W., Jolesz, F., Kikinis, R., "Statistical validation of image segmentation quality based on a spatial overlap index", Academic radiology, Vol. 11, 02 2004, str. 178-89.

[92] Hordley, S. D., Finlayson, G. D., "Re-evaluating colour constancy algorithms", in Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004., Vol. 1. IEEE, 2004, str. 76–79.

[93] Adelson, E. H., "Perceptual organization and the judgment of brightness", Science, Vol. 262, No. 5142, 1993, str. 2042–2044.

[94] Hordley, S. D., "Scene illuminant estimation: past, present, and future", Color Research & Application: Endorsed by Inter-Society Color Council, The Colour Group (Great Britain), Canadian Society for Color, Color Science Association of Japan, Dutch Society for the Study of Color, The Swedish Colour Centre Foundation, Colour Society of Australia, Centre Français de la Couleur, Vol. 31, No. 4, 2006, str. 303–314.

[95] Gijsenij, A., Gevers, T., Lucassen, M. P., "Perceptual analysis of distance measures for color constancy algorithms", JOSA A, Vol. 26, No. 10, 2009, str. 2243–2256.

[96] Han, K., Xiao, A., Wu, E., Guo, J., Xu, C., Wang, Y., "Transformer in transformer", arXiv preprint arXiv:2103.00112, 2021.

Publications

Publication 1

Vršnak, D., Domislović, I., Subašić M., Lončarić S., *2021 12th International Symposium on Image and Signal Processing and Analysis (ISPA)*, pp. 276-281

Illuminant estimation error detection for outdoor scenes using transformers

Donik Vršnak, Ilija Domislović, Marko Subašić, Sven Lončarić
{donik.vrsnak, ilija.domislovic, marko.subasic, sven.loncaric}@fer.hr
Image Processing Laboratory, Faculty of Electrical Engineering and Computing, University of Zagreb

2021 12th International Symposium on Image and Signal Processing and Analysis (ISPA) | 978-1-6654-2639-8/21/$31.00 ©2021 IEEE | DOI: 10.1109/ISPA52656.2021.9552045

***Abstract*—Color constancy is an important property of the human visual system that allows us to recognize the colors of objects regardless of the scene illumination. Computational color constancy is an unavoidable part of all modern camera image processing pipelines. However, most modern computational color constancy methods focus on the estimation of only one illuminant per scene, even though the scene may have multiple illuminations, such as very common outdoor scenes illuminated by sunlight. In this work, we address this problem by creating a deep learning model for image segmentation based on the transformer architecture, which can identify regions in outdoor scenes where the global estimation and subsequent color correction of the image is not accurate. We compare our convolution-free model to a convolutional model and a more simple baseline model and achieve excellent results.**

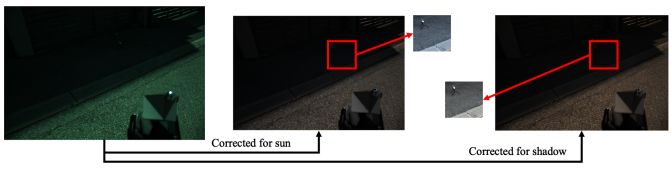


Figure 1. Image showing two different corrections of a raw image obtained by Nikon D7000 DSLR camera. The first image is corrected for the primary illuminant which is the sun, which casts the shadow regions to an unnaturally blue hue. The second image is corrected for the shaded illuminant, which casts the sunlit region into an orange hue. Values for the primary and secondary illuminants were obtained from the gray sides of the SpyderCube calibration object.

I. Introduction

Color constancy is the ability of the human visual system (HVS) to perceive colors of objects independent from the color of the light that is illuminating them. Computational color constancy refers to the method for eliminating the color of illumination in the scene (and thus making the scene look as if it was taken under canonical illumination) and is usually done as a post-processing step in modern camera image processing pipelines. However, unlike the HVS, which can perform this operation very accurately and in real-time, this problem is very difficult to solve using only the R, G, B values of pixels captured from the camera sensor. The problem arises because the captured pixel $p_c, c \in \{R, G, B\}$ values are the product of three factors: spectral reflectance properties $R(\lambda)$ and colors of objects, the spectrum of light that illuminates them $I(\lambda)$, and the spectral characteristic of the sensor $S_c(\lambda)$ (Eq. 1). In most cases, both $I(\lambda)$ and the $R(\lambda)$ are not known, which makes the problem of computational color constancy ill-posed, as there are infinite combinations of surface reflectance and illumination that could produce the captured pixel values:

$$p_c(x, y) = \int_\omega I(x, y, \lambda) R(x, y, \lambda) S_c(\lambda) d\lambda \qquad (1)$$

This means that to solve the illuminant estimation problem, additional assumptions have to be made. The most common assumption is to remove the spatial component and assume that there is only one global illuminant present in the scene. However, this is not enough and further assumptions and constraints have to be set on the reflectance properties of the scene. These assumptions can vary a lot, resulting in a wide variety of color constancy methods.

In this work, we focus on the first assumption used by most modern color constancy methods - that the scene is illuminated by a single global illuminant. This assumption can often be incorrect in many common scenes, such as outdoor images taken under direct sunlight. Two illuminants can be found in those scenes, one in the sunlit regions and one in the shaded regions, which can appear very different to the observer. The shaded regions differ from the sunlit regions because they are illuminated by the ambient illumination of the sky, where atmospheric diffraction and absorption create a blue illuminant in comparison to the yellowish illumination of the sun. However, many global methods can accurately estimate one of these illuminants while ignoring the other. If such an estimation is used to correct the image, the resulting image will have areas where the illumination is almost canonical (i.e. white, $e_c = [\frac{1}{3}\,\frac{1}{3}\,\frac{1}{3}]$) and other areas where that is not the case, as can be seen in Figure 1. This difference in illumination can make the image appear wrong to the observer and can even fool computer vision methods for classification and object recognition [2].

In this work, we propose a novel method using a vision transformer architecture for segmentation of outdoor scene based on the difference of illumination correction. The proposed method can accurately segment regions of the scene where the non-canonical e_u illuminant is present, which would allow us to detect areas in the original image where the illumination has to be re-estimated. To test and train the proposed model, we collected and annotated outdoor images with multiple illuminations which are available online [14]. However, the work on this new dataset is still in progress and we use only

a subset of images in all of our experiments. We show that our method outperforms baseline models and achieves results that are on par with a much more complex Unet model based on the VGG16 architecture. We also show that those results are achieved with only a small number of parameters, which makes it more feasible to implement in lower-end hardware, such as digital cameras and smartphones.

In the next section, we will explore the current state of the art in color constancy and image segmentation, as well as the ever more prevalent use of transformer architecture for computer vision tasks. In Section III we provide a detailed description of the model architecture and and implementation details, whose code is available online [1]. Then, in Section IV we describe the experiments and results of our models, which we tested on a set of outdoor multi-illuminant images.

II. Related Work

A. Color Constancy

Color constancy is a field that has been studied for almost 200 hundred years, starting with publications by [26], [25] and others. However, this work focused on the human perception of illumination and the physical properties of light. With the advent of digital cameras, computational color constancy has become a highly studied topic, as it is one of the first steps in all modern camera pipelines. For this reason, computational color constancy has been extensively studied, and most color constancy methods can be sorted into two categories: statistics based and learning based methods. Statistics-based methods exploit low-level statistical properties of images to offset the fact that color constancy is an ill-posed problem. These methods assume some property of the reflectance, such as that the mean color of all pixels in an image is gray [8], or that the brightest part of the image is the pure reflection of the illuminant [19], or that the n^{th} derivation of the image is gray [23]. These methods usually have very low complexity and high performance and are suitable for implementation on camera hardware. On the other hand, learning-based methods require training on images to learn the best assumptions needed to solve the color constancy problem. Such methods can be used to learn the gamut mapping of illuminants[11], or some higher-level statistical properties like the use of exemplar learning to find similar surfaces in images [15]. With the popularization of convolutional neural networks (CNNs) for computer vision tasks, many modern learning based color constancy methods utilize some sort of deep learning to estimate the illuminant. The first use of CNNs to solve illuminant estimation was done by [7], which used a relatively simple CNN. Other methods such as [4] and [5] utilize only chroma histograms to predict the illumination, while [13] proposed a fully connected network with simple attention mechanism for illuminant estimation. [18] simplifies estimation by first doing light source classification to reduce the space of illuminant that the model has to predict.

One thing all of the methods described above have in common is that they assume that there is only one global illumination in the scene. However, even in the case of outdoor scenes, this assumption is often broken (See Section I), which results in wrong estimation of both or just one illuminant. In [6], the authors proposed a method that used image patches and applied a conditional random field to try to offset the inherent problem of estimating illumination from a small patch. However, their method had high computational complexity for predicting the illumination mask. On the other hand, a simple method based on brightness threshold for outdoor images presented in [20] is the closest method that can be used for comparison as, to our knowledge, there aren't any other published methods that perform segmentation based on illumination. However, as with many multi-illuminant estimation methods, the main problem is the lack of a large well annotated dataset, as all work that has been done in the field was tested only on small datasets, each of which provided only a small number of images. In this work, we tested our method on a subset of annotated images from our dataset, which we will make available online as a whole in the future, while the images used in this work are already available online [14].

B. Vision Transformer Model

Transformer architecture [24] has gained a lot of traction in the natural language processing (NLP) space as it provided State of the Art (SOTA) performance and could be trained on large corpora of unannotated text. Because of the great success of transformers in natural language processing, [10] used the architecture on an image classification task. They achieved results comparable to the state of the art CNNs while not using any convolutions. However, the main drawback is that to train the model, significant time and computational resources were required, as well as a lot of pretraining on a large number of images. Recent advances in training optimization for vision transformers as well as combination with convolutional layers, such as [22], [9] and [12] made vision transformers easier to train. Applications of transformer models in computer vision extended into medical and volumetric image segmentation, where [16] proposed an architecture for block segmentation of 3D CT images without using any convolutional layers. In this work, we propose a method that is based on the same idea proposed in [16], but we adapted the architecture to serve the need of 2D illumination segmentation for outdoor scenes.

III. Model Description

A. Transformer model

Figure 2 shows the architecture of the model, which takes a block $B \in \mathbb{R}^{b \times b \times c}$ of image $I \in \mathbb{R}^{H \times W \times c}$ as input and outputs the segmentation class for the center patch of the block. The input block is then split into $N = n \times n$ non-overlapping patches $p_i \in \mathbb{R}^{\frac{b}{n} \times \frac{b}{n} \times c}$ which are flattened and stacked to form the input $P \in \mathbb{R}^{N \times \frac{b}{n}^2 c}$, embedded to match the dimensions of the transformer model d_{model} using an embedding matrix $W_{emb} \in \mathbb{R}^{\frac{b}{n}^2 c \times d_{model}}$. Learnable positional encoding $W_{pos} \in \mathbb{R}^{\frac{b}{n}^2 c \times d_{model}}$ is added to the embedded

[1] https://github.com/donikv/TIEED

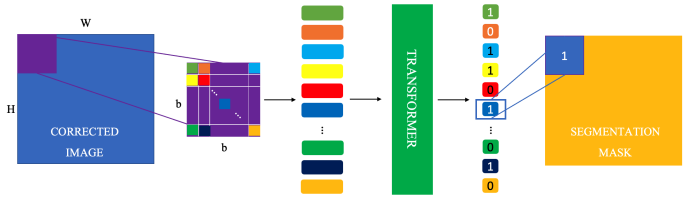


Figure 2. Image showing a simple diagram of the input pipeline to the transformer model. The transformer model is composed of K MHSA blocks. The final output for one block of the input image is determined by the classification of the central patch of the block. Blocks are taken from the input with overlap, using stride equal to the size of one patch of the block. Thus the resolution of the segmentation map is only reduced by the size of the patch, and not by the size of a whole block.

input P_{emb} to create the final input to the transformer model $P_{in} = PW_{emb} + W_{pos}$.

The transformer model is composed of several layers which are in turn composed of multi-head self-attention (MHSA) followed by layer normalization and two fully-connected layers with ReLU activation. Skip connections are added between MHSA and FC layers. The $j^{th}, k \in 1...K$ layer of the transformer performs the following set of operations:

1) The input P_{j-1} is passed to the MHSA layer which first computes the query q_i, key k_i and value v_i matrices using learnable matrices $W_{q,i}, W_{k,i}, W_{v,i} \in \mathbb{R}^{d_{model} \times d_h}$ ($d_h = d_{model}/m$) for each attention head i, where m is the number of heads in multi-attention.

$$q_i = W_{q,i}P_{j-1}, k_i = W_{k,i}P_{j-1}, v_i = W_{v,i}P_{j-1} \quad (2)$$

$$A_i = softmax(q_i k_i^T) \quad (3)$$

$$SA_i = A_i V_i \quad (4)$$

2) The self-attention output are of each head i is concatenated to create the final self-attention matrix $SA = [SA_0, \dots SA_i, \dots, SA_m]$ which is added to the input P_{j-1}

$$X_{j,sa} = P_{j-1} + SA_j \quad (5)$$

3) The output $X_{j,sa}$ of the MHSA layer is then feed through layer normalization and through two fully connected layers. The output of the MSHA layer $X_{j,sa}$ is finally added to the output of the fully connected layers $X_{j,fc}$ and passed through another layer normalization.

$$X_{j,fc} = ReLU(X_{j,sa}W_{j,1} + b_{j,1})W_{j,2} + b_{j,2} \quad (6)$$

$$X_k = X_{j,fc} + X_{j,sa} \quad (7)$$

4) The output of the final layer X_K is passed through another fully connected layer with sigmoid activation to predict the segmentation class Y_K for the central patch $p_{\lceil N/2 \rceil}$ of block B. The predictions for the other $N-1$ patches in the block are discarded.

$$\begin{aligned} Y_K &= sigmoid(X_K W_K + b_K) \\ W_K &\in \mathbb{R}^{d_{model} \times 1}, Y_K \in \mathbb{R}^{N \times 1} \end{aligned} \quad (8)$$

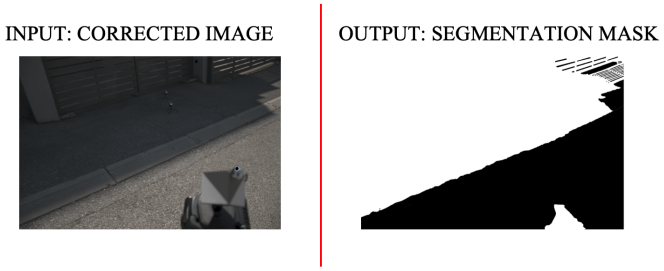


Figure 3. Example of the input and the groundtruth images in the dataset.

5) The final prediction is done by arranging the block segmentation predictions into an output mask $M \in \mathbb{R}^{\frac{Hn}{b} \times \frac{Wn}{b} \times 1}$. To compensate for the reduction in size of the predicted mask we upscale the prediction by a factor of $\frac{b}{n}$.

The input to the transformer model was corrected for the illumination of the sun using the Von Kries model [25] to simulate the effects of global correction which is correct for some regions of the image and wrong for others. The goal of our model is to predict which areas were wrongly estimated and corrected.

B. Dataset

The lack of a large well annotated multi-illuminant dataset has slowed down the progress of multi-illuminant estimation methods. For this reason, we collected and manually annotated more than 600 images, predominantly of outdoor scenes. This initial version of the dataset is freely available [14], but is still considered a work in progress as we are still collecting and annotating images. We used this initial set of outdoor images to train our segmentation model for the task of detecting wrongly corrected regions in images. We split the data into training, validation, and testing sets, with 310, 77, and 96 images respectively. Figure 3 shows an example of the image from the dataset and the globally corrected image which fed into the model.

Prior to giving the corrected images to our model, we removed underexposed and clipped pixels (as they provide no additional data) and applied brightness normalization as a pre-processing step. We didn't do per-channel normalization common in other computer vision tasks as it can affect the colors of the image, thus making the annotation labels inaccurate. To combat the limited number of training examples and prevent overfitting, we augmented the data using random center cropping, rotation from -15 to 15 degrees, and horizontal and vertical flipping.

C. Experiments and Training

The model was implemented in TensorFlow 2.4.[1] and trained on an RTX 2080Ti GPU. The training was done over 10000 epochs, where each epoch consisted of a pass through $1/50^{th}$ of the dataset. Learning rate was set using cosine annealing warm restart [21] schedule, with range from $1 * 10^{-4}$ to $1 * 10^{-6}$ and the weights were optimized by Adam optimizer

[17] with weight decay of $1 * 10^{-7}$. We implemented two different types of models, a smaller transformer model with $m = 8$ multi-attention heads, of depth $K = 7$ and $d_{model} = 64$, and a larger model with $m = 8$ and $K = 10$ and $d_{model} = 128$. Both models used the same block size and number of patches in a block of $b = 25$ and $n = 5$. We trained these models on outdoor images which were pre-processed and augmented as described in Section III-B. We also tried pretraining the larger model on a set of artificial images to offset the need for more training examples to prevent overfitting, as is described in [10]. Each artificial image consisted of several images from the Cube+ dataset [3] which were put into a collage. The border between the images in the collage was constructed using a segmented linear function to create a more complex border to prevent overfitting.

IV. Results

In this section, we present the results of our models in comparison to two baseline models and a convolutional model based on the UNet architecture with a VGG16 encoder. We also performed tests with other smaller architectures, such as the SEResnet18 UNet and a smaller custom FPN architecture with approx 6 million parameters, but for brevity, we only present the results of the VGG16 based architecture as it performed the best out of those architectures. The baseline models used are based on setting a threshold in the brightness histogram of the corrected image. We set the threshold either to the value of $c_t = 0.1$ of the maximum image brightness or by using OTSU threshold [27] method. We then classify each pixel under the threshold value as the wrongly corrected pixel (since we corrected the image for the primary illuminant created by direct sunlight, that is present in the brighter parts of the image) and those above the threshold as correctly illuminated pixels. We use the static threshold to compare our model to [20], the only other (to our knowledge) method which performs illumination segmentation independent of estimation. However, we changed the c_t value from 0.08 to 0.1 because it achieved slightly better results on our images.

Table I shows the comparison of the results between the transformer models and the baseline models, as well as with the more complex convolutional models. We report the Dice coefficient between the predicted and groundtruth mask. The results show that our transformer model significantly outperforms the OTSU and static threshold [20] methods while performing on par with the larger convolutional model. Figure 4 shows the graph of the performance of the model in relation to the number of parameters of the models. One can see that the transformer models perform on par with convolutional models while having fewer parameters. The improvement in the performance of our model compared to simple baseline models can be attributed to two main factors, first, they do have more parameters and thus have a greater capability of learning important features, but more importantly, they work on full RGB images which carry important information for color constancy, that is discarded when using only image brightness.

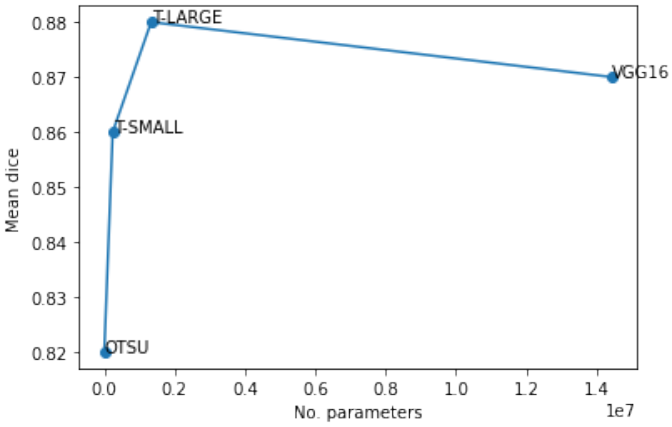


Figure 4. The relation of the performance of tested models to the number of parameters of each model.

	Segmentation Outdoor							
Model	Mean	Std	Median	Trimean	Best	Worst	Best 25	Worst 25
OTSU	0.82	0.12	0.85	0.84	0.99	0.46	0.96	0.65
THRES[20]	0.79	0.14	0.83	0.82	0.97	0.38	0.93	0.57
VGG16 UNET	0.87	0.08	0.88	0.88	0.98	0.55	0.95	0.76
T-SMALL	0.86	0.09	0.87	0.88	0.99	0.59	0.96	0.74
T-LARGE	0.88	0.09	0.90	0.89	0.99	0.60	0.96	0.76
T-LARGE PT.	0.88	0.08	0.88	0.89	0.99	0.60	0.96	0.76

Table I
Comparison of Dice coefficient results of the segmentation using proposed models and baselines. The PT. suffix indicates that the model was pretrained on the collage images.

Figure 5 shows a comparison between the output of the baseline model and the larger transformer model. The results show that the proposed transformer model is more robust and can detect the areas where the correction was applied incorrectly while being more resilient to darker colors of the objects, as can be seen on the darker patch of grass. One can also see that the borders between the classes in the predicted segmentation masks show more uncertainty than areas deep in the shaded or sunlit regions. However, even though this information is not present in the manually annotated groundtruth, there is usually some illuminant mixing in transitory areas between illuminants due to reflections of nearby objects, and one can see that the proposed model can learn this information without any supervision.

V. Conclusion

In this work, we presented a method based on the transformer architecture that can be used to detect regions of images where color correction was incorrectly applied. The usage of transformer models for image segmentation allowed us to create a model with relatively few parameters. It outperformed the baseline segmentation models by a large margin and achieved results that were on par with much larger convolutional models based on the Unet architecture with VGG16 encoder. The larger transformer architecture even slightly outperforms the convolutional models, while still having fewer parameters (See Figure 4).

Training of all of our models was done on approx. 300 images from our new multi-illuminant color constancy dataset,

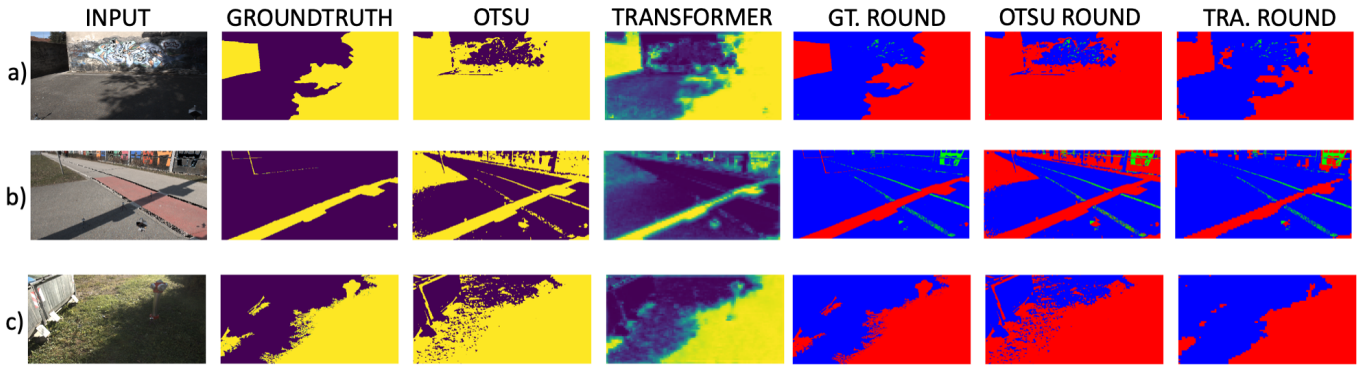


Figure 5. Comparison of the results between the groundtruth, OTSU baseline method, and the transformer model. In columns 2-4 is the comparison between the predicted probability masks that are the direct output of the transformer's sigmoid layer. Columns 5-7 show the comparison between the rounded hard segmentation masks used to compute the Dice coefficient. The green areas in the rounded masks represent clipped or underexposed pixels.

whose development is still in progress. In the future, we plan to include even more images with more than two illuminants and images with only one illuminant to make the dataset even more representative test case for both segmentation and estimation models. However, these results offer that, with the collection of more images in the dataset, even greater results could be achieved in both the task of detecting wrongly color corrected regions as well as multi-illuminant estimation tasks.

References

[1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[2] Mahmoud Afifi and Michael S Brown. What else can fool deep learning? addressing color constancy errors on deep neural network performance. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 243–252, 2019.

[3] Nikola Banic and Sven Loncaric. Unsupervised learning for color constancy. volume abs/1712.00436, pages 181–188, 2017.

[4] Jonathan T Barron. Convolutional color constancy. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 379–387, 2015.

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

[6] Shida Beigpour, Christian Riess, Joost Weijer, and Elli Angelopoulou. Multi-illuminant estimation with conditional random fields. *IEEE transactions on image processing : a publication of the IEEE Signal Processing Society*, pages 83–96, 01 2014.

[7] S. Bianco, C. Cusano, and R. Schettini. Single and multiple illuminant estimation using convolutional neural networks. *IEEE Transactions on Image Processing*, 26(9):4347–4362, 2017.

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

[9] Xiangxiang Chu, Zhi Tian, Bo Zhang, Xinlong Wang, Xiaolin Wei, Huaxia Xia, and Chunhua Shen. Conditional positional encodings for vision transformers. *arXiv preprint arXiv:2102.10882*, 2021.

[10] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *CoRR*, abs/2010.11929, 2020.

[11] Arjan Gijsenij, Theo Gevers, and Joost Van De Weijer. Generalized gamut mapping using image derivative structures for color constancy. *International Journal of Computer Vision*, 86(2-3):127–139, 2010.

[12] Kai Han, An Xiao, Enhua Wu, Jianyuan Guo, Chunjing Xu, and Yunhe Wang. Transformer in transformer. *arXiv preprint arXiv:2103.00112*, 2021.

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

[14] Marko Subašić Sven Lončarić Ilija Domislović, Donik Vršnak. Cube2: Large mutli-illuminant dataset.

https://github.com/donikv/Cube2. Accessed: 2021-06-11.

[15] H. R. V. Joze and M. S. Drew. Exemplar-based color constancy and multiple illumination. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(5):860–873, 2014.

[16] Davood Karimi, Serge Vasylechko, and Ali Gholipour. Convolution-free medical image segmentation using transformers, 2021.

[17] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017.

[18] Karlo Koščević, Marko Subašić, and Sven Lončarić. Deep learning-based illumination estimation using light source classification. *IEEE Access*, 8:84239–84247, 2020.

[19] Edwin H Land. The retinex theory of color vision. *Scientific american*, 237(6):108–129, 1977.

[20] Sang-Ho Lee, Sung-Min Woo, Ji-Hoon Choi, and Jong-Ok Kim. Two-step multi-illuminant color constancy for outdoor scenes. pages 710–714, 09 2017.

[21] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts, 2017.

[22] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. *arXiv preprint arXiv:2012.12877*, 2020.

[23] Joost Van De Weijer, Theo Gevers, and Arjan Gijsenij. Edge-based color constancy. *IEEE Transactions on image processing*, 16(9):2207–2214, 2007.

[24] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, undefinedukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6000–6010, Red Hook, NY, USA, 2017. Curran Associates Inc.

[25] J. VON KRIES. Influence of adaptation on the effects produced by luminous stimuli. *handbuch der Physiologie des Menschen*, 3:109–282, 1905.

[26] Thomas Young. *A course of lectures on natural philosophy and the mechanical arts*, volume 1. Taylor and Walton, 1845.

[27] Chen Yu, Chen Dian-ren, Li Yang, and Chen Lei. Otsu's thresholding method based on gray level-gradient two-dimensional histogram. In *Proceedings of the 2nd International Asia Conference on Informatics in Control, Automation and Robotics - Volume 3*, CAR'10, page 282–285. IEEE Press, 2010.

Publication 2

Vršnak, D., Domislović, I., Subašić M., Lončarić S., *Signal Processing: Image Communication*, Vol. 108, 2022, pp. 116822

Signal Processing: Image Communication 108 (2022) 116822

Contents lists available at ScienceDirect

Signal Processing: Image Communication

journal homepage: www.elsevier.com/locate/image

Illuminant segmentation for multi-illuminant scenes using latent illumination encoding

Donik Vršnak *, Ilija Domislović, Marko Subašić, Sven Lončarić[1]

Image Processing Laboratory, Faculty of Electrical Engineering and Computing, University of Zagreb, Unska 3, Zagreb, 10000, Croatia

ARTICLE INFO

Keywords:
Color constancy
Segmentation
Multi illuminant
Illumination estimation

ABSTRACT

Color constancy is an important part of the Human Visual System that allows us to recognize colors of object invariant to the light that is illuminating them. Computational color constancy is the process of estimating the illumination of a scene using some computational method. However, this problem is inherently ill-posed. The RGB value of each pixel is dependent on spectral reflectance of the object and the spectral power distribution of the illumination. Hence, methods that try to solve the computational color constancy problem have to introduce assumptions about the illumination. One common assumption is that there is only one global source of illumination, i.e., that the illumination is constant across the whole scene. Under this assumption, modern color constancy methods achieve excellent results, usually predicting the illumination color with accuracy better than the human eye. This assumption is broken in many real-world multi-illuminant scenes, e.g., outdoor images where parts of the scene are illuminated by either sunlight or skylight. This leads to significant drop in accuracy of single-illuminant estimation methods. Therefore, in this work, we propose a novel method for segmenting images based on illumination in multi-illuminant scenes. This method detects regions where there is only one illuminant, thus detecting areas where the single-illuminant assumption holds. We show that our method produces excellent results and outperforms all baseline models by a large margin.

1. Introduction

Color constancy is the ability of the human visual system (HVS) to adapt to illumination so that the colors of objects remains relatively constant under different illumination conditions. This allows humans to more easily recognize objects. While the HVS does this constantly in real time, replicating this functionality is difficult because the underlying neurophysiological mechanism is still not fully understood. However, when considering images, color constancy is an ill-posed problem, as the resulting chromaticity of an image is a combination of the intrinsic colors of objects and the illumination. The color value of each pixel $p_c(x,y)$ (where $c \in \{R,G,B\}$ and x,y represent the coordinates of the pixel) in an image is a function of three main factors and can be expressed as [1,2]:

$$p_c(x,y) = m_b(x,y)\int_\omega I(x,y,\lambda)R(x,y,\lambda)S_c(\lambda)d\lambda + m_c(x,y)\int_\omega I(x,y,\lambda)S_c(\lambda)d\lambda, \quad (1)$$

where ω is the visible spectrum and m_b and m_c are scale factors that model the ratio of body and specular reflectance of the light reflected from coordinates (x,y). The first part is the illumination distribution, $I(x,y,\lambda)$ where λ represents the wavelength of the illumination at (x,y) coordinates of the image. $R(x,y,\lambda)$ is the surface reflectance of objects in the scene at coordinates (x,y) and $S(\lambda)$ represents the sensitivity of the camera sensor at wavelength λ. We can simplify this equation by introducing the *Lambertian assumption*, i.e., that the specular reflectance can be ignored. With this assumption, the model for each pixel is as follows:

$$p_c(x,y) = m(x,y)\int_\omega I(x,y,\lambda)R(x,y,\lambda)S_c(\lambda)d\lambda, \quad (2)$$

where $m(x,y)$ is Lambertian shading [3]. Furthermore, we can take that the observed color of the light source $e(x,y) = \int_\omega I(x,y,\lambda)S(\lambda)d\lambda$ is dependent on both the illumination $I(x,y)$ and the sensitivity of the camera S. So, we can see that the estimation of the observed light source $e(x,y)$ from only the RGB values of the pixel $p(x,y)$ is under-constrained if we do not know the reflectance properties of the object R, which is usually the case. However, additional assumptions can be made, and an approximate solution for the color of the illumination can be acquired. Moreover, most state-of-the-art methods assume that there is only one global illuminant present in a scene. This removes the spatial component of the illumination, which greatly simplifies estimation. However, this means that multi-illuminant estimation and segmentation is still a mostly unsolved problem.

* Corresponding author.
E-mail address: donik.vrsnak@fer.hr (D. Vršnak).
[1] Served as an associate editor of the EURASIP Journal of Image and Video Processing.

https://doi.org/10.1016/j.image.2022.116822
Received 3 September 2021; Received in revised form 25 April 2022; Accepted 8 July 2022
Available online 14 July 2022

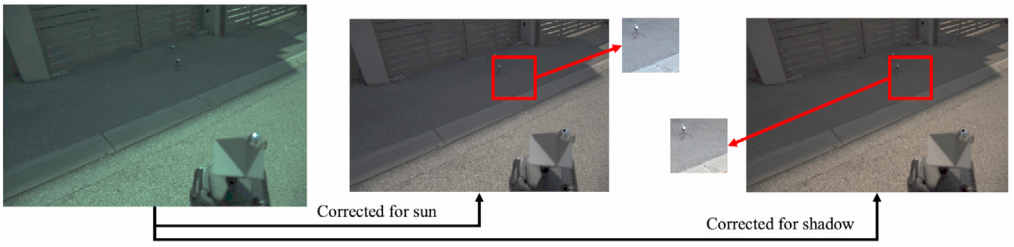


Fig. 1. Image showing the effects on the shaded parts of an outdoor image when applying correction for the sun illuminant. The top corrected image is corrected by the sun and looks natural at the first glance, but the smaller cutout shows that the shaded parts of the image appear to blue. The rightmost image is corrected using the shadow illuminant, which produces an image with unnaturally orange shade in the sun, while the color of the shaded region looks correctly reproduced.

Even though single global illuminant assumption is widely used, it does not hold true even in some common scenes, e.g., in outdoor scenes illuminated by the sun with shaded regions which are illuminated by the sky or indoor scenes where the illumination can come from multiple natural and artificial sources. Fig. 1 shows images where the single illuminant assumption is broken, and the correction done with global-illuminant estimation can lead to parts of the corrected image being incorrectly reproduced. This incorrect illuminant estimation and correction can lead to images looking unnatural to human observers and can also impact the performance of various image classification, object recognition and detection tasks [4]. This motivated us to create a method which can iteratively segment one illuminant from the rest of the image.

In this work, we propose a novel deep fully convolutional neural network image segmentation method based on scene illumination for images where at least one illuminant is known. If we assume that we can accurately estimate one illuminant in a multi-illuminant scene, which can be done using a variety of methods, we can use the proposed method to segment the image into regions that have the estimated illuminant and those that are illuminated by the unknown illuminants. In Section 2 we first take a look at current methods for single and multi-illuminant estimation and image segmentation and then, in Section 3, we continue by describing the proposed segmentation method. In Sections 4 and 5 we describe the experiments and our results on different subsets of the problem of image segmentation based on the illuminants, and in Section 6 we conclude our article.

2. Related work

Color constancy has been studied as part of cognitive and behavioral science, color science and computer science for more than 200 years, with works from [5–9] among others. Computational color constancy refers to methods used in computer vision to estimate and correct the influence of unknown illumination and retrieve original colors of the scene when it is illuminated by a canonical illuminant. As Eq. (2) shows, color constancy from RGB data is an inherently ill-posed problem, so computational color constancy methods have to impose further assumptions about the properties of some parts of the scene. The earliest such assumption was the Gray World assumption [10], which states that the average reflectance of any scene under white illumination is some shade of gray, and that any deviation from gray is caused by the illumination of the scene. Similarly, broad assumptions about the scene properties include the White Patch (Max RGB) assumption and different forms of Gray Edge framework (Eq. (3)).

$$\left(\int \left| \frac{\partial^n f_{c,\sigma}(x)}{\partial x^n} \right|^p dx \right)^{\frac{1}{p}} = k e_c^{n,p,\sigma}, \tag{3}$$

where $|\cdot|$ is the Frobenius norm, $c \in R, G, B$, n is the order of the derivative and p is the Minkowski-norm. These methods fall into the category of simple statistics methods. The next step in color constancy was the use of more complex learning based statistical models and gamut models [11,12], classical machine learning methods [13–15], generative and Bayesian models [16–20] among others. In the last decade, fast rise in computing power and the advent of deep learning networks has given rise to color constancy methods based on convolutional neural networks (CNNs, [21]). Methods described in [22] show that CNNs can outperform other statistics and learning based methods. On the other hand, [23,24] completely ignore the spatial relations of pixels and only use convolutional filters on log-chromaticity histograms to determine illuminant color. In [25] a fully convolutional network with simple attention is used to achieve state-of-the-art performance on many color constancy datasets. However, all of those methods assume that there is only one single illuminant present in the scene. In other words, they show that it is possible to very accurately predict the global illumination. The use of attention mechanism in [25] makes the model learn to ignore some parts of images which carry little information for estimation of illumination. This also means that the model can more easily focus on very accurately estimating the dominant global illuminant of multi-illuminant scenes, while discarding the information about the secondary illuminants. The main idea behind our model is to work in tandem with such state-of-the-art (SOTA) single illuminant estimation models to determine the presence of secondary illuminants, which single illuminant methods do not take into account.

There also exist some multi-illuminant estimation methods, but a lack of a large multi-illuminant dataset has slowed down the development of such methods. Hence, most proposed multi-illuminant estimation methods utilize some low level image statistics. One such method is described in [11], where they learn the properties of surfaces on train images and using those to produce illumination estimations for each surface in the test image. A multi-illuminant method based on estimating illumination for image patches and achieved improved performance compared to single illuminant estimation methods if the illuminants present in the scenes differed by more than one degree is proposed in [26]. On the other hand, in [12] a multi-illuminant estimation method based on conditional random fields (CRF) is developed. Both of these works included a small ($\lesssim 100$ images) datasets. However, both of these datasets do not contain raw unprocessed linear images, which have become the standard in modern color constancy datasets. More recently, [27] proposed a two-step illuminant estimation method for outdoor images, which was tested on 30 images that are not publicly available. Also, [28] proposed a multi-illuminant estimation model which incorporates bottom-up estimation and top-down refinement, inspired by the color constancy mechanisms of the human visual system. In order to train and test our method, we are creating a new large multi-illuminant dataset which will be publicly available. However, this dataset is still under development as we are still collecting and annotating images and the images will be changed in the future.

Image segmentation is a computer vision task where the goal is to assign a label to each pixel of the input image. Most modern state-of-the-art solutions (e.g., [29] or [30]) utilize some deep neural network consisting of an encoder and a decoder, where the encoder is used to extract features on different scales and the decoder is used to combine multi-level features into the final output. These methods are most often used in medical images and object detection tasks. As far as we know, there aren't any methods that try to do image segmentation based on illumination, like those described in this paper.

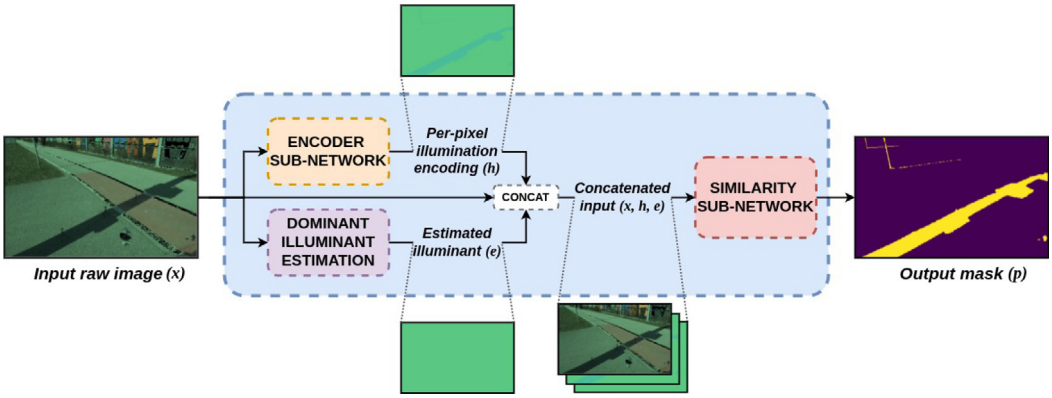


Fig. 2. Schematic representation of the proposed framework for illuminant segmentation. The input to the framework is the raw debayered image x taken directly from a camera sensor with no processing from the camera. The bottom path consists of estimating the dominant illuminant in the image (it can be considered optional, as it is not strictly part of the framework). It can be done using any method for single illuminant estimation. The top and middle paths compose the actual framework, with two identical subnetworks. The encoder subnetwork is responsible for creating the latent illumination encoding h, which is then given as the input to the similarity subnetwork, along with the original image x and the estimated illuminant e. The final output of the framework is the segmentation map $\hat{p}$ which represents the influence of the estimated (or known) dominant illuminant e on the illumination of the whole image.

3. Illuminant segmentation framework

In this work, we propose a novel segmentation framework that can be used to segment multi-illuminant images depending on local properties of illumination. We make the assumption that the dominant illuminant is known. After running experiments, the results demonstrate that the assumption that the dominant illuminant is known can be made without sacrificing real-world performance. The reason we accept the need for one illuminant is that state-of-the-art single-illuminant estimation models tend to work well even in multi-illuminant scenes, where they estimate one illuminant with a high degree of precision. In the following sections, we will present our novel illuminant segmentation framework and then present the results on our multi-illuminant dataset. We also implement several strategies and show that our approach outperforms all of them in real-world scenes.

3.1. Illuminant segmentation framework

The goal of our framework is to approximate a function $f(x; e)$ that can identify the influence of a known illuminant e on different regions of the image. We formulate this problem as a segmentation task, where the goal is to segment the input image into regions where the known illuminant e is present. If we define the output of our model as $\hat{p} = f(x; e)$, where e represents the known illuminant, and $\hat{p}$ is the per-pixel influence of that illuminant for each pixel in the input image x. In our work, the known illuminant is chosen to be the dominant illuminant in the scene, where dominant refers to the brighter illuminant, e.g., sunlight in outdoor images.

The proposed framework consists of two identical subnetworks, as is shown in Fig. 2. The first subnetwork (the encoder subnetwork) is tasked with producing a latent representation of illumination h. The dimensions of the latent representation h can be arbitrarily large, however we decided to limit it to three channels. This means that h can be considered as a per-pixel estimation of the illumination. Thus, it allows us to train the encoder in a supervised fashion, as we possess the ground truth information about per-pixel illumination. However, this representation contains errors in estimation and would, if used as is, produce an inferior reproduction of the corrected image. That is the reason for using the second subnetwork, which we call the similarity subnetwork. Its goal is to create a mask $\hat{p} = f(x, h; e)$, which is the estimated segmentation mask of the illuminants. It takes as input the original input image x, the latent estimation h (which is the direct output of the encoder subnetwork) and the known illuminant e, and determines in what amount is the known illuminant e present for each image pixel. Both subnetworks were implemented using the deep neural network architecture described in Section 3.2.

The framework, as it is proposed, would be able to perform segmentation of only two illuminants. However, real-world scenes can, in some scenarios, contain more than two sources of illumination. We make our method fully general in regard to the number of illuminants by iterative application of the segmentation framework on regions in which the illuminant of the previous iteration was not present. The iterations can be stopped either after n segmentations or automatically after more than some threshold t percent of the image has been attributed to one of the $n-1$ illuminants. The main advantage of this approach is that the number of scenes with $n = 2, 3, \ldots$ illuminants in real-world scenes decreases as n increases. However, this also poses a problem for training and evaluating such models, as there are currently no annotated images with three illuminants. We use artificially generated images to train and test our model for more than 2 illuminants.

The framework is trained in a supervised setting, as both illumination and segmentation ground truth information is available. Thus, the loss function used to optimize the parameters consists of two terms, one corresponding to the estimation h and the other segmentation $\hat{p}$. Each term is computed by taking the average mean absolute error between the predicted value and the ground truth over all pixels:

$$MAE(y^{(p)}, y^{(gt)}) = \left(\frac{1}{n * m}\right) \sum_{i=0}^{n} \sum_{j=0}^{m} \left| y_{ij}^{(p)} - y_{ij}^{(gt)} \right| \tag{4}$$

$$L = \alpha MAE(h, I_{gt}) + \beta MAE(\hat{p}, p_{gt}),$$

where (n, m) are the height and width of the image, $\hat{p}$ the segmentation output of the model and p_{gt} is the segmentation ground truth, h is the latent three channel representation and I_{gt} ground truth map of per-pixel illuminations.

3.2. Network architecture

Deep convolutional neural networks have shown as very promising in image segmentation and illuminant estimation tasks. We propose a custom novel architecture based on an FPN architecture (Feature Pyramid Network [30]), which is suitable for both illumination estimation and segmentation tasks. Thus, we are able to use this architecture for both our encoder and similarity subnetworks in our framework. This architecture is presented in Fig. 3. An FPN network usually consists of two parts, a bottom-up and top-down part, where the goal of the former is to extract features and the goal of the latter is to create an output of sufficient spatial dimensions using those features. The bottom-up part in our model consists of two branches, one with large pooling

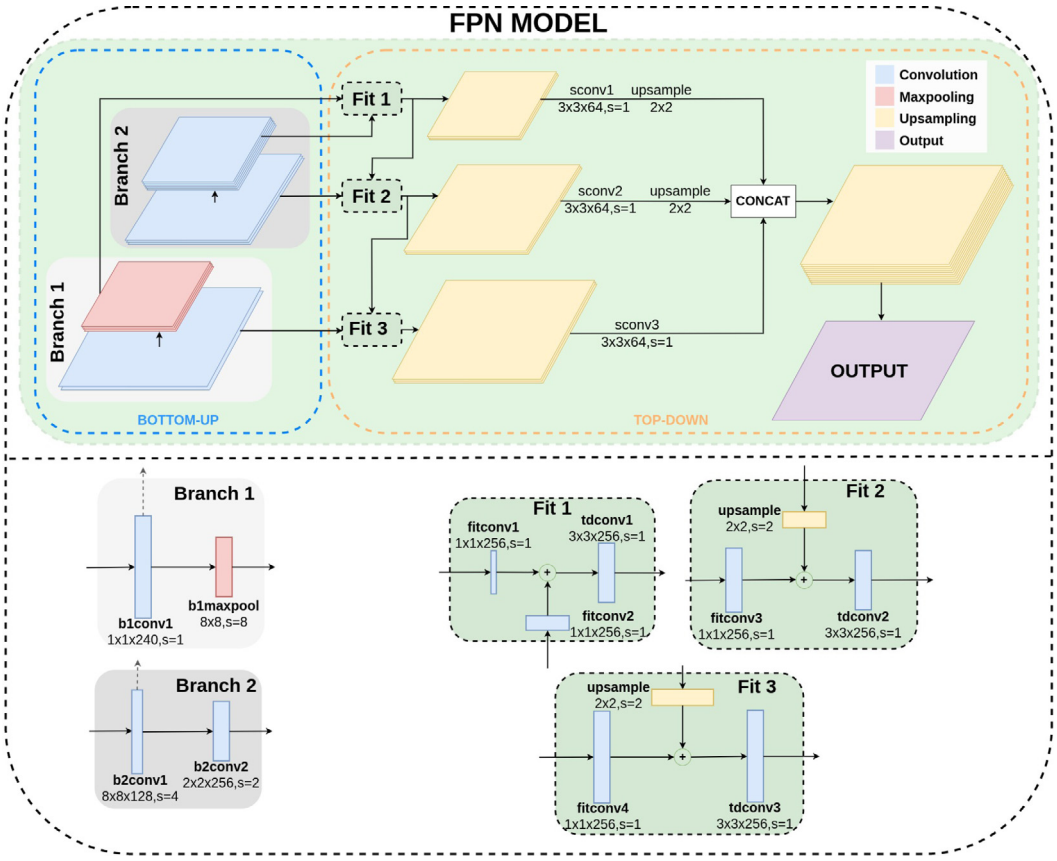


Fig. 3. Scheme showing the custom FPN model with the bottom-up part with two branches. One branch has large filters to capture more spatial information, while the smaller branch has a large pooling layer and small filters. The input is fed directly to both branches. On the scheme, blue layers represent convolutional layers followed by batch normalization and ReLU activation. Yellow layers represent the up-sampling operation and red layer represents maxpooling. Fitconv layers are used to fit the number of features of intermediary outputs of the bottom-up branches to the number of features maps which are the output of tdconv convolutional layers in the top-down part. Sconv layers are used to remove any artifacts that can appear by adding together top-down feature maps and fitted bottom-down feature maps. The final layer of the network can have different activation and number of output channels based on the task. The boxes on the bottom of the scheme show a more detailed view of the parts of the network to improve readability.

layers and convolutions with 1×1 receptive fields, while the other has convolutions with large 8×8 kernels. This allows our architecture to simultaneously attend to extraction of both local and global features without an increase in depth. As can be seen in Fig. 3, the outputs at different spatial resolutions from both branches are up-scaled and added to feature maps produced at different stages of the top-down part of the network. The final set of feature maps is produced from these intermediary outputs of the top-down part with 3×3 convolutions whose output is then up-scaled to the final resolution which is two times smaller in both height and width to the original image. The output of the network is done using a single 3×3 convolution with the appropriate number of output channels c.

Our network was built as to satisfy two important requirements for models designed for color constancy:

1. The model needs to be complex enough to detect important semantic features in images, such as faces, street signs, trees etc., which can all be useful for color constancy, as their colors do not change much from scene to scene, and thus make ideal places for illumination color of the scene to be estimated.
2. Increase in depth of neural networks has shown to decrease the performance on color constancy tasks [25]. They were primarily designed for the task of image recognition and classification, both of which benefit greatly from invariance to the color of the illumination, as those models need to, for example, correctly classify an apple whether it has been illuminated by white or orange light.

We also took into consideration that our model had to be trained from scratch, on a limited number of multi-illuminant images (compared to other modern computer vision datasets with tens of millions of images). When we combine the limited number of training examples with the constraint that most color constancy work is done on camera hardware, and therefore should not be computationally too expensive, we focused on creating an encoder that has the fewest parameters with adequate performance. Table A.6 (see Appendix A) provide more details about the implementation.

3.3. Dominant illuminant estimation

To prove that global estimation on multi-illuminant scenes can be used to provide the first known illuminant, we remove the known dominant illuminant e and replace it with an estimation from a global estimation method $\hat{e}$. The estimation was done using a custom variant of the FC4 model [25]. We replaced the AlexNet feature extractor of the original model with the bottom-up of our custom FPN architecture (see Fig. 3). This custom architecture is identical up to the b2conv2 and b1maxpool layers (see Table A.6), which are concatenated and followed by one convolutional layer with 256 filters, 5×5 kernel and stride 1×1. The final prediction was done using the two convolutional layers and an attention layer as described in [25]. This network was then pretrained for global illuminant estimation on the Cube+ [31] dataset, with further refinement on the same set of training images used for training of the segmentation framework. However, it was trained to predict only the dominant illumination $\hat{e}$ (as can be seen in the top part of the scheme in Fig. 2). After we acquired $\hat{e}$, we fed it and the image x to the segmentation framework just like we would if the illumination was known. This allows us to test the performance of our framework on real-world images where no illumination is known.

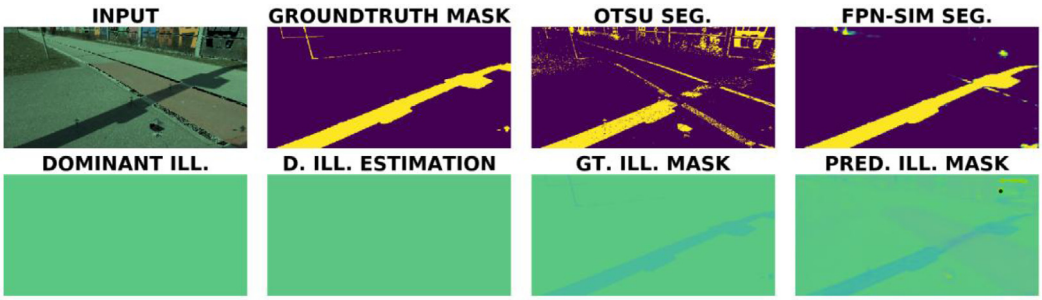


Fig. 4. Comparison of the segmentation results between the best performing baseline Otsu segmentation method and our model with estimation of the dominant illuminant (FPN-SIM+EST). The upper row shows the results of the segmentation, while the bottom row shows the comparison between the ground truth for the dominant illuminant and its estimation, as well as the comparison of the ground truth illumination mask and the output of the encoder subnetwork. We can see that the Otsu method can be fooled by darker objects which are sunlit (and vice versa), such as the grass patch in the left of the image, while our method can determine that the grass is a darker object but is illuminated by the sun, which can also be seen in illumination map output by the encoder.

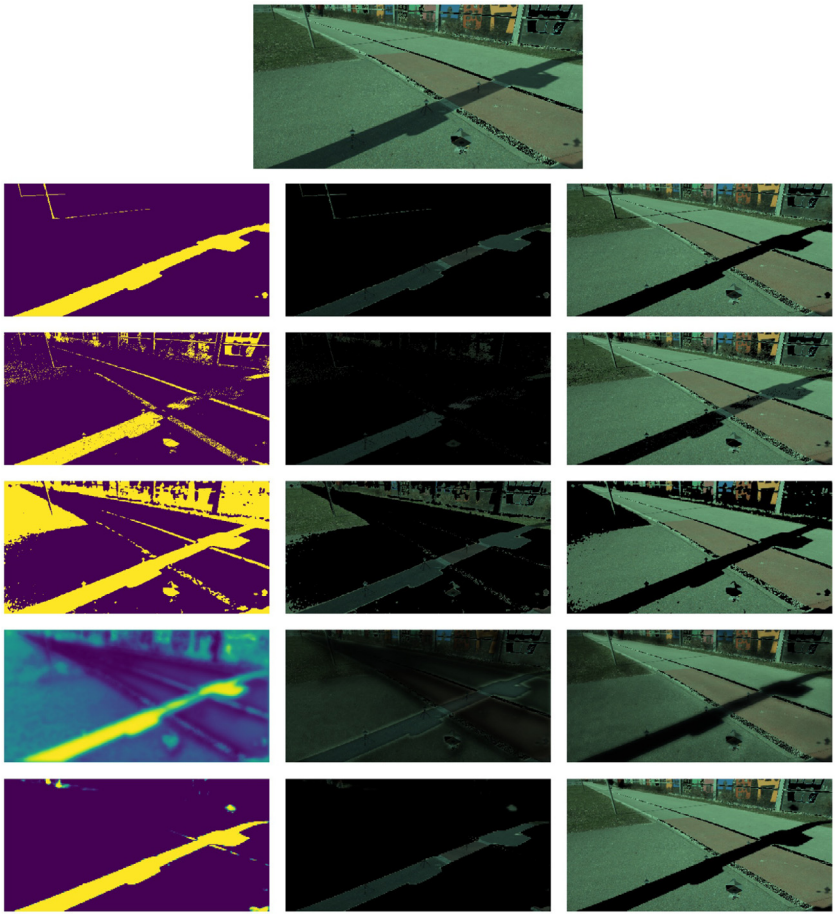


Fig. 5. Comparison of the segmentation results between all models where all illuminants are unknown (see Table 4). The left column represents the segmentation mask produced when the raw image (on the top) is given to each of the models. The middle and right columns show the regions that remain when the segmentation mask is applied by multiplying it with the raw image. The topmost row with three columns shows the ground truth, while other rows correspond to segmentation models rows in Table 4, i.e., brightness threshold, Otsu brightness threshold, SE-ResNet with dominant illuminant estimation and our framework with dominant illuminant estimation. Our framework produces the most accurate segmentation masks, and produces masks with much greater levels of confidence than the SE-ResNet model.

3.4. Baseline models

In addition to the proposed model, we implemented three classes of baseline models for comparison. We had to create those models as, to our knowledge, there is currently only one model that deals with the problem of illuminant segmentation [27]. The most simple class of baseline models was to apply a threshold on the values found in the image. We used 5 types of thresholds:

- Image brightness > some set value (as described in [27])
- Otsu threshold based on brightness
- Adaptive Gaussian threshold on brightness
- Image hue > $\frac{gt1_{hue}+gt2_{hue}}{2}$
- Otsu threshold based on hue

A more complex baseline model uses a random forest classifier on a collection of image feature patches. These features include mean, max, and median angular distances to each of the illuminants in addition to the mean, median, and max average brightness of the patch. The classifier was then used to classify whether the patch was illuminated

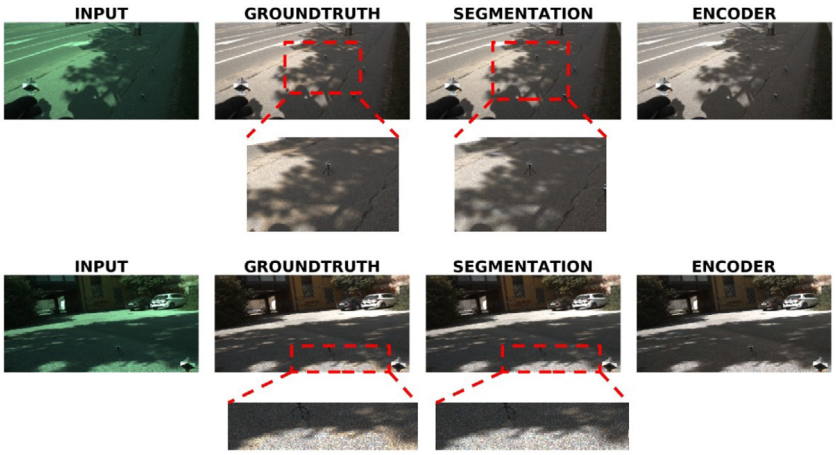


Fig. 6. Comparison of the corrected images using the ground truth, the predicted segmentation mask and the output of the encoder. The predicted mask assigned the uncertain areas, where illuminant mixing is probable due to the shadows being cast by foliage, as the dominant illuminant. This produced more visually more pleasing results than the correction using the ground truth mask, since the dominant illuminant contributed more tho the overall color of the illumination than the secondary illuminant.

using the first or the second illuminant. The third method is unsupervised clustering, which is done using Gaussian mixture models that can be performed on hue and brightness histograms of the image. Two histograms were used because hue clustering performed better on indoor images, while brightness segmentation performed better on outdoor images. In order to determine which histogram to use on different images, we used a pretrained Places365 model [32] to classify images into outdoor or indoor class, after which we applied brightness or hue histogram respectively. Hue clusters were initialized using the two hue values of the two known illuminants. Note that the second and third baseline models require that all the illuminations are known.

4. Experiments

The proposed model architecture is fully convolutional and can be trained end-to-end to predict the ground truth illuminant segmentation mask using the combined loss function as described in Section 3, Eq. (4). For our main experiment, we trained the encoder subnetwork by minimizing the proposed loss function. In this way, both the encoder and the similarity subnetworks of our framework are trained in parallel. We did this because we noticed that it was possible for the encoder, when trained separately, to converge to the mean global illumination of the image. However, we did not have this problem when the whole framework was trained in parallel. We believe that this happens because the illuminations in outdoor scenes are similar, and as such provide a smaller training signal than the 0–1 segmentation masks. The visual results shown in Fig. 5 show that the learned segmentation benefits from this approach. It can be seen that the segmentation produced by our framework is much more certain (i.e., mask values are much closer to 0–1 values) than the next best performing CNN model, which does not use any encoder subnetwork. This is inline with the ground truth, where the borders between the illuminations are hard, i.e., there is no gradual transition in the areas near the class borders. However, some images contain shadow regions produced by foliage, where illuminant mixing is present, but the manually annotated ground truth does not reflect this property. In such images, our model was able to learn to produce masks where there is either some uncertainty on the borders or, more commonly, it produced masks where the dominant illuminant was extended further into the region than the ground truth. Fig. 6 shows examples of two images showing this property. We can see that the correction done using the predicted masks produced visually more satisfying images in the uncertain areas. Additionally, as can be seen in Fig. 4, the accuracy of estimated latent illumination maps h falls short of most SOTA models for estimating illuminations (and angular error is above 3 degrees which is considered to be the limit of human perception of illumination). We then performed a series of experiments (see Section 4.2) to show the benefits of using this latent illuminant representation h, which improves performance compared to much larger models.

4.1. Training setup

The models were all implemented in TensorFlow 2.4 and trained on a system with an RTX 2080Ti GPU and AMD Ryzen 3700x CPU. Training was done using backpropagation over the parameters of the framework, 1000 epochs with cosine annealing warm restart [33] schedule for the learning rate, with range from $1 * 10^{-3}$ to $1 * 10^{-6}$ combined with Adam optimizer [34] with weight decay of $1 * 10^{-7}$, and the parameters α and β were both set to 1. All the hyperparameters were tuned once on outdoor images and used later for all experiments.

The models were tested on a variety of indoor and outdoor images, but since our dataset currently contains predominantly outdoor images, the results for outdoor-type images are the most representative of the real performance of the model. The outdoor models were trained on 310 images and validated on 77 images, and the test set contained 96 images (results are shown in Table 2). Models trained on outdoor (both nighttime and daytime images) and indoor images were trained on 367 images, validated on 60 images and tested on 63 images. For training, outdoor and nighttime images were re-sampled to somewhat mitigate the issues with image class imbalance. Because of the relatively small number of training images, we applied data augmentation during training in order to reduce overfitting. Each training image was randomly cropped, flipped and then rotated for a random angle between −15 and 15 degrees. Prior to feeding the image to the model, the brightness of the image was normalized so that $\sum_{i,j} \|p_{i,j}\|_2 = 20$, but we did not perform per channel standardization as such processing can change the colors of the image which would affect the ability of our model to learn to differentiate regions based on illumination color.

We experimented with different types of pretraining using artificially generated images with multiple illuminants, but this approach showed no significant improvement in performance of the model when tested on real images. However, because of the lack of annotated real-world images with more than two illuminants, testing for scenarios on scenes with more than two illuminants was performed on artificial collage images where three illuminants are present, in order to test the validity of our approach. Results of all segmentation models can be seen in Table 2.

Table 1
Comparison of the number of parameters of deep learning models and their computational time. The proposed architecture has fewer parameters than the more complex U-Net models, but approximately double the number of parameters compared to its variants without the similarity network. Computational times correlate with the number of parameters, except for the VGG16 model. However, this can be explained by the fact that the VGG16 model uses only basic convolutional operators (unlike the more complex Squeeze-and-Excitation layers of the SEResnet18 model [35]). Also, both the SE-ResNet and VGG models were taken from a library of segmentation models [36], and are optimized in terms of number of operations and computational complexity. On the other hand, our models were implemented without considering optimization of computational times. However, even unoptimized, our model can still be executed at 50 fps. All measurements were done on a PC with Ryzen 3700x CPU and Nvidia 2080Ti graphics card. M indicates 10^6 parameters.

Model	N. Parameters	Computation time (s)
FPN-RBF	3M	0.0136
FPN	3M	0.0135
VGG16 U-NET	19M	0.0158
SERESNET U-NET	14M	0.0231
FPN-SIM	6M	0.0201

4.2. Ablation and comparison study

In this section, we will describe the ablation study of each part of the encoder-similarity network we performed, to show that the supervised latent representation h of the illumination significantly improves the segmentation performance of our model.

In order to test how each subnetwork of our proposed framework worked, we trained and tested first the similarity subnetwork $g(x, e)$ in isolation, using only the image x and one illuminant e as input. However, to test the encoder subnetwork in isolation we needed to create a function $f(h, e)$ which would take as input the latent representation h and either one illuminant e, or all illuminants $e_1, \ldots, e_n$. For this we used an inverse quadratic RBF (Radial basis function, [37]) kernel function to map the latent illuminant estimations $h_{i,j}$ into the final segmentation mask. The RBF kernel measures the normalized distances of each latent predicted illuminant to the known illuminant e_i:

$$p_{e_i}(x;\gamma) = \frac{1}{1 + (\gamma * \|h - e_i\|_2)^2} \tag{5}$$

Those distances are then combined to represent the probability of the pixel being illuminated by the secondary illuminant (note that this approach assumes that all illuminants are known instead of just one):

$$p'(x;\gamma) = \frac{p_{e1} - p_{e2} + 1}{2} \tag{6}$$

We also decided to compare our approach with some well known networks used for image segmentation, based on the U-Net architecture [29], with encoders based on well known architectures such as VGG-16 [38] and SEResnet18 [35]. Here, we changed the input to the network by first correcting it for one of the known illuminants. The goal of the networks was to segment regions that were illuminated by the canonical illuminant from those that were illuminated by unknown illuminants. Segmentation output mask $\hat{p}(x) = \sigma(f_{ij}(x))$, where σ is the sigmoid activation function, again represents the probability that each pixel (i, j) of the input is illuminated by an unknown illuminant. Comparison of the number of parameters of each model and the computational time needed to process a single image is shown in Table 1. We present the results of all of these experiments in Section 5. All the models used for testing and comparison are shown in the appendix (Table B.7).

5. Results

Tables 2 and 3 present the results of the proposed segmentation model on only outdoor and both indoor and outdoor images, respectively, in terms of Dice coefficient. The number of illuminants known to each model is represented by the number of $*$ symbols next to the model, while -LUM and -HUE suffixes indicate that brightness and hue threshold were used, respectively. In Table 2 Best 25% and Worst 25% refer to the mean performance of the models on the 25% of the highest scoring images and 25% of the lowest scoring images, respectively. The results show that our model outperforms all other segmentation models used for comparison, managing to achieve the highest Dice coefficient score on both outdoor and combined indoor outdoor images, with means of 0.92 and 0.89 respectively. These results correspond to 12% and 14% improvement over the best performing baseline, and 2%–3% improvement over the other best performing deep model (SEResNet18), while having approx. 50% of the parameters (see Table 1 for comparison of the number of parameters of different models). We also see that there is a significant improvement in the worst 25% Dice coefficient over both the baseline models and the other proposed segmentation models, which illustrates the robustness of our method for illuminant segmentation. On the other hand, the RBF model performed much worse in the worst case scenario (indicated by the Worst 25 column in Table 2). This decrease in performance happens when two illuminants were very similar and small errors in the estimation could change the class of the output pixel, underlying the importance of a dedicated segmentation network. However, using only the dedicated segmentation network (either our custom FPN architecture or larger VGG-16 and SE-ResNet18 U-Net models) without an encoder also performed worse than when both an encoder and a similarity subnetworks were combined. For combined outdoor and indoor images, the best median results were achieved by the VGG model. We postulate that such results can be explained due to VGG model being slightly worse at adapting to a smaller set of indoor images

Table 2
Comparison of Dice coefficient results of outdoor segmentation for our model and other baseline and deep models. The first seven models are the baseline models are, while deep models are shown below them. The proposed method outperformed all other models, with a significant improvement to the worst 25% metric. This metric is very important for color constancy related tasks, as the HVS is very sensitive to even very small errors in illuminant correction. The best results are shown in bold.

	Segmentation Outdoor Images							
Model	Mean	Std	Median	Trimean	Best	Worst	Best 25%	Worst 25%
THRES-LUM	0.79	0.14	0.83	0.82	0.97	0.38	0.93	0.57
OTSU-LUM	0.81	0.12	0.83	0.83	0.98	0.49	0.94	0.64
GAT-LUM	0.67	0.08	0.69	0.68	0.80	0.45	0.74	0.56
THRES-HUE**	0.67	0.17	0.69	0.68	0.95	0.28	0.87	0.44
OTSU-HUE**	0.68	0.19	0.71	0.70	0.97	0.06	0.89	0.41
GMM**	0.65	0.13	0.65	0.65	0.92	0.33	0.81	0.48
RF**	0.75	0.25	0.88	0.81	0.97	0.07	0.95	0.37
FPN-RBF*	0.85	0.16	0.91	0.90	0.99	0.36	0.97	0.63
FPN*	0.87	0.09	0.89	0.88	0.99	0.57	0.96	0.74
VGG*	0.87	0.08	0.88	0.88	0.98	0.55	0.95	0.76
SERESNET*	0.89	0.08	0.91	0.91	0.99	0.61	0.97	0.77
FPN-SIM*	**0.92**	**0.07**	**0.94**	**0.94**	**0.99**	**0.72**	**0.98**	**0.82**

Table 3
Comparison of Dice coefficient results of outdoor and indoor segmentation for our model and other baseline and deep models. The best results are shown in bold.

	Segmentation Outdoor & Indoor							
Model	Mean	Std	Median	Trimean	Best	Worst	Best 25%	Worst 25%
THRES-LUM	0.77	0.15	0.82	0.80	0.96	0.35	0.92	0.54
OTSU-LUM	0.78	0.13	0.83	0.82	0.99	0.40	0.92	0.58
GAT-LUM	0.67	0.08	0.70	0.69	0.78	0.44	0.74	0.55
THRES-HUE**	0.69	0.18	0.70	0.71	0.95	0.18	0.89	0.44
OTSU-HUE**	0.62	0.20	0.63	0.64	0.97	0.11	0.86	0.34
VGG*	0.88	0.10	**0.92**	0.91	1.00	0.65	0.97	0.73
SERESNET*	0.81	0.10	0.80	0.81	0.98	0.60	0.93	0.68
FPN*	0.87	0.11	0.90	0.89	1.00	0.55	0.96	0.69
FPN-SIM*	**0.89**	**0.09**	0.91	**0.91**	**1.00**	**0.65**	**0.97**	**0.75**

Table 4
Comparison of dice coefficient results of outdoor segmentation using SE-ResNet and our model with known and with estimated illuminants to best performing baseline models. These results show that there is no difference in performance if the illuminant is known or if it was estimated. This validates our assumption that modern color constancy methods are able to predict one of the illuminants with adequate accuracy for our model to work reliably. The best results are shown in bold.

	Segmentation Outdoor with Illuminant Estimation							
Model	Mean	Std	Median	Trimean	Best	Worst	Best 25%	Worst 25%
THRES-LUM	0.79	0.14	0.83	0.82	0.97	0.38	0.93	0.57
OTSU-LUM	0.81	0.12	0.83	0.83	0.98	0.49	0.94	0.64
SERESNET*	0.89	0.08	0.91	0.91	0.99	0.61	0.97	0.77
FPN-SIM*	0.92	0.07	0.94	0.94	0.99	0.72	0.98	0.82
SERESNET+EST	0.89	0.08	0.91	0.91	0.99	0.61	0.97	0.77
FPN-SIM+EST	**0.92**	**0.06**	**0.94**	**0.94**	**0.99**	**0.73**	**0.98**	**0.83**

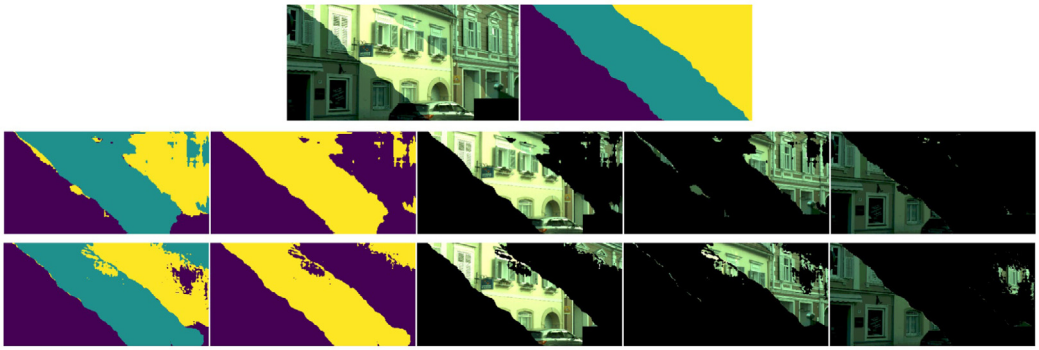


Fig. 7. Comparison of the segmentation results between two deep models on the artificially generated images with 3 illuminants. The first column represents the segmentation mask after two iterations (all known illuminants were found in the image) and the second column represents the mask after the first iteration. The third, fourth, and fifth columns show the regions that remain when the segmentation mask is applied by multiplication with the raw image (image in the first row next to the ground truth segmentation mask). In the second row is the output of SE-ResNet model, and in the third row is the output from the proposed framework.

(i.e., overfitting on outdoor images). That is why our model manages to achieve better overall performance, but the median is shifted because of the much larger number of outdoor images.

Table 4 reports the results of the experiment, where the dominant illuminant was estimated before performing segmentation. If we compare the results of the same models with known illumination and with estimation (represented by +EST suffix in Table 4), we see that not knowing the illuminant does not decrease the segmentation performance of the model. The estimation was done using the model described in Section 3.3. This network was trained separately to predict the dominant illuminant of the scene, and achieves a mean angular error of 1.41 degrees when estimating the dominant illuminant on images in the outdoor test set. This is in line with our assumption about modern color constancy methods being able to relatively accurately estimate the dominant illuminant while ignoring secondary illuminants.

Fig. 5 shows the performance of all models where no illuminants were known. Visual inspection of the results shows that our proposed framework creates the most accurate segmentation maps. If we compare the output of the framework and the output of the SE-ResNet model, we can see that our model is also much more confident in its prediction, as noted by the much larger areas where the output is clamped close to 0–1 values (see the last two images in the first column in Fig. 5). This shows greater certainty of our model in areas where there is relatively little mixing of the illuminants. However, if we take a look at additional images and segmentation results that can be found in Appendix C, we can see that for some images (see Figs. C.10(c), C.11(b), C.11(d), C.13(b)) the predicted border between illuminants has a more gradual change between regions. This shows that our model manages to somewhat learn about the mixing property of illumination, even though it is not present in the original ground truth information (which all contain only hard borders). We postulate that the model learned this property because MAE loss function was used during training, which is more commonly associated with regression tasks, and thus encouraged the model to keep parts of the output uncertain. Another mechanism that the model learned in these regions was to extend the dominant illuminant to the mixing regions (see Figs. C.12, C.12(d), C.13(c)), which produces more visually pleasing results than the correction using the ground truth mask (see Fig. 6). This property also makes sense, as

Table 5
Comparison of Dice coefficient results of the three illuminant segmentation problem for our model and other baseline and SE-ResNet model and baseline clustering models. The best results are shown in bold.

	Segmentation Three Illuminants							
Model	Mean	Std	Median	Trimean	Best	Worst	Best 25%	Worst 25%
THRES-CLUSTER***	0.30	0.12	0.30	0.30	0.75	0.04	0.46	0.16
THRES-HUE**	0.30	0.18	0.28	0.28	0.91	0.01	0.53	0.09
SERESNET**	0.67	0.17	0.66	0.67	**1.00**	**0.27**	**0.90**	**0.45**
FPN-SIM**	**0.69**	**0.16**	**0.70**	**0.70**	0.98	0.19	0.88	0.44

Table A.6
Brief description of the custom FPN model used. Each Convolutional entry (except the last one) represents a convolutional layer, followed by batch normalization and ReLU activation. Multiple inputs into a layer are concatenated before being passed to the layer.

FPN Model Description			
Layer	Type	Parameters	Inputs
input	Image	dim = (256, 512, variable)	–
b1conv1	Convolution	f = 240, k = (1, 1), s = 1	input
b1maxpool	Maxpooling	k = (8, 8), s = 8	b1conv1
b2conv1	Convolution	f = 128, k = (8, 8), s = 4	input
b2conv2	Convolution	f = 256, k = (4, 4), s = 2	b2conv1
fitconv1	Convolution	f = 256, k = (1, 1), s = 1	b1maxpool
fitconv2	Convolution	f = 256, k = (1, 1), s = 1	b2conv2
fitconv3	Convolution	f = 256, k = (1, 1), s = 1	b2conv1
fitconv4	Convolution	f = 256, k = (8, 8), s = 2	input
tdadd1	Add	–	fitconv1, fitconv2
tdconv1	Convolution	f = 256, k = (3, 3), s = 1	tdadd1
upsample1	Upsampling	size = (2, 2)	tdconv1
tdadd2	Add	–	upsample1, fitconv3
tdconv2	Convolution	f = 256, k = (3, 3), s = 1	tdadd2
upsample2	Upsampling	size = (2, 2)	tdconv2
tdadd3	Add	–	upsample2, fitconv4
tdconv3	Convolution	f = 256, k = (3, 3), s = 1	tdadd3
sconv1	Convolution	f = 64, k = (3, 3), s = 1	tdconv1
supsample1	Upsampling	size = (4, 4)	sconv1
sconv2	Convolution	f = 64, k = (3, 3), s = 1	tdconv2
supsample2	Upsampling	size = (2, 2)	sconv2
sconv3	Convolution	f = 64, k = (3, 3), s = 1	tdconv3
seg	Convolution	f = variable, k = (3, 3), s = 1	supsample1,supsample2,sconv3

Table B.7
Description and usage of all models used for comparison and evaluation. The number of known illuminants needed for each method is shown in the last column.

Model	Description	Usage	N. Illuminants
THRES	Fixed luminance/chromaticity based threshold	Comparison with a non learning based method	0
GAT	Gaussian luminance/chromaticity threshold	Comparison with a non learning based method	0
OTSU	Otsu luminance/chromaticity threshold	Comparison with a non learning based method	0
RF	Random forest patch based model using angular distance from the know illuminants	Comparison with a simpler machine learning approach	2
GMM	Histogram clustering model based on Gaussian mixture model	Comparison with an unsupervised learning method	2
FPN-SIM	Framework with FPN architecture encoder and similarity networks	The proposed model	1
FPN	FPN similarity architecture used on images corrected for one illuminant	Ablative comparison to a model without the encoder	1
FPN-RBF	FPN (encoder sub-network) with RBF layer instead of the similarity network	Ablative comparison to a model without the similarity network	2
SERESNET	Unet VGG16 segmentation architecture used on images corrected for one illuminant	Comparison with well known segmentation architecture	1

(*continued on next page*)

Table B.7 (*continued*).

Model	Description	Usage	N. Illuminants
VGG	Unet SqueezeNet ResNet 18 architecture used on images corrected for one illuminant	Comparison with well known segmentation architecture	1
FPN-SIM + EST	Framework with FPN architecture encoder and similarity networks with dominant illuminant estimation	Determining the impact of estimating the dominant illuminant on segmentation accuracy	0
SERESNET + EST	SqueezeNet ResNet 18 architecture with dominant illuminant estimation	Determining the impact of estimating the dominant illuminant on segmentation accuracy	0

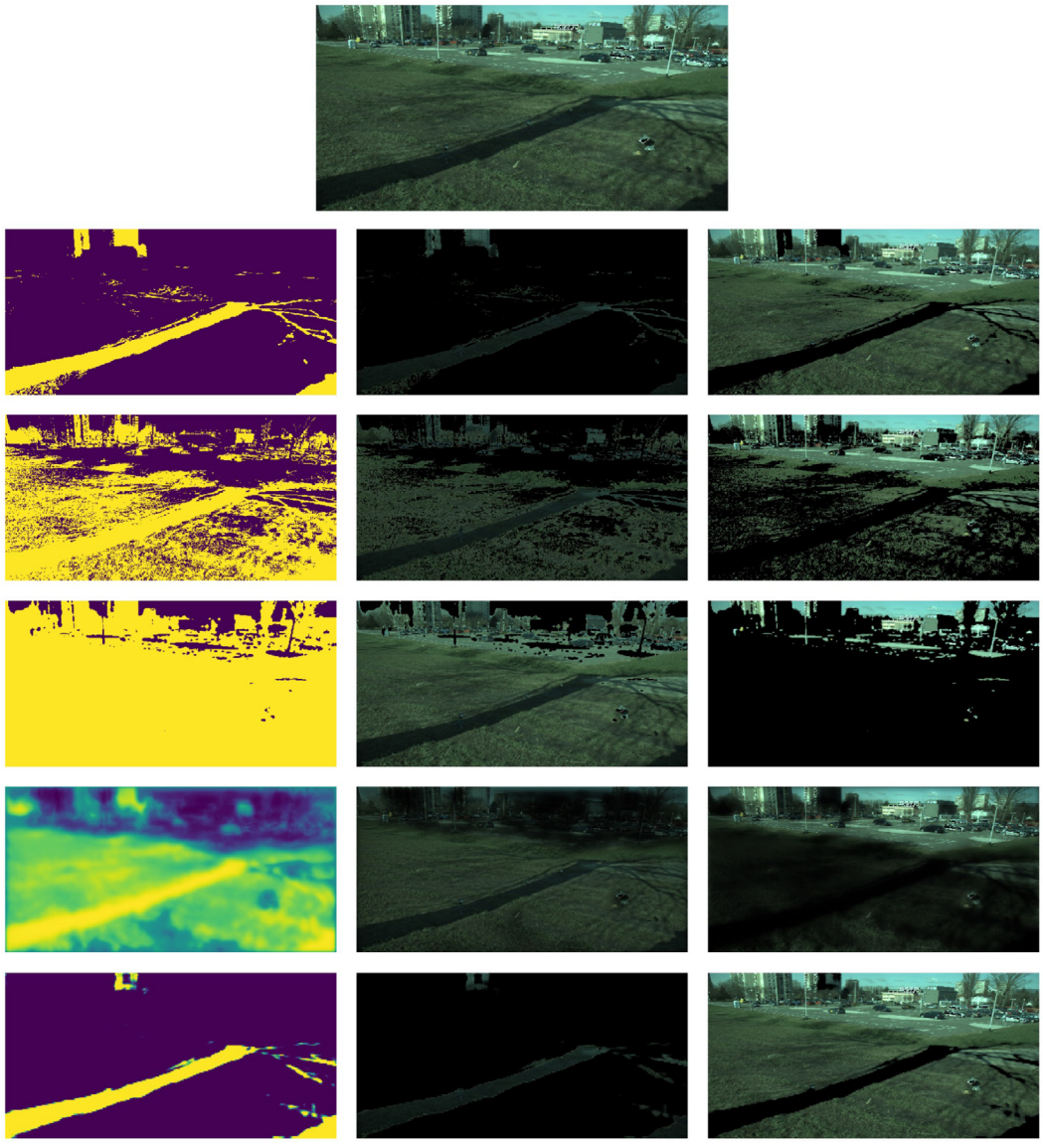


Fig. C.8. Visual comparison of the output, the ground truth segmentation, threshold, Otsu, SE-ResNet and our proposed framework.

the dominant illuminant contributed more to the overall illumination in those areas. Nevertheless, our method was trained and works best on images where the mixing of the illuminants is minimal (e.g., outdoor scenes during both day and night, or indoor scenes with a clearly defined border between illuminants), and will need to be retrained in order to fully encapsulate the problem of illuminant mixing.

Table 5 shows a comparison of our model with baseline models and the SEResNet18 segmentation model on the artificially generated set of images with three illuminations. We can see that the results show lower dice coefficient than on the two illuminant segmentation task. However, both deep models still significantly outperform the baseline models. The quantitative and visual results seem to show that the main

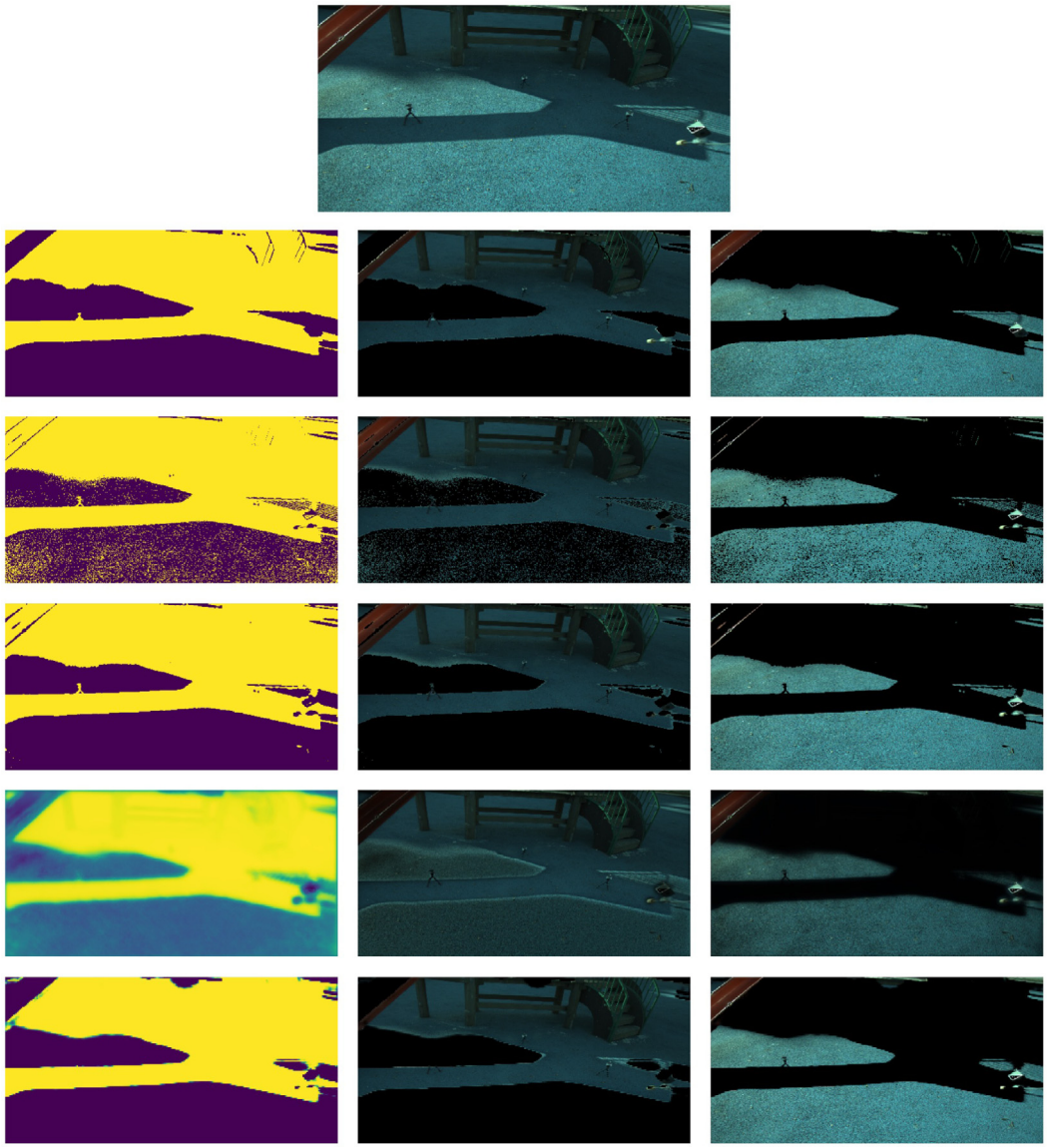


Fig. C.9. Visual comparison of the output, the ground truth segmentation, threshold, Otsu, SE-ResNet and our proposed framework.

weakness of our algorithm on the artificially illuminated images is that the models lack the capacity to discriminate between very similar illuminant colors used in artificial images (see Fig. 7, C.14 and C.15). Also, in the artificial images the illuminants are independent of the other properties of the scene, and as such require more capacity to represent both semantic and illuminant features. We can also see that both the SE-ResNet model and our framework perform similarly, but with overall lower performance compared to real images, which further points to artificial image segmentation being a harder problem. We can also see that the SE-ResNet model performs slightly better in the best and worst case scenario. We believe that this happens due to the larger number of parameters of that model, which allows it to learn a wider variety of scenes, which is important in artificial images due to the before stated independence of illuminations and scene properties.

6. Conclusion

In this work, we have shown a novel framework capable of segmenting images based on illumination. The framework was based on the idea of approximately encoding per-pixel illuminations of the image can augment the performance of the segmentation based on illumination. We tested our proposed framework on outdoor and indoor images with two illuminants against five different baseline models and three different deep learning models to test the validity of our approach since, to our knowledge, there is only one other model [27] that does explicit illuminant segmentation. All the proposed deep models assume that at least one illuminant is known. In Section 5 we showed that the proposed model outperforms all baseline and deep-learning models on outdoor and indoor images. We also show that the models work best when evaluated on outdoor images with well-defined borders, because the dataset currently contains more outdoor images than indoor ones. Even without knowing the illuminations, the results can be obtained by estimation. Besides, the performance is not decreased when combined with global estimation. This validates our claim that SOTA illuminant estimation methods work well on estimating only the dominant illuminant and can be used in conjunction with our models. We also proposed an iterative method which extends our segmentation model to perform the segmentation on scenes with more than two illuminants by iterative segmentation of regions with known illumination from those where the illumination is not known. We tested the iterative approach on artificial images, as we currently do not have any real annotated images with more than two illuminants.

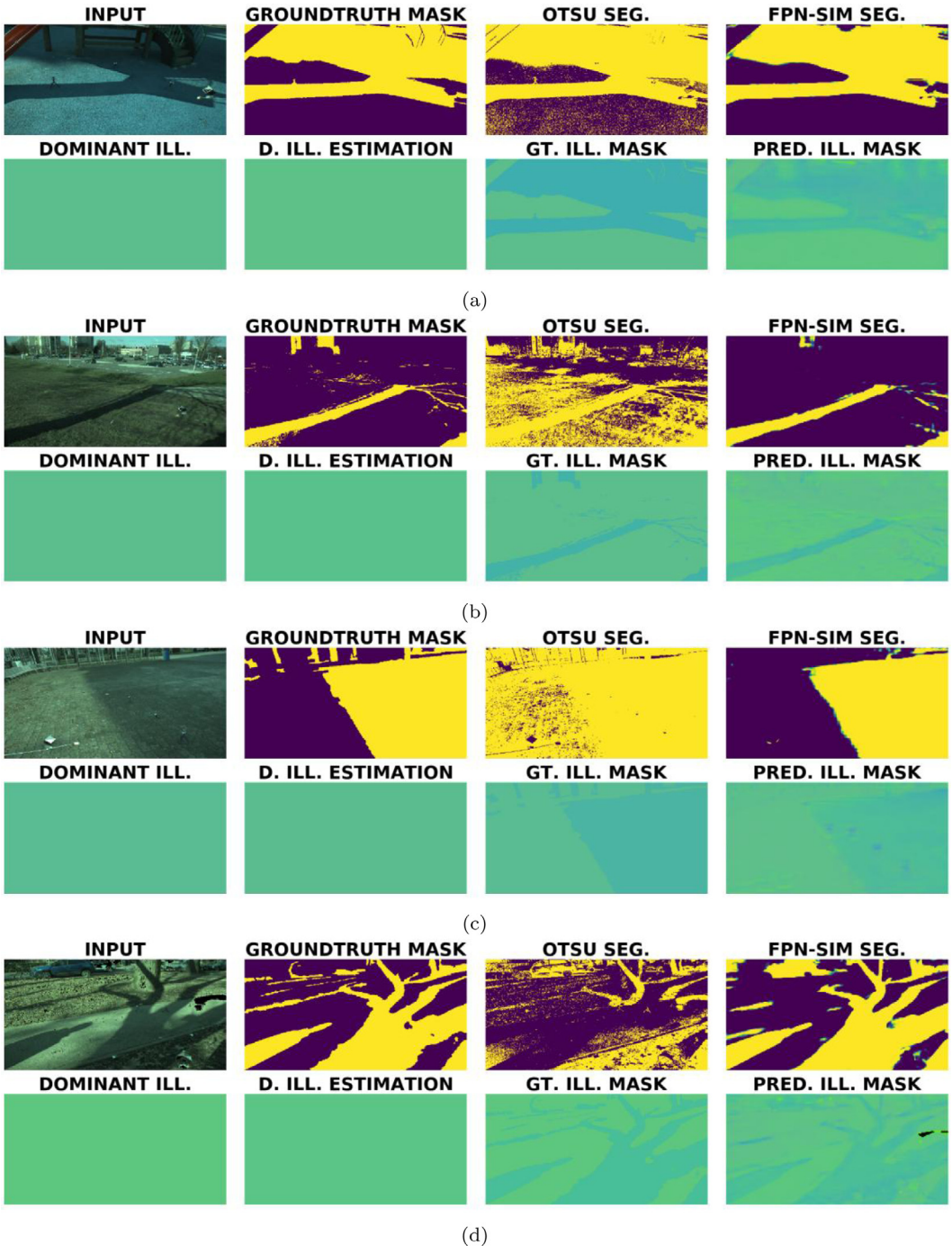


Fig. C.10. Additional results in the same format as Fig. 4.

CRediT authorship contribution statement

Donik Vršnak: Conceptualization, Methodology, Software, Formal analysis, Investigation, Writing – original draft, Visualization. **Ilija Domislović:** Data Curation, Validation. **Marko Subašić:** Writing – review & editing, Conceptualization. **Sven Lončarić:** Resources, Project administration, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix A. Model description

See Table A.6.

Appendix B. Models

See Table B.7.

Appendix C. Results

See Figs. C.8–C.15.

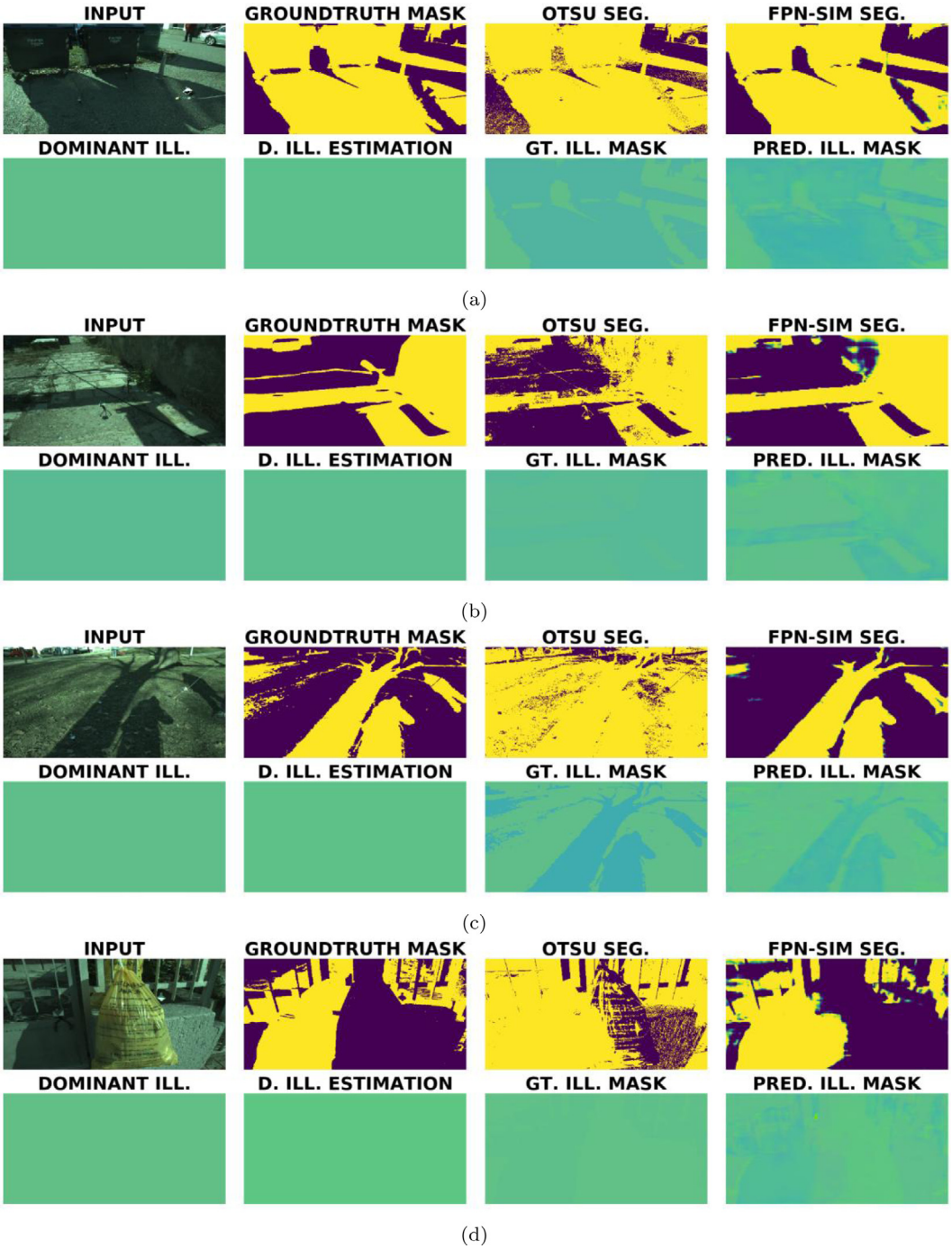


Fig. C.11. Additional results in the same format as Fig. 4.

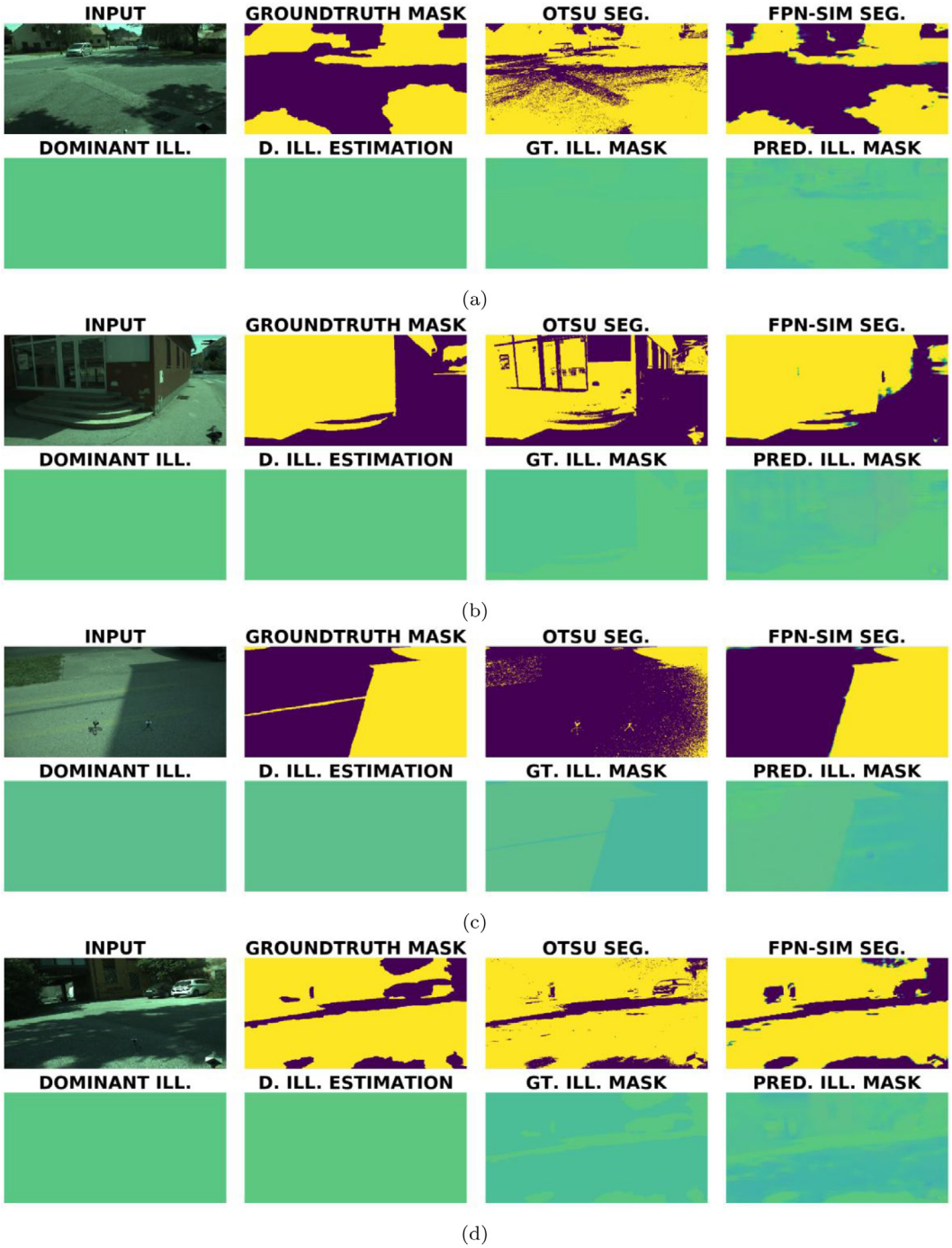


Fig. C.12. Additional results in the same format as Fig. 4.

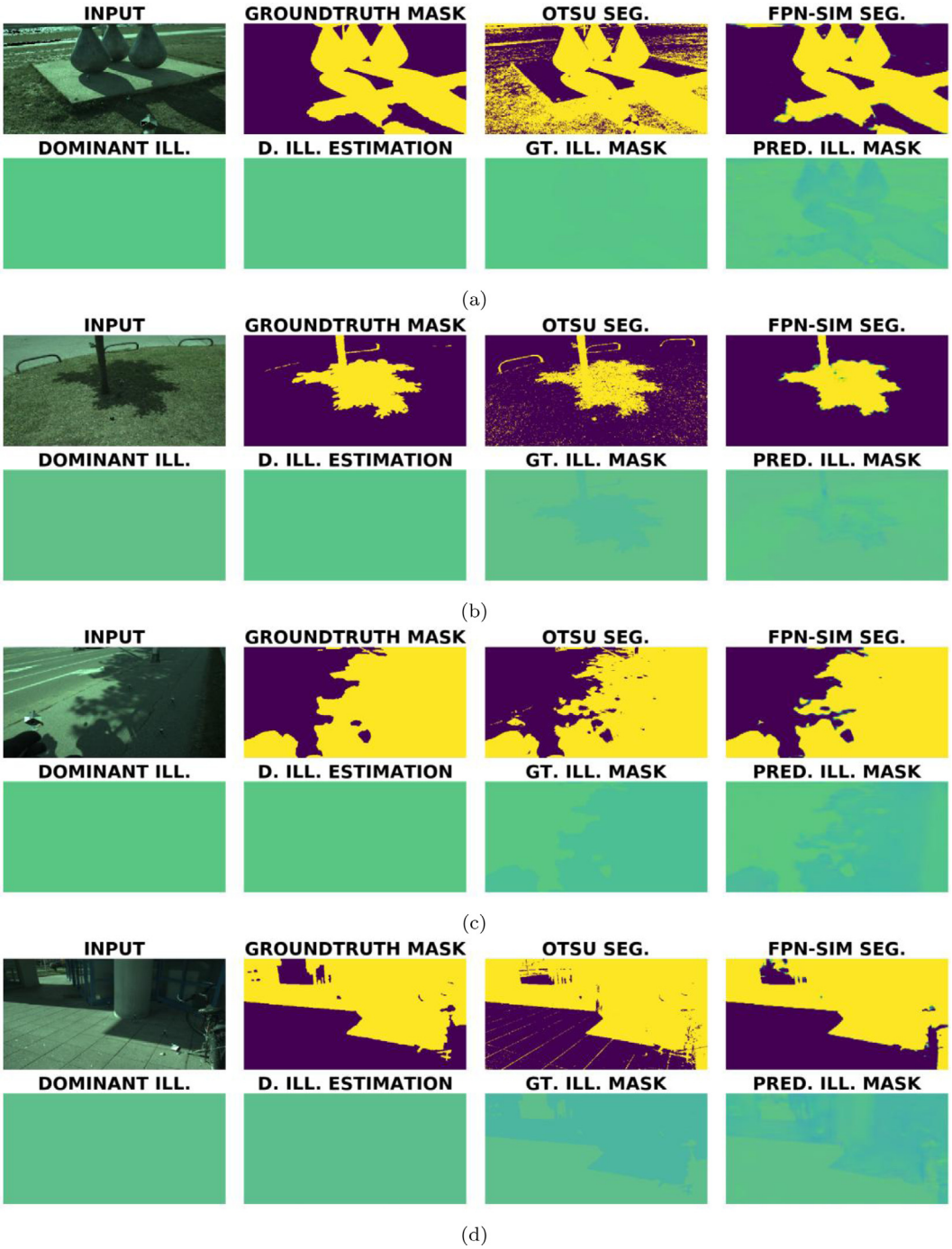


Fig. C.13. Additional results in the same format as Fig. 4.

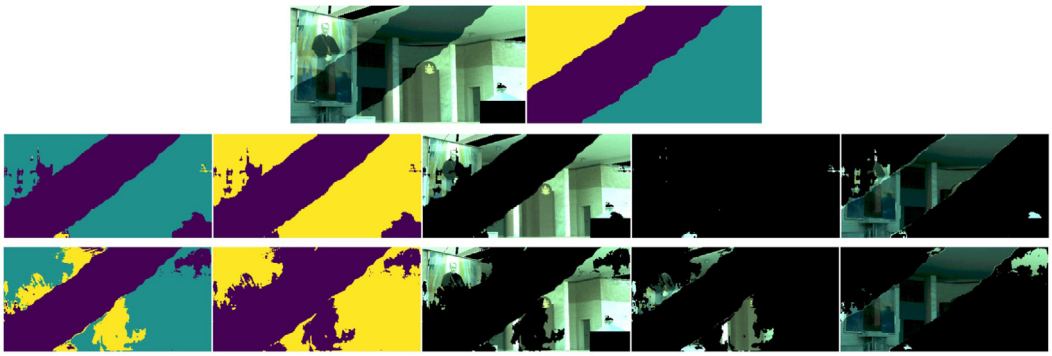


Fig. C.14. Visual comparison of the proposed framework and the SERESNET model on artificial images with 3 illuminants. This image shows the failure case in which the final mask is almost identical to the mask after the first iteration (first and second column). It can be seen that here the models could not distinguish between the illuminant on the top left and the illuminant in the bottom right part of the scene, which is why the image in the fourth column is almost completely black in both rows.

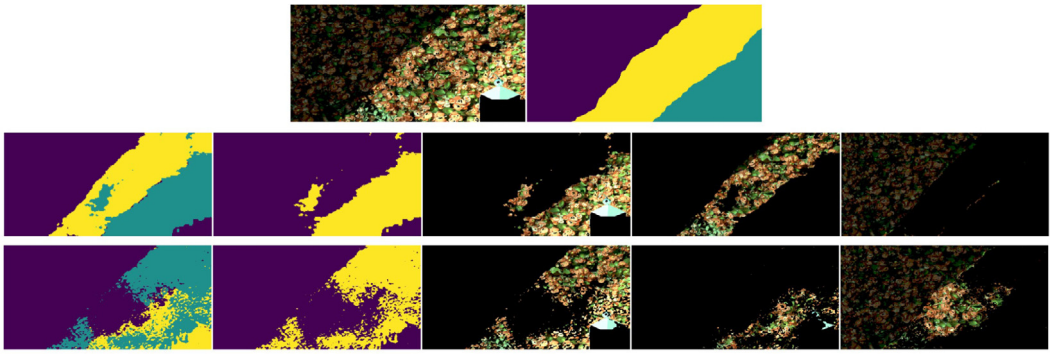


Fig. C.15. Visual comparison of the proposed framework and the SERESNET model on artificial images with 3 illuminants.

References

[1] Steven A. Shafer, Using color to separate reflection components, Color Res. Appl. 10 (4) (1985) 210–218.

[2] Gudrun Klinker, Steven Shafer, Takeo Kanade, A physical approach to color image understanding, Int. J. Comput. Vis. 4 (2004) http://dx.doi.org/10.1007/BF00137441.

[3] Sanjeev J. Koppal, Lambertian reflectance, in: Computer Vision: A Reference Guide, Springer US, Boston, MA, 2014, pp. 441–443, http://dx.doi.org/10.1007/978-0-387-31439-6_534.

[4] Mahmoud Afifi, Michael S. Brown, What else can fool deep learning? Addressing color constancy errors on deep neural network performance, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 243–252.

[5] G. Monge, Mémoire sur quelques phénomènes de la vision, Ann. Chim. 3 (1789) 131–147.

[6] Thomas Young, A Course of Lectures on Natural Philosophy and the Mechanical Arts, vol. 1, Taylor and Walton, 1845.

[7] J. Von Kries, Influence of adaptation on the effects produced by luminous stimuli, Handbuch Phys. Menschen 3 (1905) 109–282, URL https://ci.nii.ac.jp/naid/10030415665/en/.

[8] Deane B. Judd, Hue saturation and lightness of surface colors with chromatic illumination, JOSA 30 (1) (1940) 2–32.

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

[10] A. Gijsenij, T. Gevers, J. van de Weijer, Computational color constancy: Survey and experiments, IEEE Trans. Image Process. 20 (9) (2011) 2475–2489.

[11] H.R.V. Joze, M.S. Drew, Exemplar-based color constancy and multiple illumination, IEEE Trans. Pattern Anal. Mach. Intell. 36 (5) (2014) 860–873.

[12] Shida Beigpour, Christian Riess, Joost Weijer, Elli Angelopoulou, Multi-illuminant estimation with conditional random fields, IEEE Trans. Image Process. (2014) 83–96, http://dx.doi.org/10.1109/TIP.2013.2286327.

[13] Brian Funt, Weihua Xiong, Estimating illumination chromaticity via support vector regression, 50, 2004, pp. 47–52, http://dx.doi.org/10.2352/J.ImagingSci.Technol.(2006)50:4(341),

[14] Vivek Agarwal, Andrei V. Gribok, Mongi A. Abidi, Machine learning approach to color constancy, Neural Netw. 20 (5) (2007) 559–563, http://dx.doi.org/10.1016/j.neunet.2007.02.004, URL https://www.sciencedirect.com/science/article/pii/S0893608007000846.

[15] Vivek Agarwal, Andrei V. Gribok, Andreas Koschan, Mongi A. Abidi, Estimating illumination chromaticity via kernel regression, in: 2006 International Conference on Image Processing, 2006, pp. 981–984, http://dx.doi.org/10.1109/ICIP.2006.312652.

[16] Peter Vincent Gehler, Carsten Rother, Andrew Blake, Tom Minka, Toby Sharp, BayesIan color constancy revisited, in: 2008 IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2008, pp. 1–8.

[17] David H. Brainard, William T. Freeman, BayesIan color constancy, J. Opt. Soc. Amer. A 14 (7) (1997) 1393–1411.

[18] Charles Rosenberg, Alok Ladsariya, Tom Minka, BayesIan color constancy with non-gaussian models, Adv. Neural Inf. Process. Syst. 16 (2003) 1595–1602.

[19] Firas Laakom, Jenni Raitoharju, Alexandros Iosifidis, Uygar Tuna, Jarno Nikkanen, Moncef Gabbouj, Probabilistic color constancy, 2020, CoRR abs/2005.02730 arXiv:2005.02730.

[20] G. Sapiro, Color and illuminant voting, IEEE Trans. Pattern Anal. Mach. Intell. 21 (11) (1999) 1210–1215, http://dx.doi.org/10.1109/34.809114.

[21] Ian Goodfellow, Yoshua Bengio, Aaron Courville, Deep Learning, MIT Press, 2016, http://www.deeplearningbook.org.

[22] S. Bianco, C. Cusano, R. Schettini, Single and multiple illuminant estimation using convolutional neural networks, IEEE Trans. Image Process. 26 (9) (2017) 4347–4362.

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

[24] Jonathan T. Barron, Convolutional color constancy, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 379–387.

[25] Yuanming Hu, Baoyuan Wang, Stephen Lin, Fc4: Fully convolutional color constancy with confidence-weighted pooling, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 4085–4094.
[26] A. Gijsenij, R. Lu, T. Gevers, Color constancy for multiple light sources, IEEE Trans. Image Process. 21 (2) (2012) 697–707.
[27] Sang-Ho Lee, Sung-Min Woo, Ji-Hoon Choi, Jong-Ok Kim, Two-step multi-illuminant color constancy for outdoor scenes, 2017, pp. 710–714, http://dx.doi.org/10.1109/ICIP.2017.8296373.
[28] Shao-Bing Gao, Yan-Ze Ren, Ming Zhang, Yong-Jie Li, Combining bottom-up and top-down visual mechanisms for color constancy under varying illumination, IEEE Trans. Image Process. 28 (9) (2019) 4387–4400.
[29] Olaf Ronneberger, Philipp Fischer, Thomas Brox, U-Net: Convolutional Networks for Biomedical Image Segmentation, in: LNCS, vol. 9351, 2015, pp. 234–241, http://dx.doi.org/10.1007/978-3-319-24574-4_28.
[30] T. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, S. Belongie, Feature pyramid networks for object detection, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2017, pp. 936–944.
[31] Nikola Banic, Sven Loncaric, Unsupervised learning for color constancy, 2017, pp. 181–188, CoRR abs/1712.00436 arXiv:1712.00436.
[32] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, Antonio Torralba, Places: A 10 million image database for scene recognition, IEEE Trans. Pattern Anal. Mach. Intell. (2017).
[33] Ilya Loshchilov, Frank Hutter, SGDR: Stochastic gradient descent with warm restarts, 2017, arXiv:1608.03983.
[34] Diederik P. Kingma, Jimmy Ba, Adam: A method for stochastic optimization, 2017, arXiv:1412.6980.
[35] Jie Hu, Li Shen, Samuel Albanie, Gang Sun, Enhua Wu, Squeeze-and-excitation networks, 2019, arXiv:1709.01507.
[36] Pavel Yakubovskiy, Segmentation models, 2019, GitHub Repository GitHub https://github.com/qubvel/segmentation_models.
[37] Mark J.L. Orr, et al., Introduction to Radial Basis Function Networks, Technical Report, center for cognitive science, University of Edinburgh, 1996.
[38] Karen Simonyan, Andrew Zisserman, Very deep convolutional networks for large-scale image recognition, 2015, arXiv:1409.1556.

Donik Vršnak (Graduate Student Member, IEEE) received his B.Sc. in computer science in 2018 and M.Sc. also in computer science in 2020. He is currently a first-year doctoral student in the field of computing at the Faculty of Electrical Engineering and Computing, University of Zagreb, Croatia. The research he is most interested in are image processing and analysis, bioinformatics, and machine learning. Research he is currently doing is related to color constancy, focused primarily on multi-illuminant segmentation.

Ilija Domislović (Graduate Student Member, IEEE) received the B.Sc. and M.Sc. degrees in computer science in 2018 and 2020, respectively He is currently in his first year of the technical sciences in the scientific field of computing Ph.D. program with the Faculty of Electrical Engineering and Computing, University of Zagreb, Croatia. His research interests include image processing, image analysis, and neural networks. His current research is in the area of color constancy, with a focus on illumination estimation.

Marko Subašić (Member, IEEE) received the Ph.D. degree from the Faculty of Electrical Engineering and Computing, University of Zagreb, in 2007. Since 1999, he has been working with the Department for Electronic Systems and Information Processing, Faculty of Electrical Engineering and Computing, University of Zagreb, where he is currently an Associate Professor. He teaches several courses at the graduate and undergraduate levels. His research interests include image processing and analysis and neural networks, with a particular interest in image segmentation, detection techniques, and deep learning. He is also a member of the IEEE Computer Society, the Croatian Center for Computer Vision, the Croatian Society for Biomedical Engineering and Medical Physics, and the Centre of Research Excellence for Data Science and Advanced Cooperative Systems.

Sven Lončarić (Senior Member, IEEE) received the Ph.D. degree in electrical engineering from the University of Cincinnati, Cincinnati, OH, USA, in 1994, as a Fulbright Scholar. He was an Assistant Professor with the New Jersey Institute of Technology, Newark, NJ, USA, from 2001 to 2003. He is currently a Professor of Electrical Engineering and Computer Science at the Faculty of Electrical Engineering and Computing, University of Zagreb, Croatia. He was the principal investigator on a number of R&D projects. He is the Director of the Center for Computer Vision, University of Zagreb and the Head of the Image Processing Group. He is a Co-Director of the Center of Excellence in Data Science and Cooperative Systems. He has coauthored more than 250 publications in scientific journals and conferences. His research interests include image processing and computer vision. He was the Chair of the IEEE Croatia Section. He is a member of the Croatian Academy of Technical Sciences. He received several awards for his scientific and professional work.

Publication 3

Vršnak, D, Domislović, I., Subašić M., Lončarić S., *Journal of the Optical Society of America A*, Vol. 39, 2022, pp. 1076-1084

Journal of the Optical Society of America A
OPTICS, IMAGE SCIENCE, AND VISION

Autoencoder-based training for multi-illuminant color constancy

DONIK VRŠNAK,* ILIJA DOMISLOVIĆ, MARKO SUBAŠIĆ, AND SVEN LONČARIĆ

Image Processing Laboratory, Faculty of Electrical Engineering and Computing, University of Zagreb, Zagreb, Croatia
**Corresponding author: donik.vrsnak@fer.hr*

Received 7 March 2022; revised 6 May 2022; accepted 6 May 2022; posted 10 May 2022; published 25 May 2022

Color constancy is an essential component of the human visual system. It enables us to discern the color of objects invariant to the illumination that is present. This ability is difficult to reproduce in software, as the underlying problem is ill posed, i.e., for each pixel in the image, we know only the RGB values, which are a product of the spectral characteristics of the illumination and the reflectance of objects, as well as the sensitivity of the sensor. To combat this, additional assumptions about the scene have to be made. These assumptions can be either handcrafted or learned using some deep learning technique. Nonetheless, they mostly work only for single illuminant images. In this work, we propose a method for learning these assumptions for multi-illuminant scenes using an autoencoder trained to reconstruct the original image by splitting it into its illumination and reflectance components. We then show that the estimation can be used as is or can be used alongside a clustering method to create a segmentation map of illuminations. We show that our method performs the best out of all tested methods in multi-illuminant scenes while being completely invariant to the number of illuminants.

https://doi.org/10.1364/JOSAA.457751

1. INTRODUCTION

Color constancy allows us to perceive the colors of objects invariant to the colors of the lights illuminating them. To mimic this behavior, modern digital cameras have to emulate this process using image processing. However, unlike the human visual system, which can perform this operation in real time, the RGB values taken from the camera sensor provide little information about the nature of the illumination and the underlying scene. The captured pixels p_c, $c \in \{R, G, B\}$ are the product of three properties: reflectance of objects $R(\lambda)$, the spectrum of illumination $I(\lambda)$, and the spectral characteristic of the sensor $S_c(\lambda)$ {Eq. (1), [1,2]}. For real-world scenes, neither $I(\lambda)$ nor $R(\lambda)$ is known. Hence, the problem of color constancy is inherently ill posed, as there is an infinite number of combinations of surface reflectance and illumination that could produce the captured pixel values:

$$p_c(x, y) = \int_\omega I(x, y, \lambda) R(x, y, \lambda) S_c(\lambda) \mathrm{d}\lambda. \tag{1}$$

Computational color constancy can be separated into two sub-tasks. The first task is illumination estimation, which is the more difficult of the two, since it is constrained by the ill-posed nature of the problem. The second task is color correction, whose goal is to retrieve the canonically illuminated image using the original image and the estimated illumination. The correction is done using the von Kries model [3]. This operation can be written in matrix form as follows:

$$\begin{bmatrix} c_R \\ c_B \\ c_G \end{bmatrix} = \begin{bmatrix} d_R & 0 & 0 \\ 0 & d_B & 0 \\ 0 & 0 & d_G \end{bmatrix} \begin{bmatrix} p_R \\ p_B \\ p_G \end{bmatrix}, \tag{2}$$

where $[c_R\ c_B\ c_G]^T$ is the corrected image pixel, $[p_R\ p_B\ p_G]^T$ is the original image pixel acquired from the sensor, and $[d_R\ d_B\ d_G]^T$ represents the correction factor from the unknown estimated illuminant $\boldsymbol{e}_u$ to the canonical illuminant $\boldsymbol{e}_c$ (i.e., white, $e_c = \left[\frac{1}{3}\ \frac{1}{3}\ \frac{1}{3}\right]$) and can be calculated as

$$\begin{bmatrix} d_R \\ d_B \\ d_G \end{bmatrix} = \begin{bmatrix} e_{c,R}/e_{u,R} \\ e_{c,B}/e_{u,B} \\ e_{c,G}/e_{u,G} \end{bmatrix}. \tag{3}$$

So, to solve the illuminant estimation problem, additional assumptions have to be made about either the properties of the scene or the properties of the illumination. These assumptions can vary a lot, resulting in a wide variety of color constancy methods. Usually, these assumptions are based on the reflectance properties of the whole scene; however, this allows such methods to predict only one illumination in the scene, which in turn leads to improper correction of some areas in multi-illuminant scenes (see Fig. 1). Section 2 gives a more detailed overview of these assumptions. We propose a novel method that can learn these assumptions about the reflectance properties of the scene, and thus is able to extract the color of the illumination. To encode information about the reflectance properties and illumination, our method takes the raw image as input X and produces two separate outputs C and I. The first output is the canonically

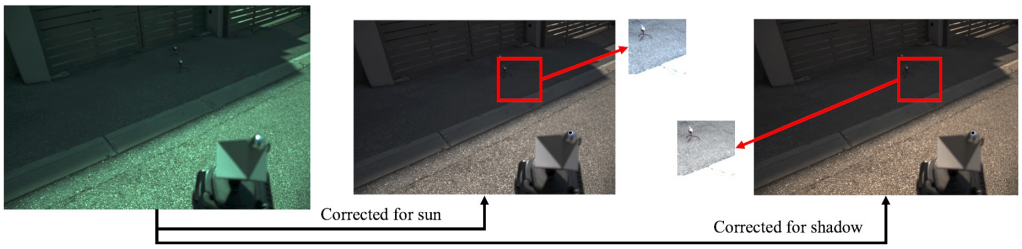


Fig. 1. Two corrections by different illuminants present in the raw image obtained from a Nikon D7000 DSLR camera. The middle image is corrected for sunlight, which results in the shaded regions having an unnatural blue hue. The right image is corrected for skylight, which gives the sunlit region an orange cast. Values for the primary and secondary illuminants were obtained from the gray sides of the SpyderCube calibration object.

illuminated image C and the second is the per-pixel estimation of the illumination I. The final output of the network R is the reconstructed input image X, which is obtained by multiplying the canonical image with the illumination estimation [inverse of the von Kries correction; see Eq. (2)]. The use of the canonical image helps our model learn the normal appearance of scenes under canonical illumination, and allows for easier convergence and improved estimation and segmentation results. Also, since our estimation is not constrained in any way, it allows our model to estimate illumination in multi-illuminant scenes with an arbitrary number of illuminants. To test the performance of our model in a multi-illuminant scenario, we used real-world images with two illuminants and artificially generated images with three illuminants. However, this type of per-pixel estimation can still produce local artifacts and noticeable errors in estimation. For this reason, we propose a simple clustering strategy that can convert the per-pixel estimation into a segmentation mask, where each region of the image is illuminated with only one illuminant. This in turn allows us to convert any multi-illuminant image, which would require multi-illuminant methods to solve, into a series of single illuminant estimation problems, which is beneficial as single illuminant estimation methods usually preform much better than their multi-illuminant counterparts.

In the next section, we will describe other color constancy methods and different types of autoencoder networks. Then, in Section 3, we describe our model and the loss function used for training, and in Section 4, we present the results of our method on real-world and generated multi-illuminant images, for both illuminant estimation and segmentation. Finally, in Section 5, we provide a conclusion.

2. RELATED WORK

A. Color Constancy

Color constancy in humans and our perception of illumination is a phenomenon that has been studied for almost 200 years, starting with [3–7], among others. More recently, with the rapid development of digital cameras, computational color constancy has become an important area of research, as it is implemented in all modern cameras. Furthermore, as shown by Eq. (1), color constancy with illumination estimation is an ill-posed problem. This means that additional assumptions have to be made about the properties of the scene and the illumination. These assumptions can vary greatly, which gives rise to many types of color constancy methods. Roughly speaking, there are two main categories of color constancy methods [8]: statistics based and learning based. The most common example of statistics-based methods is the gray world method [9], where it is assumed that the natural reflectance of a scene is gray, and any deviation from that has to be caused by illumination. Gray world can be considered as a subset of a larger framework called the gray edge framework [10], where the assumption about gray reflectivity is imposed on the nth derivation of the image.

On the other hand, learning-based methods, as their name suggests, use machine learning or deep learning techniques to learn the properties of both illuminations and scenes. Some such machine learning methods include illuminant gamut mapping models [11], higher-level statistics of the surfaces in the images [12], typical machine learning methods [13–15], and generative and Bayesian methods [16–20], among others. One notable multi-illuminant approach utilized conditional random fields to predict more than one illuminant per image [21], while [22] proposed a method based on the top-down and bottom-up mechanisms of the human visual system. With the development of deep learning methods, computational color constancy methods also started to employ those techniques. Some of the earliest deep learning models included simpler convolutional neural networks (CNNs), such as [23–25]. Other methods such as [26,27] utilize only chroma histograms to predict illumination. More recently, many network architectures have been utilized for color constancy, such as generative adversarial networks [28] or a combination of illumination estimation and classification of the type of illuminant [29,30]. There have also been specialized networks, such as the one proposed in [31], where the final layer uses an attention mechanism to detect regions suitable for illuminant estimation. The method most similar to ours is the one proposed in [32], where an autoencoder network ([33], Ch. 14) is used to learn illuminant estimation. However, unlike our approach, this method was limited to single illuminant scenes, as it was able to predict only one illuminant. Another method similar to ours, in terms of illuminant segmentation, was proposed in [34], where a vision transformer architecture segments images based on illumination, and detects errors in estimation.

One of the main problems with modern multi-illuminant color constancy methods is the lack of a large multi-illuminant dataset. Methods proposed in [21,35] both included small multi-illuminant datasets. However, both contained less than 100 images, which is considerably too small for training larger deep learning models. Another small dataset was proposed in [36], but these images were not publicly available. To combat

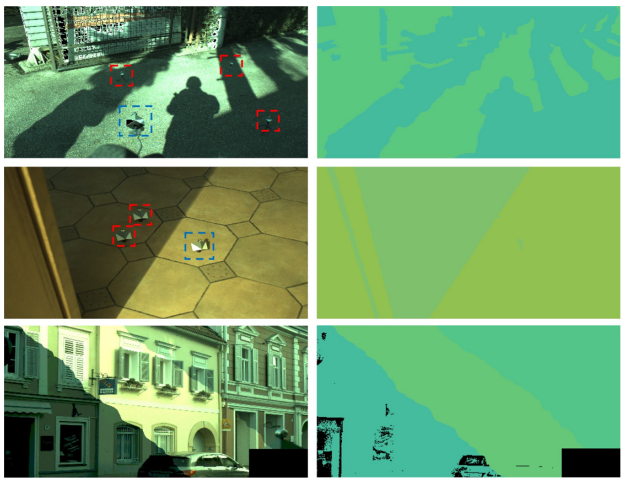


Fig. 2. Examples of images used for training. The first and second rows show an outdoor and indoor scene, with four and three SpyderCube calibration objects, respectively. Multiple cubes marked by red squares are placed in the region illuminated by the ambient illuminant, which can vary throughout the scene more than direct illumination (e.g., sunlight or one light bulb). We used only images where the difference in the ground truth between the measured ambient illumination was less than 1° to ensure that the manual annotation of the regions shown in the second column is accurate. Our annotation procedure is similar to that described in [35]. The third row shows an example of an artificially generated image and its ground truth illumination.

this, most methods used artificially relighted images. However, it is questionable how methods trained and tested on artificial images perform in real-world environments. In this work, we trained and tested our proposed model on many real-world outdoor and indoor images with two illuminants that we collected and annotated (which we will make available online), and on artificial images with three illuminants. We also tested the performance of our model trained on artificial images on real-world images, to assess how representative our artificial images were of real-world scenes. It was possible to test our method that was trained on images with three illuminants on images with two illuminants, because our model is not dependent on the number of illuminants (this is also true of the convolutional model (BCNN) presented in [24], which we use for comparison). Figure 2 shows different types of real-world and generated images that were used for training our model, as well as the manually annotated and generated ground truth, respectively. Figure 2 also shows that the artificial images were generated so that each illumination is present in larger areas of the image, to closer match the distribution found in real images. We decided on this approach because our experiments showed that it improved the generalization properties of models trained on such images. Additionally, a similar approach was also used in [24], which we use for comparison with our model.

3. AUTOENCODER MODEL

In this work, we propose a novel training method for learning the assumptions required for multi-illuminant scenes. The network is trained similarly to a classical autoencoder, where the goal is to reconstruct the original image. However, unlike traditional autoencoders, where the architecture is composed of

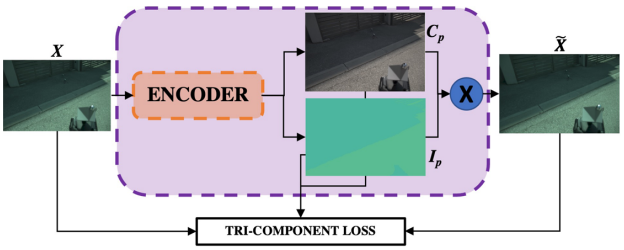


Fig. 3. Schema of the autoencoder training strategy. The output of the encoder is the canonically illuminated image C_p and the per-pixel estimation I_p, whose multiplication creates $\tilde{X}$, the reconstructed input image X. The loss is calculated using all three outputs as described in Eq. (5).

an encoder whose goal is to create an embedding z and a decoder that uses z to reconstruct the input, our approach can be used with any network architecture that produces an output of the same spatial dimensions as the input. The procedure can be described as follows (for the schema, see Fig. 3).

1. The final layer of the encoder network p, whose spatial dimensions (h, w) match those of the input X, is composed of six channels. The first three channels of p are the canonically illuminated scene C_p, while the other three channels represent the normalized per-pixel estimation of the illumination I_p. We emphasize that this estimation is independent of the number of illuminants in the scene (i.e., the model does not know anything about the number of illuminants in the scene).
2. The input image X (which is a colored image with three channels) is then reconstructed at the output by simply multiplying C_p and I_p, following the von Kries [3] approximation.
3. For training, a loss function can be computed for all three outputs of the network, since the ground truth information is available in our dataset for the intermediary outputs C_p and I_p, and any reconstruction loss usually used for training autoencoders can be used for the reconstructed input [Eq. (2)].

This type of supervised training of the model for color constancy improves the performance by encouraging it to create a valid canonical image and thus learn to separate the color of the illumination from the colors of objects in the scene. We apply this training strategy to a custom fully convolutional deep neural network with 4 million parameters, based on a feature pyramid network (FPN) [37] architecture. All three outputs of the network, along with the ground truth information from the dataset, can be used to compute the composite loss L used in training. For the illumination estimation loss, we propose a new regularized loss function designed to preserve the gradual transition between illuminations in real-world scenes. This loss is composed of the mean squared error (MSE) term between the predicted and actual illumination for each pixel, as well as a regularization term that ensures smoothness, by penalizing fast changes in the color of illumination. We use this regularization to reduce the number of artifacts in the estimation that can arise from large colored objects present in the scene (see Fig. 5). The regularization term is computed by convolution of the image with a "uniformity" filter of size $n \times n$:

$$f_n = \begin{bmatrix} -1 & -1 & \dots & -1 & -1 \\ \vdots & \ddots & (n^2-1) & \ddots & \vdots \\ -1 & -1 & \dots & -1 & -1 \end{bmatrix}. \quad (4)$$

As Eq. (4) shows, the "uniformity" filter is an approximation of the Laplacian filter of size $n \times n$, which serves as a discrete filter to find the second derivative of the image and highlights areas of rapid change of values ([38], pp. 98–99). Since we want our estimation to have smooth transitions, we wish to keep the Laplacian of the image as close to zero as possible. Another useful property of the Laplacian is that it is very sensitive to noise, and thus its minimization will also tend to minimize the noise in the image. We apply regularization to the per-pixel illumination estimation only. For both the reconstruction and the extracted canonical image, MSE is used without any regularization, because, unlike in the case of illumination, we want to preserve details about the edges and sharp transitions in color. This encourages the model to learn that in real-world scenes, fast and large changes in the color of the scene are usually the result of a change in surface reflectively and not in illumination. The final loss function is computed as

$$L(I_p, C_p, R, I_r, C_r, X) = \alpha \mathrm{MSE}(C_p, C_r) + \beta \mathrm{MSE}(\tilde{X}, X) + \gamma \mathrm{MSE}(I_p, I_r) + \delta \frac{1}{M} \sum_{i,j} (f_n * I_p)^2_{i,j}, \quad (5)$$

where I is the per pixel illumination map, C is the canonical image, $\tilde{X}$ is the reconstruction, X is the original image, and α, β, γ, δ are coefficients used to tune the importance of each component to the overall loss.

Another use of our model is image segmentation based on the dominant illuminant in regions of the scene. We propose this use of our model because it is possible for the model to inaccurately predict the color of the illuminant in some more difficult scenes; the regions where that illuminant was detected were consistent with one of the illuminants in the scene. So, in those scenarios, we can use the segmentation mask to separate the scene into regions with only one illumination, and use some more complex and accurate single illuminant estimation methods to produce a new estimation map for the scene. To achieve that, we use the per-pixel estimation output of our model and apply k-means clustering to it. The number of clusters was set manually to either two or three for real-world and artificial images, respectively. The output per-pixel estimation was first down-sampled eight times to reduce the number of points given to the clustering algorithm. The clustering was done in RGB three-dimensional space using the Euclidean distance between points. The final segmentation mask was produced by finding the nearest centroid produced by the clustering algorithm for each pixel in the per-pixel estimation.

A. Experiments and Training

We implemented our model in TensorFlow 2.4 [39], and it was trained on a computer with a Ryzen 3700x CPU and Nvidia RTX 2080Ti GPU. Training was done over 5000 epochs, and each epoch consisted of passing through 1/50th of our dataset. Learning rate was set using a cosine annealing warm restart [40] schedule, with range from $1 * 10^{-4}$ to $1 * 10^{-6}$, and the weights were optimized by the Adam optimizer [41] with weight decay of $1 * 10^{-7}$. For images with two illuminations, dataset contained 600 images of outdoor scenes taken by two digital single-lens reflex (DSLR) cameras (Canon EOS 550D and Nikon D7000), for each of which ground truth illumination was extracted using SpyderCube calibration objects, and the regions where the illuminants were present were annotated manually. Figure 4 (first row and first column of second row) shows an example of the image from the dataset and the corresponding map of per-pixel illuminations. The training was done using the loss function described in Eq. (5) with the parameters set to $\alpha = 1$, $\beta = 1$, $\gamma = 1.4$, $\delta = 1.4$, and the filter size was set to $n = 5$. These hyperparameters were tuned on outdoor images and were kept the same for all experiments. We also applied contrast normalization by stretching the image histogram such that the pixel values for each image were in the range [0, 1]. Augmentation that consisted of random rotation from $-15°$ to $15°$, random center cropping, and random horizontal and vertical flipping was applied during training to prevent overfitting, since the number of outdoor images available is not large.

Images with three illuminants were artificially created since, to our knowledge, there does not exist any dataset with such images and per-pixel illumination ground truth. These images were created by taking canonically illuminated images from Cube+ [42], which were then illuminated with new illuminants using the von Kries model. The spatial distribution of illuminants was generated by combining random segments of linear functions to create a complex enough mask. We experimented with using simple masks where the regions were separated by straight lines, or by coloring random patches, but those proved to be less representative of real-world scenes, which resulted in much lower performance when the models were tested on real-world scenes. We trained a subset of models on only artificial images with three illuminants, and tested them on real-world images with two illuminants, to allow us to better test the generalization properties of the models on different types of images from those used for training. We also tested the best performing model on images from the Color Checker [16] and Cube+ [42] datasets, and we present our findings on those images in Section 4.

To test the improvements in performance that can be obtained using our training method, we performed an ablation study by removing the canonical image and the reconstruction, and the model was trained to simply minimize the MSE between the ground truth illumination map and the estimation. We also experimented with removing the smoothness regularization from our loss function, which also led to decreases in performance in some areas. We also compared our model to illumination segmentation models proposed in [34,36], a U-Net model with a Visual Geometry Group network (VGG16) [43] as the encoder (implemented such that one illuminant was known, the same as the vision transformer in [34]) and to a baseline Otsu threshold applied to the brightness histogram of the image. The segmentation results are shown in Tables 3 and 4 for outdoor and artificial images, respectively. For illuminant estimation, we implemented some classical single illuminant estimation models, as well as the multi-illuminant CNN model described in

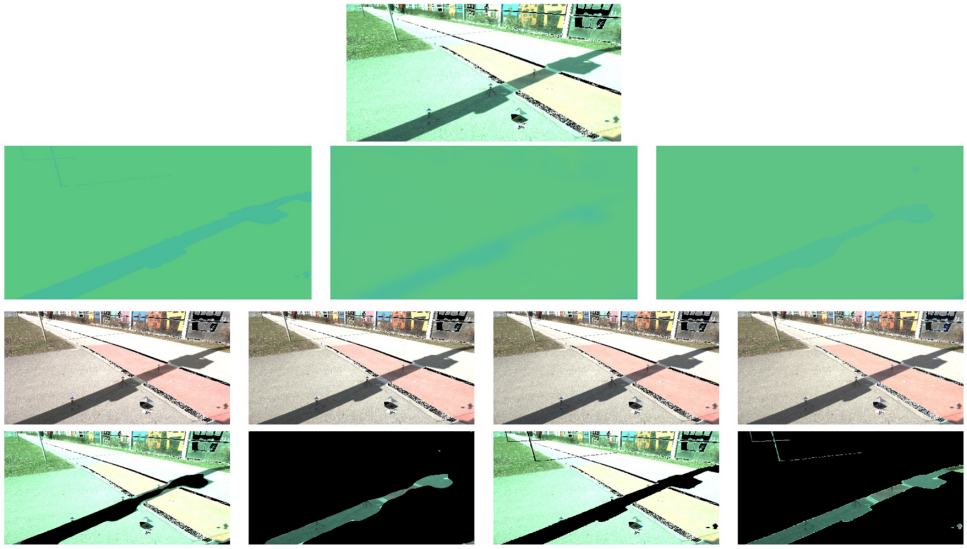


Fig. 4. Example of all outputs of the model and the comparison to the ground truth. The first row shows the input image, the second shows, in order, ground truth illumination, estimated per-pixel map, and clustered estimation with only two illuminants. In the third row, the illuminant corrected images are shown, with the first column showing the ground truth correction, the second and third showing corrections by the second and third illumination maps from the previous row, respectively, and the fourth column shows the canonical output when using the proposed training strategy. The bottom row shows the masked output of the segmentation task, where the first two columns are the segmentation produced by clustering the illuminant map, while the second two are the ground truth segmentation. The model was given linear raw images, but gamma correction was applied to these images for easier visualization.

[24] and the Multi-Illuminant Random Field (MIRF) method described in [21].

4. RESULTS

In this section, we present the results of our autoencoder training method for both illuminant estimation and segmentation tasks. We tested our method for illumination segmentation on real-world and artificial images by applying k-means clustering to the output illuminant estimation, and compared our model with two baselines and four other deep learning models.

For illuminant estimation, we experimented with three different ways of acquiring the final per-pixel estimation of the illuminants. The first approach simply used the estimation output from our network, which is independent of the number of illuminants in the scene. This is also true for the BCNN [24]. For the second approach, cluster centers from the k-means clustering used for segmentation were applied as the global estimation for the region where they were the centroid. For the third approach, a Fully Convolutional Color Constancy with Confidence-weighted Pooling network (FC4) [31], trained on Cube+ [42], was used to produce a global estimate for each segmented region, which was then used instead of cluster centers for the final estimation map. We purposefully trained the FC4 model on the Cube+ dataset to show that our model can be used in combination with different single illuminant estimation methods, while still obtaining good results. To make the comparisons fair, we applied the estimated values from the second and third approaches to the segmentation maps and thus once again obtained per-pixel estimates, then calculated the mean angular error between each pixel in the per-pixel estimate and the ground truth. The results on outdoor and artificial images are reported in Tables 1 and 2, respectively. We can see that the performance for illuminant estimation is highest when direct output is used, and is again achieved by our model. However, the proposed segmentation and estimation approach allows us greater flexibility, as different estimation methods can be used. So, as the more researched single illuminant methods continue

Table 1. Comparison of Angular Errors on Real-World Images[a]

	Real-World Estimation				
Model	**Mean**	**Median**	**Trimean**	**Best 25%**	**Worst 25%**
GW [9]	5.55	5.37	5.45	2.54	8.84
GE1 [10]	5.44	5.63	5.51	1.98	8.93
GE2 [10]	5.77	5.71	5.75	2.22	9.45
FC4 [31]	4.09	3.85	3.91	1.78	6.76
MIRF [21]	7.08	6.20	6.40	2.70	12.67
BCNN [24]	3.11	2.77	2.82	1.75	5.14
AE-EST	3.17	2.75	2.88	1.79	5.18
AE-EST (pt)	3.13	2.74	2.80	1.80	5.11
AE-NOSMO	2.96	**2.66**	2.67	1.66	4.88
AE-FULL-GW	4.68	4.14	4.31	2.17	7.75
AE-FULL-MIRF	6.04	5.29	5.51	2.30	10.76
AE-FULL-FC4	3.01	2.74	2.71	1.73	4.90
AE-FULL	**2.96**	2.71	**2.67**	1.72	**4.81**

[a]The suffixes (GW, MIRF, and FC4) added to the AE-FULL model indicate that the segmentation mask was used to determine the single illuminant regions and the method in the suffix was used to estimate the illumination for each region. We can see that when those methods were combined with the segmentation mask, they produced better results than when used only to produce the single illuminant estimation. However, it is important to note that we need to know the number of illuminants when employing this segmentation strategy. Methods without those suffixes do not need to know the number of illuminants in advance.

Table 2. Comparison of Angular Errors on Artificial Images with Three Illuminants

	Artificial Estimation				
Model	**Mean**	**Median**	**Trimean**	**Best 25%**	**Worst 25%**
GW [9]	6.29	5.49	5.69	3.21	10.82
GE1 [10]	5.92	5.25	5.43	3.25	9.69
GE2 [10]	7.98	7.15	7.35	4.32	13.04
FC4 [31]	3.51	3.24	3.25	2.46	5.12
AE-EST	4.04	3.77	3.70	1.68	7.08
BCNN [24]	3.58	3.50	3.52	1.80	5.52
AE-NOSMO	3.44	**3.33**	**3.39**	1.53	5.41
AE-FULL	**3.41**	3.35	3.40	**1.46**	**5.35**

to improve, the current FC4 model can easily be replaced with a more accurate model. We see the same behavior on artificial images with three illuminants. The visual comparison of the estimation results among methods can be seen in Figs. 5 and 6. In Fig. 6, we can see that our model was able to produce accurate corrections, and even managed to learn the mixing properties of illumination, as can be seen in the indoor scenes

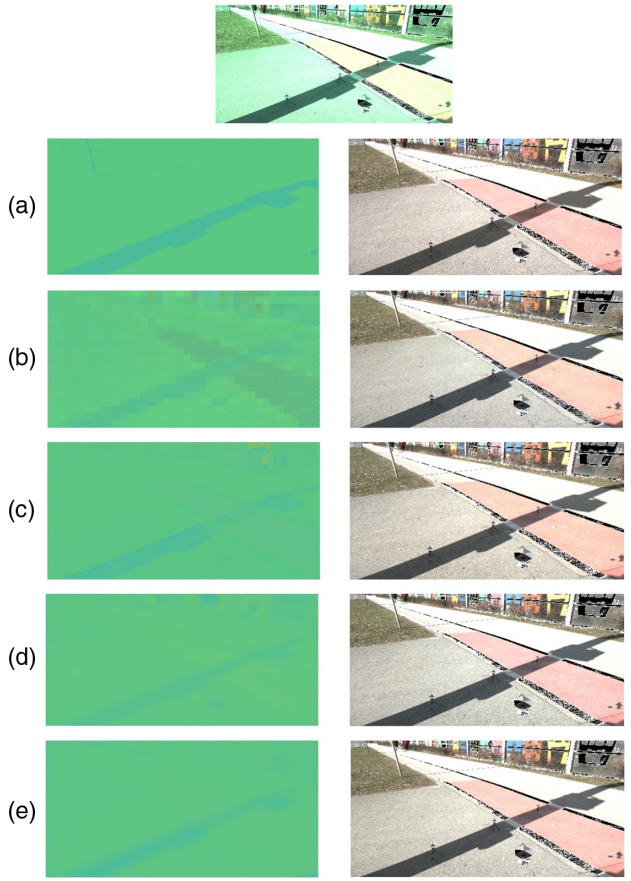


Fig. 5. Comparison of per pixel estimation results and corrections for different methods: (a) ground truth, (b) patch-based BCNN [24], (c) per pixel estimation using FPN network, (d) FPN autoencoder without smoothing, and (e) FPN autoencoder with smoothing and tri-component loss. Our models in the bottom row produced the best estimation and the correction with the fewest artifacts. We can also notice that methods (b) and (c) were not able to fully discriminate between the red paint on the ground and the color of the illumination, resulting in wrong corrections on those parts of the image, while our proposed method did not have that problem.

in the third, second to last, and last columns. We can also see that the model struggled with the scene in the second to last column, where the illumination was produced by a fluorescent bulb, which produces strongly colored illumination, making it more difficult to accurately predict. Another probable reason for the poorer performance on those images is the fact that our dataset contained the fewest scenes under such illumination. In all of our tables, we use different abbreviations to describe different models and their variations. AE-FULL represents the autoencoder model trained with both smoothing and all three outputs, AE-NOSMO is the autoencoder model without the smoothing constraint, and AE-EST is the model trained using only the loss between the estimated illumination and the ground truth, without the reconstruction and canonical outputs. Furthermore, "gen" stands for generalization, and indicates that the model was trained on the artificial images and tested as is (without any fine-tuning) on real-world images, while "pt" stands for pretraining, and means that the model was first trained on the artificial images and then fine-tuned on real-world images before testing.

We present the results of the segmentation based on illumination for real-world images in Table 3. As we can see, our method outperforms all other methods on the illuminant segmentation task. The first baseline method uses a brightness threshold as described in [36], however with the threshold value set to $c_t = 0.1$ instead of $c_t = 0.08$ as described in the original paper, as it achieves better performance on our dataset. Our model, trained using our tri-component loss function, outperformed all other models as well as the same model when it was trained using only simple MSE loss (visual comparison of results can be seen in Fig. 7). Also, when we compare our model to the vision transformer model from [34] or VGG U-Net model, our model does not rely on the fact that one illuminant will be known. However, it is important to note that model is specialized for illuminant segmentation, is more complex (19 million parameters compared to 4 million parameters of our model), relies on knowledge about or estimation of one illuminant *a priori*, and is also limited to segmenting only two illuminants in a single pass. On the other hand, when we take a look at the performance of our model on the harder problem of artificial images with three illuminants (see Table 4), we can see that our model, when trained using the proposed training strategy, outperforms all other models.

The last set of experiments that we conducted was to test the generalization of models trained on artificial images with three illuminants by evaluating them on real-world images with two illuminants. The test set of real-world images was selected to match the test set used in previous experiments to allow for fair comparison of the results. Additionally, in Tables 5 and 6 results with the "gen" suffix show the results of models trained on artificial images and then tested on real-world images. The much smaller drop in performance of models trained using the proposed training strategy shows that they generalized much better than models trained using other methods. This indicates that our training strategy allows the model to simultaneously learn the correct distribution of both illuminants and reflectance properties of real-world scenes, which is beneficial because of the lack of larger multi-illuminant datasets that could be used for training.

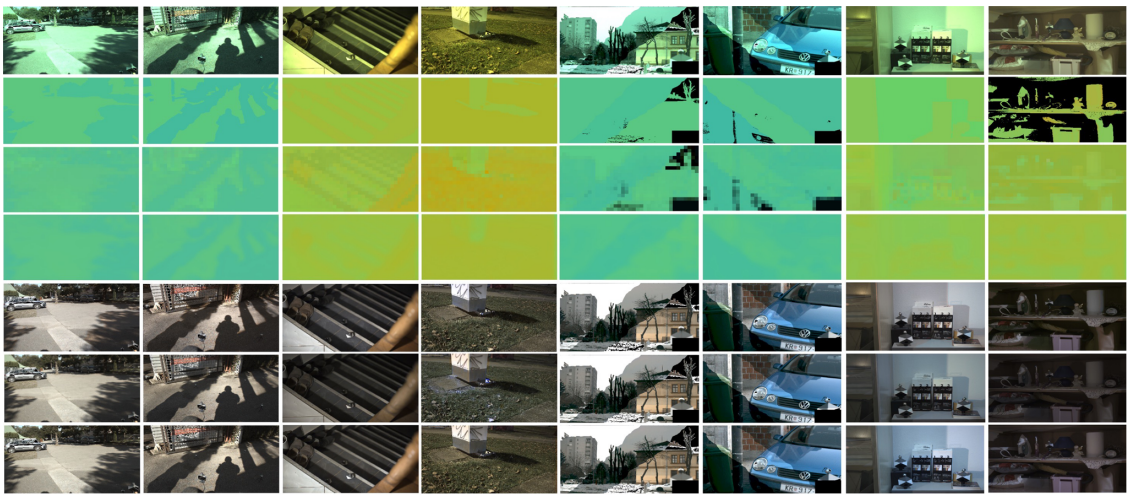


Fig. 6. Comparison of per pixel estimation results and corrections between the ground truth (second and fifth rows), patch-based BCNN [24] (third and sixth rows), and FPN autoencoder with smoothing and tri-component loss (AE-FULL, fourth and seventh rows). Images in the indoor scenes in the third, second to last, and last columns show the more difficult indoor examples, where the scenes contain mixing of very different illuminants, e.g., sunlight and light bulbs.

Table 3. Dice Coefficient Results of Segmentation on Outdoor Images

	Real-World Segmentation				
Model	**Mean**	**Median**	**Trimean**	**Best 25%**	**Worst 25%**
Otsu [44]	0.82	0.85	0.84	0.96	0.65
TRESH [36]	0.79	0.83	0.82	0.93	0.57
VGG16 (U-Net)	0.87	0.83	0.82	0.93	0.57
T-LARGE [34]	0.88	0.90	0.89	0.96	0.76
BCNN [24]	0.88	0.90	0.90	0.96	0.77
BCNN (gen)	0.59	0.59	0.59	0.71	0.49
AE-EST	0.88	0.89	0.89	0.96	0.76
AE-EST (gen)	0.87	0.88	0.88	0.95	0.75
AE-EST (pt)	0.89	0.91	0.91	0.97	0.76
AE-NOSMO	0.88	0.91	0.90	0.97	0.75
AE-NOSMO (gen)	0.88	0.90	0.89	0.96	0.75
AE-FULL (gen)	0.87	0.89	0.89	0.96	0.76
AE-FULL	**0.90**	**0.92**	**0.91**	**0.97**	**0.78**

Finally, we tested our best performing model trained on real-world images with two illuminants (AE-FULL) on single illuminant images from the Color Checker [16] (we used the latest version of ground truths [45]) and Cube+ [42] datasets. We present a visual comparison of those results in Fig. 8. As we can see in those example scenes, some images in those datasets actually contain more than one illumination, which our model was able to predict and produced a subjectively more pleasing correction than the single illuminant ground truth. For example, some outdoor images contain more than one illuminant (i.e., sun and shadow), such as the example in the first column, where the white part of the Japanese flag looks orange when corrected using the provided ground truth, and white when corrected using our method. We do not provide any numerical comparison of the results, as our model produced a per-pixel illumination estimation and the ground truth contained only a single illumination value, even though, as we show, some scenes contain more than one illumination, which would make the comparison to other methods unfair.

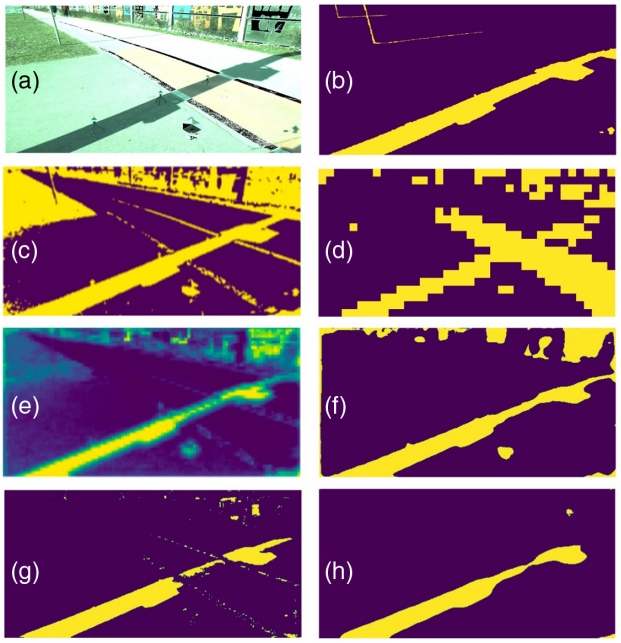


Fig. 7. Comparison of illumination segmentation for different methods: (a) input image, (b) ground truth mask, (c) Otsu segmentation, (d) patch-based BCNN [24], (e) transformer network [34], (f) FPN network without the autoencoder tri-component loss training, (g) FPN autoencoder without smoothing, and (h) FPN autoencoder with smoothing and tri-component loss. The methods in the bottom row, which were both trained using the autoencoder procedure, produced the most accurate segmentation masks. Furthermore, methods (c), (d), and (f) were not able to fully discriminate among the red paint, green grass, and colored objects in the background, which resulted in inaccurate segmentation in those regions.

Table 4. Dice Coefficient Results of Three Illuminant Artificial Images

Artificial Segmentation					
Model	**Mean**	**Median**	**Trimean**	**Best 25%**	**Worst 25%**
LUM-TRESH	0.57	0.55	0.56	0.71	0.44
BCNN [24]	0.74	0.75	0.74	0.88	0.58
AE-EST	0.79	0.82	0.81	0.93	0.61
AE-NOSMO	0.81	**0.84**	**0.83**	**0.95**	0.62
AE-FULL	**0.81**	0.83	0.82	0.94	**0.64**

Table 5. Comparison of Angular Errors on Real-World Images, between Models Trained on Artificial Images and Tested on Real-World Images

Real-World Estimation Generalization					
Model	**Mean**	**Median**	**Trimean**	**Best 25 %**	**Worst 25%**
BCNN (gen) [24]	3.47	3.08	3.19	1.97	5.57
AE-EST (gen)	4.22	4.07	4.02	**1.43**	7.35
AE-NOSMO (gen)	**3.32**	**3.17**	**3.09**	1.54	**5.48**
AE-FULL (gen)	3.41	3.18	3.16	1.73	5.61

Table 6. Dice Coefficient Results of Generalization Segmentation Test on Real-World Images

Real-World Segmentation Generalization					
Model	**Mean**	**Median**	**Trimean**	**Best 25%**	**Worst 25%**
BCNN (gen)	0.59	0.59	0.59	0.71	0.49
AE-EST (gen)	0.87	0.88	0.88	0.95	0.75
AE-NOSMO (gen)	**0.88**	**0.90**	**0.89**	**0.96**	0.75
AE-FULL (gen)	0.87	0.89	**0.89**	**0.96**	**0.76**

5. CONCLUSION

In this work, we presented an efficient method for training deep neural networks for both illuminant estimation and illuminant segmentation. The models were trained to simultaneously predict the illumination and the reflectance properties of objects by creating a per-pixel estimation of illumination and a canonically illuminated image. These images are then combined using the von Kries model to recreate the input image, with all three images being used by our tri-component loss function. For evaluation and usage, the information about the recreated image and the canonically illuminated image can be discarded, and only per-pixel estimation is used. Additionally, we show that clustering these per-pixel illuminant estimations can be used to create a segmentation of illuminants in the scene. The results of our experiments show that our method benefits from our approach on both real-world and artificial multi-illuminant images, for both estimation and segmentation tasks, where it performs the best for illuminant estimation and is on par with more specialized illuminant segmentation methods. We also show that, when using our training method, our models show greater generalization, as they outperform all other models when trained on purely artificial images and then evaluated on real-world images. Another benefit of our approach is that it is invariant to the number of illuminants, and can even be used on images from single illuminant datasets, with the possibility to detect images containing more than one illuminant, e.g., in Color Checker [16] or Cube+ [42] datasets. Furthermore, the segmentation mask produced by our method can be combined with different single illuminant estimation methods, which results in the improvement of their performance on multi-illuminant scenes and allows for greater flexibility, as the segmentation and estimation models can be changed depending on the scenarios in which they are used. In the future, another avenue of research in the area of image segmentation end estimation is the use of unannotated images, which are easy to acquire but difficult to annotate. This could be achieved by adapting our tri-component loss function to allow for different combinations of annotated and unannotated images, but more research is required to find the best performing function.

Disclosures. The authors declare no conflicts of interest.

Data availability. Data underlying the results presented in this paper are not publicly available at this time but may be obtained from the authors upon reasonable request.

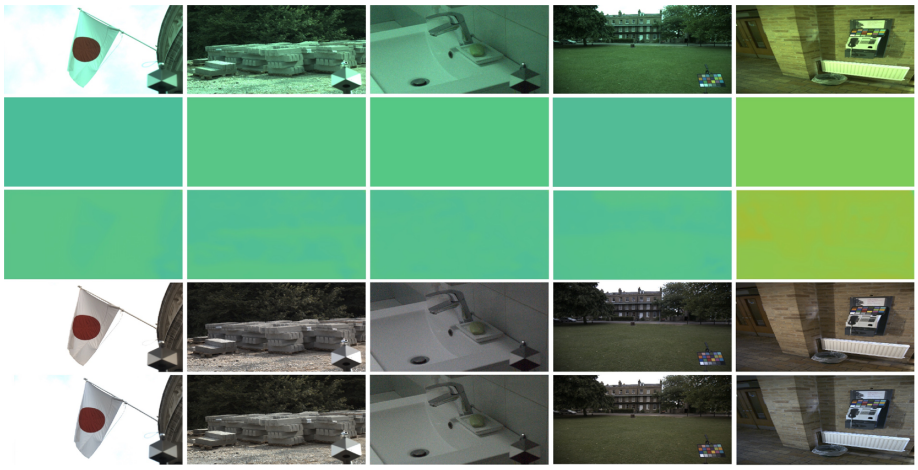


Fig. 8. Comparison of per pixel estimation results and corrections between the ground truth (second and fourth rows) and FPN autoencoder with smoothing and tri-component loss (AE-FULL, third and fifth rows) on the images from the Cube+ [42] (first three columns) and Color Checker [16] datasets.

REFERENCES AND NOTES

1. G. Klinker, S. Shafer, and T. Kanade, "A physical approach to color image understanding," Int. J. Comput. Vision **4**, 7–38 (2004).
2. S. A. Shafer, "Using color to separate reflection components," Color Res. Appl. **10**, 210–218 (1985).
3. J. Von Kries, "Influence of adaptation on the effects produced by luminous stimuli," Handbuch Physiol. Menschen **3**, 109–282 (1905).
4. G. Monge, "Mémoire sur quelques phénomènes de la vision," in *Annales de Chimie* (1789), Vol. **3**, pp. 131–147.
5. T. Young, *A Course of Lectures on Natural Philosophy and the Mechanical Arts* (Taylor and Walton, 1845), Vol. **1**.
6. D. B. Judd, "Hue saturation and lightness of surface colors with chromatic illumination," J. Opt. Soc. Am. **30**, 2–32 (1940).
7. E. H. Land and J. J. McCann, "Lightness and retinex theory," J. Opt. Soc. Am. **61**, 1–11 (1971).
8. A. Gijsenij, T. Gevers, and J. van de Weijer, "Computational color constancy: survey and experiments," IEEE Trans. Image Process. **20**, 2475–2489 (2011).
9. G. Buchsbaum, "A spatial processor model for object colour perception," J. Franklin Inst. **310**, 1–26 (1980).
10. J. Van De Weijer, T. Gevers, and A. Gijsenij, "Edge-based color constancy," IEEE Trans. Image Process. **16**, 2207–2214 (2007).
11. A. Gijsenij, T. Gevers, and J. Van De Weijer, "Generalized gamut mapping using image derivative structures for color constancy," Int. J. Comput. Vis. **86**, 127–139 (2010).
12. H. R. V. Joze and M. S. Drew, "Exemplar-based color constancy and multiple illumination," IEEE Trans. Pattern Anal. Mach. Intell. **36**, 860–873 (2014).
13. B. Funt and W. Xiong, "Estimating illumination chromaticity via support vector regression," in *Color Imaging Conference* (2004), Vol. **50**, pp. 47–52.
14. V. Agarwal, A. V. Gribok, and M. A. Abidi, "Machine learning approach to color constancy," Neural Netw. **20**, 559–563 (2007).
15. V. Agarwal, A. V. Gribok, A. Koschan, and M. A. Abidi, "Estimating illumination chromaticity via kernel regression," in *International Conference on Image Processing* (2006), pp. 981–984.
16. P. V. Gehler, C. Rother, A. Blake, T. Minka, and T. Sharp, "Bayesian color constancy revisited," in *IEEE Conference on Computer Vision and Pattern Recognition* (IEEE, 2008), pp. 1–8.
17. D. H. Brainard and W. T. Freeman, "Bayesian color constancy," J. Opt. Soc. Am. A **14**, 1393–1411 (1997).
18. C. Rosenberg, A. Ladsariya, and T. Minka, "Bayesian color constancy with non-Gaussian models," in *Advances in Neural Information Processing Systems* (2003), Vol. **16**, pp. 1595–1602.
19. F. Laakom, J. Raitoharju, A. Iosifidis, U. Tuna, J. Nikkanen, and M. Gabbouj, "Probabilistic color constancy," arXiv:2005.02730 (2020).
20. G. Sapiro, "Color and illuminant voting," IEEE Trans. Pattern Anal. Mach. Intell. **21**, 1210–1215 (1999).
21. S. Beigpour, C. Riess, J. Weijer, and E. Angelopoulou, "Multi-illuminant estimation with conditional random fields," IEEE Trans. Image Process. **23**, 83–96 (2014).
22. S.-B. Gao, Y.-Z. Ren, M. Zhang, and Y.-J. Li, "Combining bottom-up and top-down visual mechanisms for color constancy under varying illumination," IEEE Trans. Image Process. **28**, 4387–4400 (2019).
23. S. Bianco, C. Cusano, and R. Schettini, "Single and multiple illuminant estimation using convolutional neural networks," IEEE Trans. Image Process. **26**, 4347–4362 (2017).
24. S. Bianco, C. Cusano, and R. Schettini, "Color constancy using CNNs," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops* (2015), pp. 81–89.
25. W. Shi, C. C. Loy, and X. Tang, "Deep specialized network for illuminant estimation," in *European Conference on Computer Vision* (Springer, 2016), pp. 371–387.
26. J. T. Barron, "Convolutional color constancy," in *Proceedings of the IEEE International Conference on Computer Vision* (2015), pp. 379–387.
27. J. T. Barron and Y.-T. Tsai, "Fast Fourier color constancy," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2017), pp. 886–894.
28. P. Das, A. S. Baslamisli, Y. Liu, S. Karaoglu, and T. Gevers, "Color constancy by GANs: an experimental survey," arXiv:1812.03085 (2018).
29. K. Koščević, M. Subašić, and S. Lončarić, "Deep learning-based illumination estimation using light source classification," IEEE Access **8**, 84239–84247 (2020).
30. S. Bianco, G. Ciocca, C. Cusano, and R. Schettini, "Improving color constancy using indoor–outdoor image classification," IEEE Trans. Image Process. **17**, 2381–2392 (2008).
31. Y. Hu, B. Wang, and S. Lin, "FC4: Fully convolutional color constancy with confidence-weighted pooling," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2017), pp. 4085–4094.
32. F. Laakom, J. Raitoharju, A. Iosifidis, J. Nikkanen, and M. Gabbouj, "Color constancy convolutional autoencoder," in *IEEE Symposium Series on Computational Intelligence (SSCI)* (IEEE, 2019), pp. 1085–1090.
33. I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning* (MIT, 2016).
34. D. Vršnak, I. Domislović, M. Subašić, and S. Lončarić, "Illuminant estimation error detection for outdoor scenes using transformers," in *12th International Symposium on Image and Signal Processing and Analysis (ISPA)* (IEEE, 2021), pp. 276–281.
35. A. Gijsenij, R. Lu, and T. Gevers, "Color constancy for multiple light sources," IEEE Trans. Image Process. **21**, 697–707 (2012).
36. S.-H. Lee, S.-M. Woo, J.-H. Choi, and J.-O. Kim, "Two-step multi-illuminant color constancy for outdoor scenes," in *IEEE International Conference on Image Processing (ICIP)* (2017), pp. 710–714.
37. T. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2017), pp. 936–944.
38. D. Vernon, *Machine Vision: Automated Visual Inspection and Robot Vision* (Prentice-Hall, 1991).
39. M. Abadi, A. Agarwal, P. Barham, *et al.*, "TensorFlow: large-scale machine learning on heterogeneous systems," arXiv:1603.04467 (2015). Software available from https://tensorflow.org.
40. I. Loshchilov and F. Hutter, "SGDR: stochastic gradient descent with warm restarts," arXiv:1608.03983 (2017).
41. D. P. Kingma and J. Ba, "Adam: a method for stochastic optimization," arXiv:1412.6980 (2017).
42. N. Banic and S. Loncaric, "Unsupervised learning for color constancy," arXiv:1712.00436 (2017).
43. K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," arXiv:1409.1556 (2015).
44. C. Yu, C. Dian-ren, L. Yang, and C. Lei, "Otsu's thresholding method based on gray level-gradient two-dimensional histogram," in *Proceedings of the 2nd International Asia Conference on Informatics in Control, Automation and Robotics (CAR)* (IEEE, 2010), Vol. **3**, pp. 282–285.
45. G. Hemrit, G. D. Finlayson, A. Gijsenij, P. Gehler, S. Bianco, B. Funt, M. Drew, and L. Shi, "Rehabilitating the ColorChecker dataset for illuminant estimation," arXiv:1805.12262 (2018).

Publication 4

Vršnak, D., Domislović, I., Subašić M., Lončarić S., Framework for Illumination Estimation and Segmentation in Multi-Illuminant Scenes, *IEEE Access*, Vol. 11, 2023, pp. 2128-2137

IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 16 December 2022, accepted 31 December 2022, date of publication 4 January 2023, date of current version 9 January 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3234115

RESEARCH ARTICLE

Framework for Illumination Estimation and Segmentation in Multi-Illuminant Scenes

DONIK VRŠNAK, (Graduate Student Member, IEEE), ILIJA DOMISLOVIĆ, (Graduate Student Member, IEEE), MARKO SUBAŠIĆ, (Member, IEEE), AND SVEN LONČARIĆ, (Senior Member, IEEE)

Image Processing Laboratory, Faculty of Electrical Engineering and Computing, University of Zagreb, 10000 Zagreb, Croatia

Corresponding author: Donik Vršnak (donik.vrsnak@fer.hr)

ABSTRACT Color constancy is an important part of the human visual system, as it allows us to perceive the colors of objects invariant to the color of the illumination that is illuminating them. Modern digital cameras have to be able to recreate this property computationally. However, this is not a simple task, as the response of each pixel on the camera sensor is the product of the combination of spectral characteristics of the illumination, object, and the sensor. Therefore, many assumptions have to be made to approximately solve this problem. One common procedure was to assume only one global source of illumination. However, this assumption is often broken in real-world scenes. Thus, multi-illuminant estimation and segmentation is still a mostly unsolved problem. In this paper, we address this problem by proposing a novel framework capable of estimating per-pixel illumination of any scene with two sources of illumination. The framework consists of a deep-learning model capable of segmenting an image into regions with uniform illumination and models capable of single-illuminant estimation. First, a global estimation of the illumination is produced, and is used as input to the segmentation model along with the original image, which segments the image into regions where that illuminant is dominant. The output of the segmentation is used to mask the input and the masked images are given to the estimation models, which produce the final estimation of the illuminations. The models comprising the framework are first trained separately, then combined and fine-tuned jointly. This allows us to utilize well researched single-illuminant estimation models in a multi-illuminant scenario. We show that such an approach improves both segmentation and estimation capabilities. We tested different configurations of the proposed framework against other single- and multi-illuminant estimation and segmentation models on a large dataset of multi-illuminant images. On this dataset, the proposed framework achieves the best results, in both multi-illumination estimation and segmentation problems. Furthermore, generalization properties of the framework were tested on often used single-illuminant datasets. There, it achieved comparable performance with state-of-the-art single-illumination models, even though it was trained only on the multi-illuminant images.

INDEX TERMS Color constancy, segmentation, multi-illuminant, illumination estimation, deep learning, framework.

I. INTRODUCTION

Color constancy is an important part of the human visual system, as it allows us to adapt to different colors of illumination. This enables us to recognize the colors of objects and illuminants independently. For images taken by digital cameras, it is essential to be able to estimate the color of illumination as accurately as possible. Accurate estimation allows us to create a fateful reproduction of the scene which is satisfactory to the human observer. Furthermore, inaccurate estimation creates images that are influenced by illumination, which can decrease the performance of downstream image processing tasks, as described in [1]. Thus, computational color constancy has been studied by numerous authors since

The associate editor coordinating the review of this manuscript and approving it for publication was Essam A. Rashed.

the advent of digital cameras, and many methods have been proposed. Equation (1) describes the amount of light $p_c(x, y)$ recorded for each channel $c \in R, G, B$ at the position x, y in the scene:

$$p_c(x, y) = \int_\omega I(x, y, \lambda) R(x, y, \lambda) S_c(\lambda) d\lambda \,, \tag{1}$$

where $R(\lambda)$ and $I(\lambda)$ are the reflectivity and illumination spectral functions, respectively. S_c represents the spectral sensitivity of the observer (camera). Equation (1) also shows that for each value p_c, there are an infinite number of combinations of I, R, S that can produce it. Since I, R, S are normally unknown, that makes the problem of illumination estimation under constrained.

However, assumptions about the properties of the illumination or the properties of the scene can be introduced. This makes it feasible to approximate the value of the illumination present in the scene. That step of computational color constancy is called illuminant estimation. The second step is to white-balance the image, usually to make it look as it was taken under a canonical illuminant, thus eliminating the influence of the illumination. The most common approximation used is the von Kries [2] model:

$$\begin{bmatrix} c_R \\ c_B \\ c_G \end{bmatrix} = \begin{bmatrix} e_{c,R}/e_{u,R} & 0 & 0 \\ 0 & e_{c,B}/e_{u,B} & 0 \\ 0 & 0 & e_{c,G}/e_{u,G} \end{bmatrix} \begin{bmatrix} p_R \\ p_B \\ p_G \end{bmatrix} \tag{2}$$

where $[c_R \;\; c_B \;\; c_G]^T$ represents the corrected image, and $[p_R \; p_B \; p_G]^T$ is the value retrieved from the sensor. Canonical illumination is represented by $\boldsymbol{e}_c$ and $\boldsymbol{e}_u$ is the estimated illuminant. While Equation (2) does not provide true compensation for the illumination, it is an approximation that works well.

Different assumptions have been applied to the problem of illuminant estimation. One such assumption is that there was only one illuminant present in the scene. However, for many real-world scenes that is not the case. They contain at least two sources of illumination, e.g., outdoor scenes that are illuminated with direct sunlight and with shaded areas illuminated by skylight, or indoor scenes where one illuminant is a light bulb and the other is the sunlight coming through the window. For such scenes, illuminant localization is as important as the estimation, as just the color of the illumination does not provide enough information for accurate correction of the image. Figure 1 shows an example of a real-world scene with two illuminants. The effects of global correction are also shown.

In this work, we propose a novel deep learning framework that is capable of both segmentation and estimation of scenes with two sources of illumination. The main idea behind the framework is to separate the problem of illuminant localization and estimation to different specialized methods. This allows us to utilize well-researched single-illuminant estimation models for multi-illuminant scenes. The framework is composed of three main steps. First, a global illumination vector for the image is estimated. Next, this illumination vector is fed into a segmentation model, alongside the original image. The output of this step is the segmentation mask, which shows where the first estimated illuminant is dominant in the scene. Then, the original image is masked, and the masked images are fed to global estimation models. The outputs of estimation models are combined with the segmentation mask to produce the final estimation of the illumination in the whole scene.

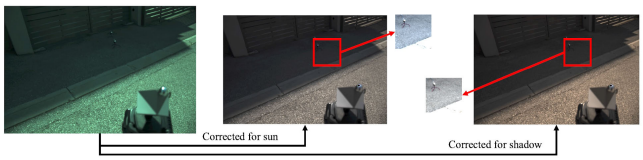


FIGURE 1. Two corrections by different illuminants present in the raw image, with gamma correction applied for easier visualization. The middle image is corrected for the sunlight, and the shaded regions end up having a blue hue. The right image is corrected for illumination in the shadow, which corresponds to the blueish skylight. This gives the sunlit region an orange cast. Groundtruth values were obtained from the gray sides of the SpyderCube calibration object that are highlighted by red squares.

Furthermore, we incorporate the possibility of illuminant mixing, and the proposed framework is capable of providing a per-pixel estimation of illumination. This is achieved by linear combination of estimated illuminants using the segmentation mask. We show that by incorporating joint end-to-end training of the framework, we achieve state-of-the-art results. Additionally, we show that joint training further improves the performance of underlying models when compared to the same models that were only trained separately. The training of the framework was done on a large multi-illuminant dataset [3] containing 2500 indoor and outdoor images. Testing was done on a hold-out set of images from the dataset for the multi-illuminant scenario. The generalization performance was tested by training the framework on multi-illuminant images and testing it on single-illuminant images from Cube+ [4] and ColorChecker [5] datasets. On multi-illuminant images, the proposed framework achieves state-of-the-art results. It also achieves results comparable with best single-illuminant estimation methods on single-illuminant datasets. Furthermore, usage of separate models for each sub-task makes the framework modular. This allows us to easily train and test different variations of the framework, thus balancing the accuracy with complexity. We describe our framework in detail in Section III, and show the quantitative and qualitative results for both the segmentation and per-pixel estimation compared to other single- and multi-illuminant methods in Section IV. Finally, in Section V we conclude the paper.

II. RELATED WORK

The term computational color constancy usually includes two basic steps. These are illumination estimation and color correction (also referred to as chromatic adaptation). The first step is determining the illumination vector for some part

of the image. The granularity of the estimation can vary, from per-pixel, through image patches all the way up to the whole image. This defines the type of estimation method that is needed, with single-illuminant estimation methods estimating only one illuminant for the whole input image. Patch and per-pixel estimations fall under multi-illumination estimation methods, as they estimate more than one illuminant per image. Since color constancy is an ill-posed problem, most research in the past focused on the problem of single-illuminant color constancy. With this assumption, it is assumed that the whole scene (or at least the vast majority) is illuminated by one global illuminant. One of the first methods for single-illuminant color constancy methods were simple methods that relied on low-level image statistics. Two of those methods are the Gray-World [6] and the White-Patch [7] (Max RGB) methods. Gray-World method assumes that for each scene, the average reflectance under white light is gray, and thus any deviation from gray is caused by the color of the illumination. On the other hand, the White-Patch (Max-RGB) method assumes that the brightest part of the scene is the reflected color of the illuminant from a specular surface. However, it is easy to find common real-world examples where these assumptions are broken. For example, for the Gray-World method, any scene with numerous plants (like forests and parks) will not have a gray reflectance under white light but instead that average will be green. For White-Patch, if the scene does not contain any specular highlights, the assumption will be broken. More complex methods were proposed over the years. They can be split into two main categories, statistics based and learning-based methods. Some of the more well-known statistics-based methods include the Gray-Edge framework [8], which generalizes all methods such as Gray-World and White-Patch by adding the possibility of using image gradients and different image norms, as described by (3):

$$\left(\int \left| \frac{\partial^n f_{c,\sigma}(x)}{\partial x^n} \right|^p dx \right)^{\frac{1}{p}} = k e_c^{n,p,\sigma} , \tag{3}$$

where $|\cdot|$ is the Frobenius norm, $c \in R, G, B$, n is the order of the derivative and p is the Minkowski-norm.

There are also gamut mapping methods, such as the method proposed in [9]. There, the goal of the method is to find the gamut that the illuminant spans in the chromaticity diagram and then use that knowledge to find the most probable illuminant color. On the other hand, learning-based methods are more complex, and can be split into two categories: simpler machine learning methods and more complex deep learning methods. One of these learning-based methods [10] learns the common surfaces in the train scenes and then uses the exemplar approach to match the surfaces in the test images to those learned surfaces. Other methods, such as [11], [12], and [13] use a probabilistic model of the illumination and reflectance as a random variable. Unfortunately, all of these methods do not achieve good enough results, particularly in more challenging conditions.

This is the reason more complex deep learning models were proposed for the task of color constancy. The first attempt at such a model was proposed in [14], where a simple network was given a raw image and produced the estimation of the illumination in the scene. Because there was no large dataset, this method was trained mostly on image patches. However, this reduced the semantic information present in each patch, and eliminated cross patch information. This was addressed in [15], where the authors proposed a method that took as input the whole image and produced estimation for patches of the image. Additionally, the method produced an attention map which was used to multiply the patch estimates and produce the final estimation mask. This approach was successful because it allowed the model to reason about the patches of the image that carry more information about the color of the illumination. In [16], the authors propose a very deep model for illuminant estimation (CRNA) that uses cascading residual connections and ResNet architecture to stabilize learning and improve performance. Similarly, in [17], the authors propose a deep network which iteratively estimates the illumination, which is also used to stabilize training and improve performance. On the other hand, in [18], a small network that still achieves state-of-the-art results for illuminant estimation is proposed. Furthermore, some methods, such as [19] and [20], use only image histograms with the deep learning models to perform illuminant estimation. This removes any spatial information and focuses only on colors present in the scene.

On the other hand, multi-illuminant color constancy has been much less studied in the past than single-illuminant color constancy. One reason for this is the lack of a large multi-illuminant dataset, since it is difficult to accurately annotate multi-illuminant images. Most of the methods that were proposed for this problem are learning-based and model the spatial distribution of illuminants. However, several statistics-based methods have been proposed in [21], [22], [23], and [24]. They share some similarity with our approach, as they separate segmentation and estimation into separate tasks that are combined. They use image texture [23] or Kmeans [22] for localization and then use Max RGB method for estimation. Finally, similar to our method, the localization is used to compute the final per-pixel illumination of the scene. On the other hand, [25] propose a white-balancing method for scenes in which the total number of illuminants is not known. They achieve this by selecting N white-balance points and map them to ground truth ones. Finally, [26] proposed a method that imitates the Adaptive Surround Modulation (ASM) capability of the human eye to regulate the receptive field of neurons based on contrast. One classical machine learning approach was presented in [27], where the authors use conditional random fields to create the MIRF algorithm, which can localize and estimate illuminants in the scene. The main drawback of this approach is its high computational cost and lower accuracy. Deep learning-based approach for multi-illuminant color constancy was proposed in [28] as an upgrade on the network proposed in [14], where

the authors use kernel density estimation to determine the number of illuminants in the scene. In [29], authors propose a framework of two networks, HypNet and SelNet. HypNet network proposes two hypotheses about the illumination of each patch, and SelNet chooses which of those hypotheses to use for the estimation.

More recently, in [30] the authors proposed a simple model that used brightness threshold to perform image segmentation, to which they applied simple estimation methods. This method works very fast, but it produces many artifacts and incorrect corrections in parts of the scene where the brightness assumption does not hold. Furthermore, three methods for image segmentation and estimation using deep learning models were proposed. In [31], the authors introduced a vision transformer method that was able to perform segmentation of parts of the scene that were incorrectly white-balanced. In [32], the authors proposed an autoencoder training strategy and a novel loss function which was capable of learning the common distribution of colors in scenes, to produce per-pixel estimation of the illumination. Finally, in [33] the authors created a segmentation model that was able to segment scenes with two sources of illumination by first producing an estimation of the primary illuminant. We based our framework on the same principle: that it is possible to relatively accurately estimate one of the illuminant sources in the scene using global methods, and then localize its influence. However, unlike the model in [33] we do not stop at segmentation, as our framework allows for accurate estimation of both illuminant sources and their localization.

III. PROPOSED FRAMEWORK

In this work, we present a novel framework for simultaneous estimation and segmentation of illumination for scenes with two sources of illumination. The main idea behind our framework was to leverage well-researched single-illuminant estimation models for multi-illuminant scenes. The proposed framework consists of three main parts. A scheme of the framework can be found in Figure 2. The first part is global estimation of the dominant illuminant. Then, a segmentation model is used to localize the influence of the dominant illuminant, which is represented as a binary segmentation mask. This mask is used to create masked inputs for the two estimation models. Then, those two single-illuminant estimation models are used to estimate the dominant and secondary illuminants. Finally, the per-pixel estimation of illumination for the scene is obtained by linear combination of the estimated illuminants based on the weights from the segmentation output, using Equation (4):

$$p(x, y)_c = (1 - S_p(x, y))I_{\mathrm{p1}} + S_p(x, y)I_{\mathrm{p2}}\,, \tag{4}$$

where (x, y) are the coordinates in the image, p_c is the final per-pixel estimation, I_{p1} and I_{p2} are dominant and secondary illumination estimations, respectively, and S_p is the predicted segmentation mask.

Each layer of the framework is implemented so that it allows for the free flow of gradients using backpropagation. This allows us to train the framework end-to-end. We refer to this as joint training. Gradients in the upper layers during training of the framework are computed from both the estimation and the segmentation errors. This effect is not present when layers are only trained separately. Another benefit of this approach is in the transitional regions between the illuminations. In those regions, the segmentation model is encouraged to keep the output such that the linear combination of the illumination sources corresponds to the real mixed illumination. Thus, the segmentation output is pushed closer to 0.5 than to 0 or 1 for those areas. In the case of the pure segmentation training, where the goal is to create hard borders between classes, no such regularization effect is present. Furthermore, those regions carry less useful information for either of the single-illuminant estimation models that come after the segmentation. For them, this ambiguity in the segmentation acts as an attention mechanism, by shifting focus more to the parts of the scene where illumination is less ambiguous. We show later that this type of joint training of our framework improves the performance of both segmentation and estimation model compared with their counterparts that were trained independently.

Moreover, we propose an additional recurrent component because it can sometimes be difficult to estimate the dominant illuminant from the whole image in the first step. The recurrent connection is shown with a labeled dotted arrow in the red part of Figure 2. It naturally follows that, if we can localize and estimate one illuminant in the scene, the estimation produced would be better than the global estimation. Thus, the recurrent component enables additional passes through the framework. In the second pass through the framework, the recurrent connection replaces the initial global dominant illuminant estimation with the output of the local dominant illuminant estimation from the first pass. For the final output of the framework, all the intermediary estimation and segmentation steps are averaged. Such recurrent behavior can be implemented in as many steps as it is necessary. However, since the task of color constancy usually needs to be performed quickly, we implemented only a two-step recurrent framework. We compare the performance of this recurrent framework to that of the base framework as well as other multi- and single-illuminant models in Section IV.

For the estimation task, the framework is designed in such a way that it is interoperable with any state-of-the-art single-illuminant estimation methods. In the scope of this paper, we implemented a single-illuminant estimation model based on the FC4 [15] model, with a reduced number of parameters. We reduced the number of parameters to decrease the overall complexity of the framework. We use one of these models to first predict the dominant illuminant in the scene. Later, we use two more such models to predict the illumination in the regions highlighted by the segmentation model. Furthermore, in some variations of our framework, the weights are shared between these two models. (In practice, this is implemented with only one estimation model, to reduce memory

usage.) These estimation models are shown in yellow in Figure 2.

Finally, we limit the number of illuminants for two main reasons. Firstly, we are limited by the types of datasets that are available for multi-illuminant scenes, which are needed to train our model. All the labeled datasets that have per-pixel groundtruth information about the illuminants in the scene contain only two illuminants. Moreover, most real-world scenes actually contain either one or two illuminants. One exception are very dynamic nighttime scenes like clubs or urban areas. However, we show that even with this reduction in the number of illuminants, our model can handle complex scenes. We achieve this by allowing illuminant mixing, which is very common in real-world scenes. Furthermore, the results show that our model preforms well on single-illuminant scenes, even though it was trained only on scenes with two sources of illumination.

A. TRAINING

The framework was trained in two steps. First, each component of the framework was trained on their respective task separately. The segmentation part was trained to segment the areas of the scene where the primary illuminant was dominant, similar to the method proposed in [33]. The estimation models were trained to predict either the dominant or the secondary illuminant. After the pretraining step, the framework was combined into the final model as described in Section III and then trained end to end using backpropagation. The framework was implemented in TensorFlow 2.4 and trained on a system with an RTX 2080Ti GPU and AMD Ryzen 3700x CPU. Pretraining was done over 500 epochs, with cosine annealing scheduler [34] and stochastic gradient descent [35] optimizer. We use a linear combination of the binary cross entropy (BCE) and robust color constancy loss (IL) function [36] for the segmentation and estimation outputs, respectively. This combined loss can be expressed as:

$$\begin{aligned}&\mathrm{L}(I_{\mathrm{p0}}, I_{\mathrm{p1}}, I_{\mathrm{p2}}, S_p, I_{\mathrm{gt1}}, I_{\mathrm{gt2}}, S_{\mathrm{gt}})\\&\quad = \alpha \mathrm{IL}(I_{\mathrm{p0}}, I_{\mathrm{gt1}})\\&\qquad + \beta \mathrm{BCE}(S_p, S_{\mathrm{gt}}) + \gamma \mathrm{IL}(I_{\mathrm{p1}}, I_{\mathrm{gt1}}) + \delta \mathrm{IL}(I_{\mathrm{p2}}, I_{\mathrm{gt2}}) \quad (5)\\&\mathrm{BCE}(S_p, S_{\mathrm{gt}})\\&\quad = -S_{gt} \log(S_p) - (1 - S_{\mathrm{gt}}) \log(1 - S_p) \quad (6)\\&\mathrm{IL}(I_p, I_{\mathrm{gt}})\\&\quad = \left\| \frac{I_p - I_{\mathrm{gt}}}{I_{\mathrm{gt}}} \right\|_2, \quad (7)\end{aligned}$$

where I_{p0} is the initial estimation of the dominant illuminant, I_{p1} and I_{p2} are the final estimations of the dominant and secondary illuminant, and S_p is the predicted segmentation mask. I_{gt1}, I_{gt2}, and S_{gt} are the groundtruth information about the illuminants and the segmentation mask, respectively. BCE (Equation (6)) is the binary cross entropy function applied at the pixel level. The IL (Equation (7)) loss function is the robust color constancy loss function proposed in [36]. Coefficients α, β, γ, and δ were selected using random search of the hyperparameter space and their values were set to 0.7, 1.0, 0.9, and 0.9 respectively.

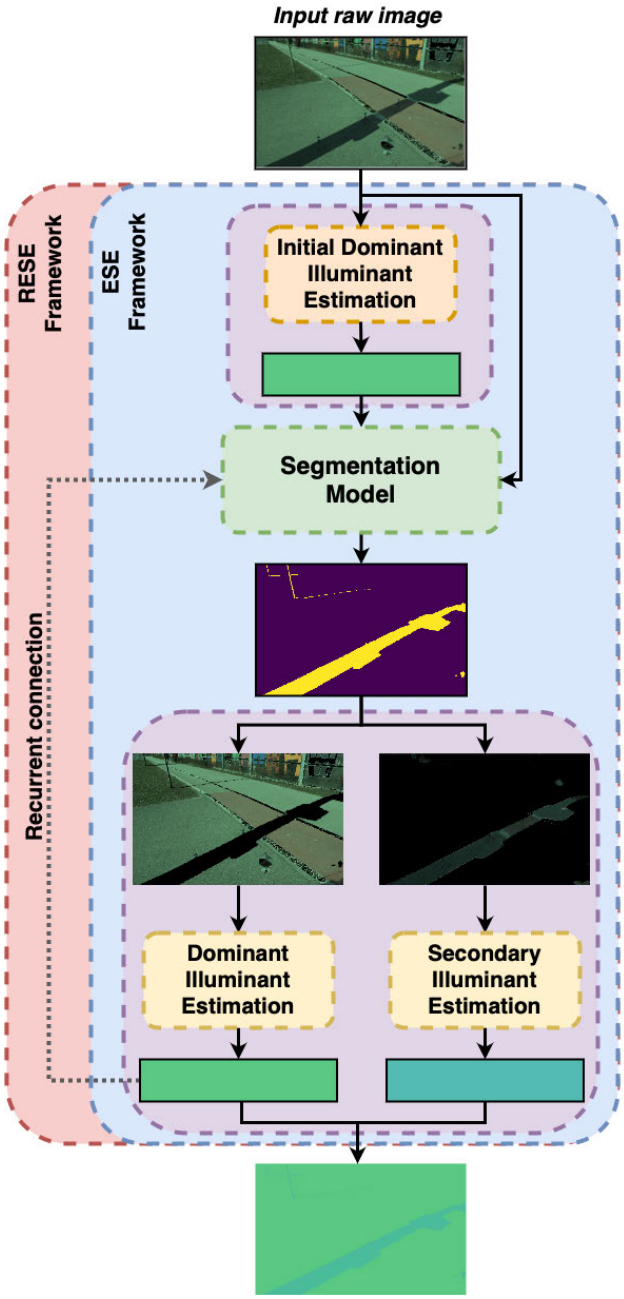


FIGURE 2. Scheme of the proposed framework. In general, the framework consists of the initial estimator of the dominant illuminant, followed by a segmentation model that is capable of localizing the presence of that illuminant in the scene. Then that output is used to create two masked images, which are then given to the estimation models. The estimation models then produce two estimations that are combined to create the final per-pixel estimation of the illumination. The estimation models in the bottom purple box can either be independent or have shared weights. The recurrent extension to our framework is shown in red. The dotted line represents the recurrent connection that allows us to use the dominant illuminant estimation as the input to the segmentation model in the second pass.

Joint training of the framework was done using the same scheduler and optimizer for another 500 epochs. To provide a fair comparison, models that were not trained jointly were all trained for 1000 epochs to eliminate any problems with

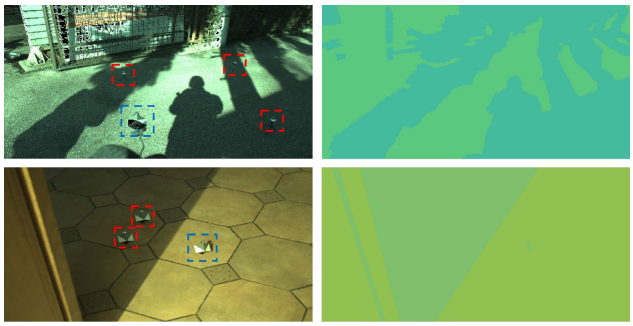


FIGURE 3. Example of the images used for training. The first and second rows show an outdoor and indoor scene, with 4 and 3 SpyderCube calibration objects, respectively. Multiple cubes marked by red squares are placed in the region illuminated by the ambient illuminant, which can vary throughout the scene more than the direct illumination (e.g., sunlight or one light bulb). We used only images where the difference in the ground truth between the measured ambient illumination was less than 1 degree to ensure that the manual annotation of the regions shown in the second column is accurate. Our annotation procedure is similar to that described in [24].

under fitting. The parameters of the model that scored the best on the validation set were taken for testing to prevent overfitting. For the training, we used a newly constructed dataset [3] containing 2500 outdoor and indoor scenes with two sources of illumination, taken by 5 different cameras. All images were manually annotated to contain per-pixel groundtruth illumination values. Few examples of images and the groundtruth from this dataset are shown in Figure 3.

B. PERFORMANCE MEASURES

We compare the model performance on a hold-out set of the two-illuminant dataset on both the segmentation and illumination tasks. For the single-illuminant datasets, we compare only the performance of illuminant estimation. To quantitatively compare the results, we use two metrics, Dice coefficient [37] for segmentation and angular distance for illuminant estimation.

Dice coefficient [37] is computed as:

$$\text{Dice} = \frac{2|\text{TP}|}{2|\text{TP}| + |\text{FP}| + |\text{FN}|}, \tag{8}$$

where TP, FP, FN are true positive, true negative and false negative values when comparing the prediction to the groundtruth. $|\cdot|$ represents the cardinality (number of elements) of the set.

For the estimation task, we use angular error, which can be computed as:

$$\text{err}_{\text{ang}} = \cos^{-1}\left(\frac{\boldsymbol{e}_r \cdot \boldsymbol{e}_p}{\boldsymbol{e}_r \boldsymbol{e}_p}\right), \tag{9}$$

where $\cdot$ denotes vector dot product, $\boldsymbol{e}_r$ is the real illuminant and $\boldsymbol{e}_p$ the estimated illuminant. Since the groundtruth and estimation are pixel-based, we report the average error over the whole image. The classes in the segmentation masks are relatively well-balanced, so the average value of the error is not biased towards either illuminant. For single-illuminant comparison, our model was only trained on the multi-illuminant images, and then tested on the images from the single-illuminant dataset. In this case, we obtain the single-illuminant estimate by applying global average pooling to the per-pixel illuminant estimations.

IV. RESULTS

The models were tested on a hold out set of our dataset [3], and on single-illuminant images from the Cube+ [4] and ColorChecker [5] datasets. Thus, we test the performance of our model in both single- and multi-illuminant scenarios. In the case of the single-illuminant images, the models were trained only on the images from our two-illuminant dataset, and then tested as is on the single-illuminant images. The framework was compared to other methods for both multi-illuminant segmentation and estimation tasks, and these results are shown in Tables 1 and 2 respectively. The comparison of results on single-illuminant images are shown in Tables 3 and 4.

Table 1 shows the results of the segmentation task. The first block of models are the simple baseline models, the second block is the segmentation models implemented from other works. The third block presents the variations of the proposed framework. They show that our framework outperform all other implemented models, and by a solid margin, independent of the number of parameters. The models that were used for comparison include the illumination segmentation models proposed in [32], [33], and [30], U-Net [38] models with VGG-16 and VGG-19 [39] encoders (implemented such that one illuminant was known, as described in [33]) and a baseline Otsu threshold applied to the brightness histogram of the image. It is important to note that the framework performs better than the pure segmentation models (VGG-16). This holds even when the number of parameters is comparable (approx. 34 million parameters in the case of the VGG-19 based autoencoder and VGG-16 based framework). This indicates that the joint training that was used to train our framework increases both the segmentation and estimation parts of our model. To further test this, we compare the jointly trained framework to one whose components were trained only separately (i.e., no joint training was done). Again, we see the improvement in performance, thus providing further evidence of the benefit of joint training (seen in the last block in Table 1). We denote the frameworks where the parameters of the estimation models are shared by omitting the "x2" modifier in the name. RESE denotes the recurrent variant of our framework with two steps.

Since our framework is primarily designed to produce a per-pixel estimate of the illumination, the main focus will be on those results. Table 2 shows the estimation results on our dataset with two illuminants for many multi- and single-illuminant methods that were implemented. In it, the first block of models are the simple baseline models single- and multi-illuminant estimation models. The second block contains the estimation models implemented from other works. The third block contains variations of the framework that were not jointly trained. Finally, the fourth block contains variations of the proposed framework with joint training.

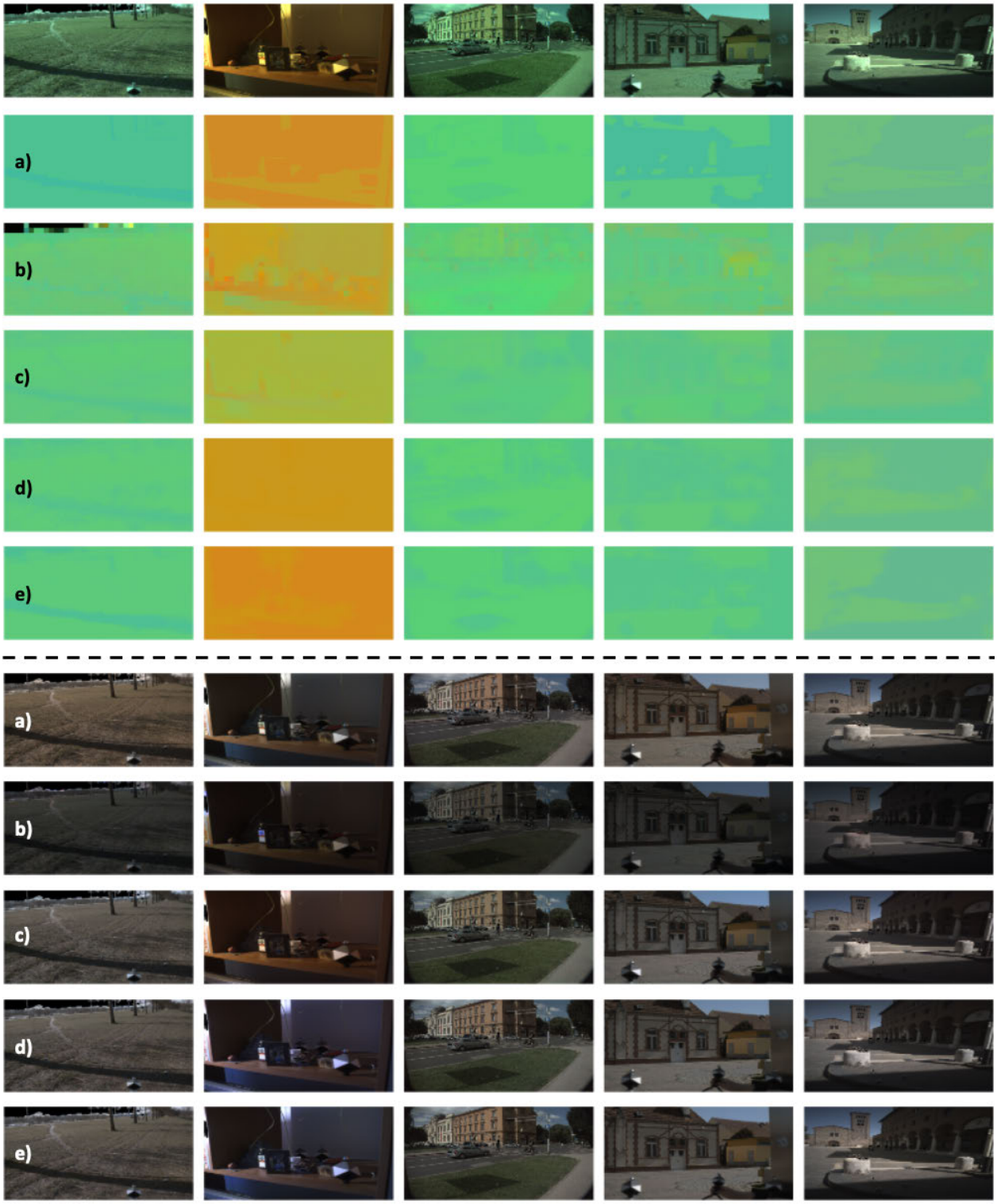


FIGURE 4. Example of the randomly selected images, corresponding groundtruths, estimations and corrections for multi-illuminant estimation methods. The first row is the input image, the first set of images are the per pixel illumination estimations, while the second set are the corrections. In each section, rows are marked with letters corresponding to different models and groundtruth. These are: (a) Groundtruth, (b) Bianco-CNN [28], (c) Autoencoder-based [32] (VGG-16), (d) VGG-16 + FC4 × 2 (non jointly trained framework), and (e) ESE(VGG-16 + FC4 × 2).

These results were obtained by computing the angular error (Equation (9)) between each pixel in the estimated per-pixel map and the groundtruth mask. It can be seen that all the jointly trained models significantly outperform the other

TABLE 1. Dice coefficient (Equation (8)) results of the models for the illuminant segmentation task. The names in parentheses show the base models used (in the case of our framework, the segmentation model is named first, followed by the estimation model). The best results are shown in bold. (Higher is better.)

Model	Mean	Median	Trimean	Best 25%	Worst 25%
Treshold [30]	0.73	0.73	0.73	0.86	0.58
Otsu [40]	0.80	0.83	0.82	0.95	0.63
Bianco-CNN [28]	0.79	0.78	0.78	0.93	0.66
Autoencoder-based [32] (VGG-16)	0.86	0.88	0.88	0.95	0.72
Autoencoder-based [32] (VGG-19)	0.82	0.84	0.83	0.93	0.66
T-large [31]	0.88	0.90	0.89	0.96	0.76
Seg [33]	0.89	0.91	0.90	0.97	0.76
VGG-16 [39]	0.88	0.90	0.90	0.96	0.76
ESE(Seg [33] + FC4)	0.89	0.92	0.92	0.97	0.77
ESE(Seg [33] + FC4x2)	0.89	0.92	0.92	0.97	0.77
ESE(VGG-16 + FC4)	**0.91**	**0.93**	**0.93**	**0.98**	**0.81**
ESE(VGG-16 + FC4x2)	0.90	0.93	0.92	0.97	0.79
RESE(VGG-16 + FC4x2)	**0.91**	**0.93**	0.92	0.97	0.80

TABLE 2. Angular error (Equation (9)) of the results of the models for the multi-illuminant estimation task. The names in parentheses show the base models used (in the case of our framework, the segmentation model is named first, followed by the estimation model, "x2" indicates two estimation models). The best results are shown in bold. (Lower is better.)

Model	Mean	Median	Trimean	Best 25%	Worst 25%
Gray-world [6]	5.55	5.37	5.45	2.54	8.84
1st-order Gray-Edge [8]	5.44	5.63	5.51	1.98	8.93
2nd-order Gray-Edge [8]	5.77	5.71	5.75	2.22	9.45
Sub-blocks Max-RGB [21]	5.71	4.94	5.12	2.57	10.17
ASM(Single) [26]	2.88	2.67	2.66	1.34	4.89
FC4(Squeezenet) [15]	4.09	3.85	3.91	1.78	6.76
ASM(Multi) [26]	22.72	23.14	22.72	10.18	35.40
MIRF [27]	7.08	6.20	6.40	2.70	12.67
Mixed-Illuminant CC [22]	8.25	7.59	7.74	3.81	13.67
Patch-based (White-Patch) [24]	4.3	2.89	-	1.02	10.13
Keypoint-based (White-Patch) [24]	5.46	3.59	-	1.11	13.15
Superpixel-based(2nd Order) [24]	4.2	3.1	-	1.09	9.32
DS-Net (HypNet+SelNet) [29]	6.31	3.95	-	0.85	15.95
Bianco-CNN [28]	4.65	4.42	4.42	2.63	7.27
Autoencoder-based [32] (VGG-16)	3.59	2.99	3.08	1.81	6.52
Autoencoder-based [32] (VGG-19)	4.38	3.82	3.99	2.45	7.26
Seg [33] + FC4x2	2.97	2.53	2.60	1.41	5.30
VGG-16 + FC4x2	2.98	2.51	2.59	1.45	5.33
ESE(Seg [33] + FC4)	2.66	2.17	2.25	1.18	5.00
ESE(Seg [33] + FC4x2)	2.56	**2.11**	2.21	**1.13**	4.74
ESE(VGG-16 + FC4)	2.58	**2.11**	2.21	1.22	4.75
ESE(VGG-16 + FC4x2)	**2.55**	2.12	**2.18**	1.19	**4.70**
RESE(VGG-16 + FC4x2)	2.64	2.23	2.31	1.24	4.81

models, with the largest margin of almost 0.5 degrees (14.5% improvement compared to the second best performing multi-illuminant model). It is also important to note that the smallest framework (composed of the small segmentation model [33] and shared reduced FC4 model) still outperforms other models. Furthermore, the models with the independent estimation estimators outperform their counterparts with shared estimators, at the cost of more parameters. This shows that different tradeoffs regarding accuracy, memory usage and speed can be implemented. Figure 4 shows the qualitative comparison of the segmentation and estimation results on images with two sources of illumination from our dataset.

Finally, we tested the performance of our framework on two commonly used single-illuminant datasets, the Cube+ [4] and ColorChecker [5] and compared it to other state-of-the-art methods. We show these results in Tables 3 and 4. The results show that, while some specialized single-illuminant learning-based models outperform our framework, it achieves by far the best results out of all tested multi-illuminant

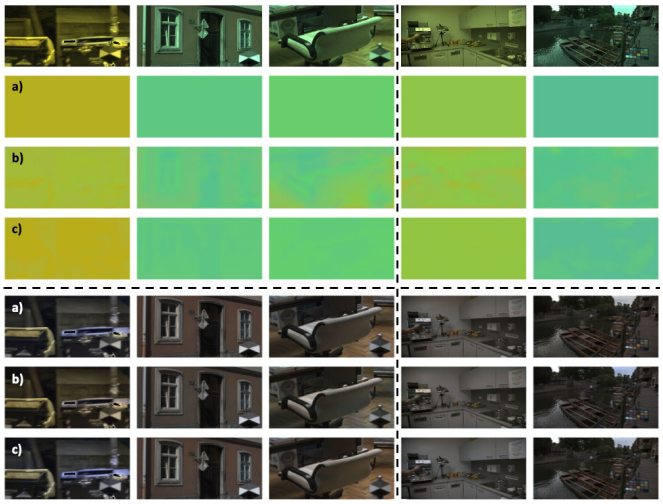


FIGURE 5. Example of the randomly selected images from the single-illuminant datasets, corresponding groundtruths, estimations and corrections. The first row is the input image, the first set of images are the per pixel illumination estimations, while the second set are the corrections. The first three columns correspond to images from the Cube+ [4] dataset, and the rest correspond to the ColorChecker [5] dataset. In each section, rows are marked with letters corresponding to different models and groundtruth. These are: (a) Groundtruth, (b) Autoencoder-based [32] (VGG-16), and (c) ESE(VGG-16 + FC4 × 2).

TABLE 3. Angular error (Equation (9)) of the results of the models for the single-illuminant estimation task on the Cube+ dataset [4]. The best results are shown in bold. The best performing multi-illuminant model is highlighted in yellow. Data for single-illuminant models was obtained from [18] (Lower is better).

Model	Mean	Median	Trimean	Worst 25%	Best 25%
White-Patch [7]	9.69	7.48	8.56	20.49	1.72
Gray-world [6]	7.71	4.29	4.98	20.19	1.01
Shades-of-gray [42]	2.59	1.73	1.93	6.19	0.46
1st-order Gray-Edge [8]	2.41	1.52	1.72	5.89	0.45
2nd-order Gray-Edge [8]	2.5	1.59	1.78	6.08	0.48
FFCC(model J) [20]	1.38	0.74	0.89	3.67	**0.19**
FC4(Squeezenet) [15]	1.35	0.93	1.01	3.24	0.3
Koscevic et. al.(VGG-16) [17]	1.34	0.83	0.97	3.2	0.28
MDLCC [43]	1.24	0.83	0.92	2.91	0.26
One-Net [18]	**1.21**	**0.72**	**0.83**	**3.05**	0.21
Autoencoder-based [32] (VGG-16)	4.21	3.99	4.03	7.18	1.85
Autoencoder-based [32] (VGG-19)	3.62	2.79	3.08	6.94	1.6
Bianco-CNN [28]	4.82	4.28	4.42	8.05	2.54
ESE(VGG-16 + FC4)	1.68	1.3	1.38	3.59	0.44
ESE(VGG-16 + FC4x2)	2.01	1.81	1.84	3.67	0.72

models. Furthermore, those results are still comparable with the best single-illuminant models, and the difference even in worst cases is less than the perceptual sensitivity of the human eye described in [41]. It is also important to note that all the single-illuminant models were trained on these datasets. However, our framework was trained on our multi-illuminant dataset and then only tested on these two single-illuminant datasets. This shows that our framework generalizes well over different images, as it is the only one of the multi-illuminant models that was able to achieve comparable results with the best single-illuminant models. Figure 5 provides a qualitative evaluation of the performance of our framework on single-illuminant datasets. It can be seen there that, even though the datasets are supposedly single-illuminant, some scenes do contain multiple illuminants, and that our model is capable of detecting this (second and last column).

TABLE 4. Angular error (Equation (9)) of the results of the models for the single-illuminant estimation task on the ColorChecker dataset [5]. The best results are shown in bold. The best performing multi-illuminant model is highlighted in yellow. Data for single-illuminant models was obtained from [15] and [16]. (Lower is better.)

Model	Mean	Median	Trimean	Worst 25%	Best 25%
White-Patch [7]	7.55	5.68	6.35	16.12	1.45
Gray-World [6]	6.36	6.28	6.28	10.58	2.33
1st-order Gray-Edge [8]	5.33	4.52	4.73	10.03	1.86
2nd-order Gray-Edge [8]	5.13	4.44	4.62	9.26	2.11
Shades-of-Gray [42]	4.93	4.01	4.23	10.2	1.14
Bayesian [12]	4.82	3.46	3.88	10.49	1.26
Exemplar based [10]	3.1	2.3	-	-	-
Regression Tree [13]	2.42	1.65	1.75	5.87	0.38
CRNA [16]	1.99	**1.01**	1.33	**3.2**	**0.22**
CCC (dist+ext) [19]	1.95	1.22	1.38	4.76	0.35
DS-Net (HypNet+SelNet) [29]	1.9	1.12	1.33	4.84	0.31
AlexNet-FC4 [15]	1.77	1.11	1.29	4.29	0.34
SqueezeNet-FC4 [15]	**1.65**	1.18	**1.27**	3.78	0.38
Autoencoder-based [32] (VGG-16)	4.78	4.29	4.41	7.92	2.41
Autoencoder-based [32] (VGG-19)	5.37	5.04	5.09	8.81	2.67
Bianco-CNN [28]	5.85	5.5	5.58	8.92	3.34
ESE(VGG-16 + FC4)	3.49	2.9	3.05	6.49	1.51
ESE(VGG-16 + FC4x2)	2.34	1.84	1.94	4.94	0.64

V. CONCLUSION

In this work, we presented a novel framework that is capable of segmenting and estimating illumination in scenes with one or two primary sources of illumination. The proposed framework is composed of specialized models for each task. First, a global estimation model is used to estimate the dominant illuminant in the scene. Then, a segmentation model is used to localize the influence of the estimated global illuminant. This produces regions of influence of illuminants, and the input image is masked using this segmentation. The masked images are then passed to estimation models that produce the estimation for those unmasked regions of the scene. The final estimation is done by linear combination of the estimated illuminants using the segmentation mask. Moreover, the proposed framework is modular as the estimation and segmentation models can easily be replaced, offering different tradeoffs in speed, memory, and accuracy.

The framework was tested on the novel dataset with 2500 images of varied indoor and outdoor scenes taken by 5 different cameras [3]. Our framework achieved the best results by a large margin, especially in the illuminant estimation task, with a 14.5% improvement above the second best scoring multi-illuminant model. We have also tested our framework on images from the Cube+ [4] and ColorChecker [5] single-illuminant datasets. For this task, we did not retrain the framework, but have used the best performing models from the multi-illuminant task. Here, our framework achieves excellent results, only slightly worse than specialized state-of-the-art single-illuminant estimation models. This shows the excellent generalization properties of our framework on cross dataset tasks.

REFERENCES

[1] M. Afifi and M. Brown, "What else can fool deep learning? Addressing color constancy errors on deep neural network performance," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 243–252.

[2] J. Von Kries, "Influence of adaptation on the effects produced by luminous stimuli," *Handbuch Der Physiologie Des Menschen*, vol. 3, pp. 109–282, Jan. 1905.

[3] I. Domislović, D. Vršnak, M. Subašić, and S. Loncarić. Accessed: Dec. 7, 2022. *Cube2: Large Mutli-Illuminant Dataset*. [Online]. Available: https://github.com/donikv/Cube2

[4] N. Banic and S. Loncaric, "Unsupervised learning for color constancy," 2017, *arXiv:1712.00436*.

[5] G. Hemrit, G. D. Finlayson, A. Gijsenij, P. Gehler, S. Bianco, B. Funt, M. Drew, and L. Shi, "Rehabilitating the colorchecker dataset for illuminant estimation," in *Proc. Color Imaging Conf.*, 2018, pp. 350–353.

[6] G. Buchsbaum, "A spatial processor model for object colour perception," *J. Franklin Inst.*, vol. 310, pp. 1–26, Jan. 1980.

[7] E. H. Land, "The retinex theory of color vision," *Sci. Amer.*, vol. 237, no. 6, pp. 108–129, 1977.

[8] J. Van De Weijer, T. Gevers, and A. Gijsenij, "Edge-based color constancy," *IEEE Trans. Image Process.*, vol. 16, no. 9, pp. 2207–2214, Sep. 2010.

[9] A. Gijsenij, T. Gevers, and J. Van De Weijer, "Generalized gamut mapping using image derivative structures for color constancy," *Int. J. Comput. Vis.*, vol. 86, no. 2, pp. 127–139, Jan. 2010.

[10] H. R. V. Joze and M. S. Drew, "Exemplar-based color constancy and multiple illumination," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 5, pp. 860–873, May 2014.

[11] B. Funt and W. Xiong, "Estimating illumination chromaticity via support vector regression," in *Proc. Color Image Conf.*, vol. 50, Jan. 2004, pp. 47–52.

[12] P. V. Gehler, C. Rother, A. Blake, T. Minka, and T. Sharp, "Bayesian color constancy revisited," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2008, pp. 1–8.

[13] V. Agarwal, A. V. Gribok, A. Koschan, and M. A. Abidi, "Estimating illumination chromaticity via kernel regression," in *Proc. Int. Conf. Image Process.*, Oct. 2006, pp. 981–984.

[14] S. Bianco, C. Cusano, and R. Schettini, "Single and multiple illuminant estimation using convolutional neural networks," *IEEE Trans. Image Process.*, vol. 26, no. 9, pp. 4347–4362, Sep. 2017.

[15] Y. Hu, B. Wang, and S. Lin, "FC4: Fully convolutional color constancy with confidence-weighted pooling," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 4085–4094.

[16] H.-H. Choi and B.-J. Yun, "Very deep learning-based illumination estimation approach with cascading residual network architecture (CRNA)," *IEEE Access*, vol. 9, pp. 133552–133560, 2021.

[17] K. Koscevic, M. Subasic, and S. Loncaric, "Iterative convolutional neural network-based illumination estimation," *IEEE Access*, vol. 9, pp. 26755–26765, 2021.

[18] I. Domislovic, D. Vrsnak, M. Subasic, and S. Loncaric, "One-Net: Convolutional color constancy simplified," *Pattern Recognit. Lett.*, vol. 159, pp. 31–37, Jul. 2022.

[19] J. T. Barron, "Convolutional color constancy," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 379–387.

[20] J. T. Barron and Y.-T. Tsai, "Fast Fourier color constancy," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 886–894.

[21] M. A. Hussain and A. S. Akbari, "Max-RGB based colour constancy using the sub-blocks of the image," in *Proc. 9th Int. Conf. Develop. eSyst. Eng. (DeSE)*, Aug. 2016, pp. 289–294.

[22] M. A. Hussain and A. S. Akbari, "Color constancy algorithm for mixed-illuminant scene images," *IEEE Access*, vol. 6, pp. 8964–8976, 2018.

[23] M. A. Hussain, A. Sheikh-Akbari, and E. A. Halpin, "Color constancy for uniform and non-uniform illuminant using image texture," *IEEE Access*, vol. 7, pp. 72964–72978, 2019.

[24] A. Gijsenij, R. Lu, and T. Gevers, "Color constancy for multiple light sources," *IEEE Trans. Image Process.*, vol. 21, no. 2, pp. 697–707, Feb. 2012.

[25] T. Akazawa, Y. Kinoshita, S. Shiota, and H. Kiya, "N-white balancing: White balancing for multiple illuminants including non-uniform illumination," *IEEE Access*, vol. 10, pp. 89051–89062, 2022.

[26] A. Akbarinia and C. A. Parraga, "Colour constancy beyond the classical receptive field," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 9, pp. 2081–2094, Sep. 2018.

[27] S. Beigpour, C. Riess, J. Van De Weijer, and E. Angelopoulou, "Multi-illuminant estimation with conditional random fields," *IEEE Trans. Image Process.*, vol. 23, no. 1, pp. 83–96, Jan. 2014.

[28] S. Bianco, C. Cusano, and R. Schettini, "Color constancy using CNNs," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2015, pp. 81–89.

[29] W. Shi, C. C. Loy, and X. Tang, "Deep specialized network for illuminant estimation," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 371–387.
[30] S.-H. Lee, S.-M. Woo, J.-H. Choi, and J.-O. Kim, "Two-step multi-illuminant color constancy for outdoor scenes," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2017, pp. 710–714.
[31] D. Vršnak, I. Domislović, M. Subašić, and S. Loncarić, "Illuminant estimation error detection for outdoor scenes using transformers," in *Proc. 12th Int. Symp. Image Signal Process. Anal. (ISPA)*, Sep. 2021, pp. 276–281.
[32] D. Vršnak, I. Domislović, M. Subašić, and S. Loncarić, "Autoencoder-based training for multi-illuminant color constancy," *J. Opt. Soc. Amer. A, Opt. Image Sci.*, vol. 39, pp. 1076–1084, Jun. 2022.
[33] D. Vršnak, I. Domislović, M. Subašić, and S. Loncarić, "Illuminant segmentation for multi-illuminant scenes using latent illumination encoding," *Signal Process., Image Commun.*, vol. 108, Oct. 2022, Art. no. 116822.
[34] I. Loshchilov and F. Hutter, "SGDR: Stochastic gradient descent with warm restarts," 2017, *arXiv:1608.03983*.
[35] S.-I. Amari, "Backpropagation and stochastic gradient descent method," *Neurocomputing*, vol. 5, nos. 4–5, pp. 185–196, 1993.
[36] Z. Li and Z. Ma, "Robust white balance estimation using joint attention and angular loss optimization," in *Proc. 13th Int. Conf. Mach. Vis.*, Jan. 2021, pp. 401–406.
[37] K. Zou, S. Warfield, A. Bharatha, C. Tempany, M. Kaus, S. Haker, W. Wells, F. Jolesz, and R. Kikinis, "Statistical validation of image segmentation quality based on a spatial overlap index," *Academic Radiol.*, vol. 11, pp. 89–178, Feb. 2004.
[38] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent. (MICCAI)* (Lecture Notes in Computer Science), vol. 9351. Cham, Switzerland: Springer, Nov. 2015, pp. 234–241.
[39] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2015, *arXiv:1409.1556*.
[40] C. Yu, C. Dian-ren, L. Yang, and C. Lei, "Otsu's thresholding method based on gray level-gradient two-dimensional histogram," in *Proc. 2nd Int. Asia Conf. Informat. Control, Autom. Robot. (CAR)*, Mar. 2010, p. 282.
[41] A. Gijsenij, T. Gevers, and M. P. Lucassen, "Perceptual analysis of distance measures for color constancy algorithms," *J. Opt. Soc. Amer. A, Opt. Image Sci.*, vol. 26, no. 10, pp. 2243–2256, 2009.
[42] G. D. Finlayson and E. Trezzi, "Shades of gray and colour constancy," in *Proc. IST/SID Color Imag. Conf.*, vol. 1, Jan. 2004, pp. 37–41.
[43] J. Xiao, S. Gu, and L. Zhang, "Multi-domain learning for accurate and few-shot color constancy," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 3258–3267.

DONIK VRŠNAK (Graduate Student Member, IEEE) received the B.Sc. and M.Sc. degrees in computer science, in 2018 and 2020, respectively. He is currently pursuing the Ph.D. degree in computing with the Faculty of Electrical Engineering and Computing, University of Zagreb, Croatia. He is also doing research on color constancy, focused primarily on multi-illuminant segmentation and the perceptional properties of the human visual system related to color constancy. His research interests include image processing and analysis, bioinformatics, astronomy, and machine learning.

ILIJA DOMISLOVIĆ (Graduate Student Member, IEEE) received the B.Sc. and M.Sc. degrees in computer science, in 2018 and 2020, respectively. He is currently pursuing the Ph.D. degree in scientific field of computing (technical sciences) with the Faculty of Electrical Engineering and Computing, University of Zagreb, Croatia. His research interests include image processing, image analysis, neural networks, and color constancy, with a focus on illumination estimation.

MARKO SUBAŠIĆ (Member, IEEE) received the Ph.D. degree from the Faculty of Electrical Engineering and Computing, University of Zagreb, in 2007. Since 1999, he has been working with the Department for Electronic Systems and Information Processing, Faculty of Electrical Engineering and Computing, University of Zagreb, where he is currently an Associate Professor. He teaches several courses at the graduate and undergraduate levels. His research interests include image processing and analysis and neural networks, with a particular interest in image segmentation, detection techniques, and deep learning. He is a member of the IEEE Computer Society, the Croatian Center for Computer Vision, the Croatian Society for Biomedical Engineering and Medical Physics, and the Centre of Research Excellence for Data Science and Advanced Cooperative Systems.

SVEN LONČARIĆ (Senior Member, IEEE) received the Ph.D. degree in electrical engineering from the University of Cincinnati, Cincinnati, OH, USA, in 1994. He was a Fulbright Scholar at the University of Cincinnati. He was an Assistant Professor with the New Jersey Institute of Technology, Newark, NJ, USA, from 2001 to 2003. He is currently a Professor of electrical engineering and computer science with the Faculty of Electrical Engineering and Computing, University of Zagreb, Croatia. He is also the Director of the Center for Computer Vision, University of Zagreb, the Head of the Image Processing Group, and the Co-Director of the Center of Excellence in Data Science and Cooperative Systems. He was the principal investigator on a number of research and development projects. He has coauthored more than 250 publications in scientific journals and conferences. His research interests include image processing and computer vision. He is a member of the Croatian Academy of Technical Sciences. He was the Chair of the IEEE Croatia Section. He received several awards for his scientific and professional work.

Biography

Donik Vršnak received his B.Sc. in computer science in 2018 and M.Sc. also in computer science in 2020. He is currently a third-year doctoral student in the field of computing at the Faculty of Electrical Engineering and Computing, University of Zagreb, Croatia. The research he is most interested in are image processing and analysis, bioinformatics, astronomy, and machine learning. Research he is currently doing is related to color constancy, focused primarily on multi-illuminant segmentation and estimation, as well as on the perceptional properties of the human visual system related to color constancy. He has authored seven research papers in international journals, as well as three conference papers.

Publications

Journal publications

1. Vršnak, D., Domislović, I., Subašić M., Lončarić S., "Autoencoder-based training for multi-illuminant color constancy", Journal of the Optical Society of America A, Vol. 39, 2022, pp. 1076-1084
2. Domislović, I., Vršnak, D., Subašić M., Lončarić S., "One-net: Convolutional color constancy simplified", Pattern Recognition Letters, Vol. 159, 2022, pp. 31-37
3. Vršnak, D., Domislović, I., Subašić M., Lončarić S., "Illuminant segmentation for multi-illuminant scenes using latent illumination encoding", Signal Processing: Image Communication, Vol. 108, 2022, pp. 116822
4. Vršnak, D., Domislović, I., Subašić M., Lončarić S., "Framework for Illumination Estimation and Segmentation in Multi-Illuminant Scenes", IEEE Access, Vol. 11, 2023, pp. 2128-2137

Conference publications

1. Vršnak, D., Domislović, I., Subašić M., Lončarić S., "Illuminant estimation error detection for outdoor scenes using transformers", 2021 12th International Symposium on Image and Signal Processing and Analysis (ISPA), Zagreb, Croatia, 2021, pp. 276-281

2. Domislović, I., Vršnak, D., Subašić M., Lončarić S., "Outdoor daytime multi - illuminant color constancy", 2021 12th International Symposium on Image and Signal Processing and Analysis (ISPA), Zagreb, Croatia, 2021, pp. 270-275

Životopis

Donik Vršnak dobio je titulu prvostupnika inženjera računarske znanosti 2018. godine, te titulu magistra inženjera računarske znanosti 2020. godine. Trenutno je student treće godine doktorskog studija u polju računarstva na Fakultetu elektrotehnike i računarstva, Sveučilišta u Zagrebu. Njegovo područje istraživanja je u području obrade i analize slike, bioinformatike, astronomije te strojnog učenja. Trenutno područje istraživanja kojim se bavi je u području postojanosti boja, s fokusom na segmentaciju i procjenu osvjetljenja u scenama s više izvora, te perceptivnim svojstvima ljudskog vizualnog sustava. Autor je sedam istraživačkih članaka u međunarodnim časopisima te tri konferencijska članka.